BUILDING AN OPEN SOURCE SOCIAL MEDIA AGGREGATION TIMELINE

A Capstone Project Submitted to Southern Utah University

in partial fulfillment of the requirements for the degree of

Master of Arts in Professional Communication

January 2014

By

Justin Aiken

Capstone Committee:

Jon Smith, Ph.D., Committee Chair

Lance Jackson, M.A.

Lindsey Harvell, Ph.D.

APPROVAL PAGE

The undersigned, appointed by the dean of the Humanities and Social Science, have examined the thesis entitled

Building An Open Source Social Media Aggregation Timeline

Presented by Justin Aiken

a candidate for the degree of Master of Arts in Professional Communication

and hereby that, in our opinion, is worthy of acceptance.

_____

Chair of Committee – Jon Smith, Ph.D.

_____

Committee Member – Lindsey Harvell, Ph.D.

_____

Committee Member – Lance Jackson, M.A.

Abstract

An ever growing number of web services and social media networks allow one to post a lot of content in to many different websites and services; however, there is no simple way to chronologically aggregate the content – especially not in a space controlled by the user. Relying on a site or service to present the user's data through their interface allows for too much noise and muddies the user's message. In response, this project built an open source, expandable timeline that pulls in a user's content from various places on the web, and hosts it on the user's own personal timeline, allowing them to control their self-presentation to the world through the Web.

Keywords: social media, open source, uses and gratifications theory, self-presentation

Table of Contents

APPROVAL PAGE....2

Abstract....3

Introduction....6

Literature Review....7

- Social Media....7
- Conventional Websites....7
- Facebook....7
- Twitter, Google Plus, LinkedIn, and Other Social Media Sites....8
- Uses and Gratification....9
- UGT and the Modern Web....10
- Self Presentation....10
- Online Self Presentation....11
- Facebook's Timeline....12

Rationale....12

- Visibility....13
- Noise....13
- Aggregation....14
- Aggregation through Existing Social Media....15
- Customized or Manual Aggregation....15
- Existing Solutions....16
- Justification....16

Method....17

What It Is....18

Open Source....18

Expendable....19

Technical Detail....20

Evaluation....22

Work Completed....22

Conclusion....23

Limitations and Future Development....23

References....25

Appendices....31

Appendix A: Various Screenshots of the Timeline....31

Appendix B: Examples of Noise Present in Facebook....34

Appendix C: ERD Diagram....35

Appendix D: Source Code....36

My Timeline....37

Demo....106

Github Plugin....171

Health Graph Plugin....200

Twitter Plugin....219

Building an open source social media aggregation timeline

The Internet provides a dizzying array of different websites and services. Almost everything imaginable is available in an online representation. Many people use several different websites on a frequent basis. Internet users want to use a social network to check up on family members, update their online resume, save a new favorite recipe, find a new TV show to watch, upload photos they took, or any number of near-infinite possibilities. Some of the most common activities done on a daily basis include searching for information, communicating via email, social networking, following the news, or just passing time (Pew Internet, 2012).

Social media is one of the most common uses, with 73% of adults using at least one social media site, and a full 42% of adults are now using multiple social media sites (Duggan & Smith, 2013). Non textual content is also on the rise; 54% of internet users have posted – not merely viewed or used, but posted – photos or videos to an online site (Duggan, 2013). With so many avenues that an individual now has available to post their content to, and with more Internet users utilizing multiple avenues to communicate and post content, a new need has arisen to curate and aggregate this content.

For example, a hypothetical photographer posts their pictures on Instagram, writes about technical aspects of photography on a personal blog, and has an online resume page on LinkedIn. If they wanted share that information with potential clients, their choices would be either have to give potential clients all three links, or do a large amount of manual cross-posting. The limitations of these options are fully examined after a closer look at the literature, but neither are ideal.

Following research on Internet usage, social media, and the options available to a user to present their image online, this project consisted of writing a tool that aggregates content posted on various social media sites into a single timeline under the control of the user. The literature is reviewed

below, followed by closer examination of the limitations of existing methods for online self-presentation. Afterwards, the solution I build is detailed, and the work that went into developing the software is explained, along with limitations and future possibilities.

Literature Review

Social Media

One of the largest segments of internet use revolves around social networks, which can be defined as "a dedicated website or other application that enables users to communicate with each other by posting information, comments, messages, images, etc" (“Social Network”, 2013, def. 1). Some social networking sites focus the communication between users that have a connection. This connection can be between one user and another user, or between one user and many users. Still other social networking sites focus on content rather than connection. For example, Reddit allows anyone to post content or communication in any sub community chosen, and anyone viewing can vote or comment on the submission (Mieghem, 2011).

Conventional Websites

Before any social networks or blogging sites emerged, the option for posting content online was the traditional or conventional website. A personal website created the opportunity for anyone to be a mass communicator (Dominck, 2009). Often hosted for free as part of an internet service provider or in return for advertising space, the personal webpage provides an online 'home base', a central presentation of identity in cyberspace (Papacharissi, 2002). The typical personal webpage consists of static content; new information must be added manually if at all; for this reason, personal websites have become less common in the age of constantly updating social media (Zeldman, 2008).

Facebook

Facebook is the largest and most well known of the social networks (Duggan & Smith, 2013).

Founded in 2004, it has expanded to have well over one billion users – with 75% of US college students using Facebook (ibid., 2013), it's more likely anyone reading this paper uses Facebook than that they do not. In fact, Facebook's prevalence is so large, that outside Asia, Facebook is used by fully half of the world's internet users (Hackman, 2012).

Facebook initially provided each user with a profile page, direct messaging (an alternative to email), and a 'wall,' a type of personal message board they can use to share public messages, or have other users write messages on (Smith, 2009). Activity on the users' walls showed up in a news feed, letting a user see what action their friends have done recently, such as wall posts they've made or commented on, or changed aspects of their profile page (Pempak, Yermolayeva, Calvert, 2009). Over time, Facebook has added many other aspects – groups, pages, events, applications, etc, that also show up in the news feed.

Twitter, Google Plus, LinkedIn, and Other Social Media Sites

With the success of Facebook, competing/alternative social networking services have sprung up in market. Some of these focus on a niche; for example, LinkedIn is a 'Facebook for professionals" (McKee, 2009), while Nextdoor is a Facebook clone for physical neighborhoods to interact online (Miller, 2013). Others are competitors offered by large companies seeking to leverage their existing market share into the social arena; Google closed down Google Buzz, their first foray into the social network market (Scott, 2012), and offer Google Plus as their new social network (Kaste, 2011).

Still, other sites don't have an interface similar to Facebook, but facilitate the sharing of content with others. Users of Twitter can post a message (called a 'tweet') limited to 140 characters to their publicly viewable timeline, which will also instantly appear to all of their 'followers' intermixed with all other tweets from people they follow (Signori, Segre, and Polgreen, 2011). Instagram and Flickr are two services that let someone easily share photos online and are in constant competition (Smith, 2013).

Vine has a rapidly growing marketshare for uploading and sharing video (Lunden, 2013). In short, there is no shortage of networks available to post and share content, whether it be text, photo, or video.

Uses and Gratification

Uses and Gratification Theory (UGT) examines the reasons people choose to consume media, and why they choose which media they do (Levy & Windahl, 1984). Rather than a passive image of the typical viewer, it examines the active role media consumers have (Levy & Windahl, 1985, p. 109). In speaking of the effects on those watching television, UGT suggests that "the term 'effect' is misleading because it suggests that television 'does something' to [the viewers]... Nothing can be further than the fact. It is the [viewers] who are most active in this relationship. It is they who use television rather than television that uses them" (Levy & Windahl, p. 1).

In the oft-cited 1973 paper (Katz, Blumler, & Gurevitch), which reviews the then-current state of UGT research, the authors examine the evolution of UGT. It begins by looking at unrelated studies examining such examples as why children are drawn to comics (Lyle, Parker, & Schramm, 1961) or why one might choose to listen to soap operas (Herzog, 1942) that formed the initial research into UGT. They then look at some of the studies that started to tie the disparate threads together, before looking at the state of UGT research at the time. Although the different strands of research were starting to come together, Katz et al (1973) still sought to see a "relevant theory of social and psychological needs" that went beyond a simple catalog and was a "clustering of groups of needs, a sorting out of different levels of need, and a specification of hypotheses linking particular needs with particular media gratifications" (p. 513). Since then, UGT research has indeed grown and expanded, evolving to keep up with cable, then satellite TV, then the Internet, then the rise of streaming video. An an example of a paper examining applications of UGT to the Internet can be found by looking LaRose & Eastin's (2004) look into a new model of media attendance.

UGT and the Modern Web

A study looking at Twitter (Johnson & Yang, 2009) posits that social media allows users opportunities to fulfill motivations and gratifications never before possible. This new potential has seen Uses and Gratification Theory often applied to social networks and webpages. One such study (Raacke & Raacke, 2008) found that a large majority of college students are using social media sites for a large part of their day, and listed eleven separate reasons. A study looking at motivations behind creating personal webpages (Groner, Weibel, & Wissmath, 2010) found that the most common motivators were enjoyment, exercise (of web skills), self-portrayal, presentation of leisure activity, and job-related presentation. Another study examined motivations for Facebook use, and among the gratifications found, a common one was to make themselves look as attractive as possible, and convey a specific impression of themselves (Day, Dong, & Urista, 2009). They also noted that feedback users received from their wall posts caused some users to become addicted to checking Facebook for responses to the presented self (ibid, 2009). Another study looked at Facebook specifically to move past the obvious 'keeping in touch' uses and found that Facebook was an “important tool” for self-presentation, as well as a related use in social capital gamification (Joinson, 2008).

Self Presentation

Self presentation then is a common theme found in UGT research on new media, as well as one this project focused on. Self Presentation Theory explores how people present themselves to others. Goffman (1959) pioneered research in this area, looking at people's outward communication as an actor wearing a mask; thus we we carry out performances in all of our interactions. He states in his first book that “All the world is not, of course, a stage, but the crucial ways in which it isn't are not easy to specify” (Goffman, 1959). The performances we give are shaped by an objective to project a specific image to the audience – a carefully crafted image that the communicator desires to create, based on the

norms of the audience (Barnhart, 1994). Carrying out this performances is often unconscious, and undertaken no matter the mental state or confidence of the performer. Reasons for enacting this performance can range from actively seeking to achieve or change an outcome, such as presenting an image more likely to achieve monetary gain, or simply hide one's embarrassing habits (Goffman, 1959).

Goffman's (1959) theory splits the human psyche into two stages – the front stage where we put performances on for the world, and the backstage area where we relax and be our true self. Impression management is the act of keeping these two areas stages separated, to avoid the dissonance that could be created if the buffer slips (ibid, 1959).

Online Self Presentation

The rise of social media has given new area to study Goffman's theories of self-presentation. Mehdizadeh (2010) found that narcissistic users with low self-esteem were more likely to spend more time online doing crafting their image through self-posting. A 2008 study looked at the identities constructed by Facebook users, finding they “show rather than tell” their identity and stress group or consumer identities over personal identities (Zhao, Grasmuck, & Martin). In looking at self-presentation on social media sites, Hogan (2010) found that a third stage is added to Goffman's front and back stage – web servers that decide which content you post goes to which audience. This role of the curator makes privacy controls very important to ensure user's postings – thoughts they may consider front stage for sharing with some audiences may very well be backstage material for another audience (Hogan, 2010).

Because communicating online gives a greater level of control over non-textual cues, users are able to carefully construct a controlled performances intended to present exactly the image they want to project (Papacharissi, 2009), making social networking sites a tool very suitable for careful self-

presentation. Some have went as far as to call it the 'ideal environment' for self-presentation (Donath and Boyd, 2004). Bullingham and Vasconcelos (2013) did a study examining social media participants, and found their participants were eager to recreate their offline persona self on the site, but a persona with facets edited; they stated that “this emphasizes the key premise in Goffman’s work that, when in ‘front stage’, people deliberately chose to project a given identity” (ibid, 2013, p. 101).

Facebook's Timeline

In 2011, Facebook introduced a new feature called “Timeline: The 'Story' of your Life” (Lessin, 2001), which at a first glance appears a perfect avenue to self-present with. Describing it as the “evolution of your profile”, Facebook described it grandly - “Imagine if there was an easy way to rediscover the things you shared, and collect all your best moments in a single place” (para. 6). Stripped of marketing speak, the actual patent describes a system or program for generating a social timeline, where multiple data items or events based on relationships are ordered based on time (Sittig and Zuckerberg, 2010).

In use, the timeline lets a user pick and choose what to highlight or hide from their profile – Facebook events such as Likes or Wall Posts, new 'life events' like moving or relationship changes; key to being useful to self-presentation, it also allows external applications. Content users have posted to other sites, such as photos to Instagram or videos to Vine, also can show up on this timeline. If someone thinks their Farmville victory is worthy of display on their personal Timeline, Facebook would even allow them star and display it.

Rationale

A study which examined users of a micro-blogging service similar to Twitter, looking closely at a dissonance between gratifications sought and obtained, found that the gratifications the users sought often went partially filled or – unfulfilled entirely (Wang and Zhu, 2012). It is therefore important to

examine the likely shortcomings found in the use of social media. This paper will focus specifically on the limitations encountered in social media for a user engaged in the self-presentation of their image online.

Visibility

One problem is how often and whether people actually see something a user posts to their friends. 'Filter bubbles' describes the algorithmic mean employed by websites to choose what users or messages to show to another user. In the best case, this is based on past history – in the worst case, based on whichever messages the server decides will be most profitable to be show advertisements around (Pariser, 2011). Facebook is therefore editing and curating content (Williams, 2013), and when they decide which of a user's posts to highlight or not, the algorithms are not taking a carefully presented identity into account!

Noise

In many academic writings on communication and communication theories, noise is a factor. Interesting enough, both technical writings on the physical transmission of analog or digital signals and communication theories often use the same types of terms to describe the level of noise interfering with the message; these terms coming from the Shannon-Weaver model, which is first described in a technical paper for Bell Systems (Shannon, 1998). This paper states that “signal may be perturbed by noise or distortion” (pg. 447), before expounding mathematical formulas describing how to calculate signal to noise ratio. Building on Shannon's paper, Henkel (2012) discusses signal-to-noise ratios on social network sites. Describing Facebook, he states that a Facebook wall is overly polluted with irrelevant videos, articles a user's friends have read, obnoxious advertisements (sometimes hilariously ineffective and inapplicable for the user viewing), and communication between people the user does not care about. He goes on to say that despite the massive amount of noise, Facebook still has the

strongest signal compared to other competing networks, and concludes that in general, social media has an issue where either the signal is not strong enough, or there's just too much noise for the signal to get through.

Advertisements are one of the more annoying forms of noise in social media. Whether engaged in a realistic self-portrait or an idealized projection, while the user chooses carefully which content to display on their Facebook profile, advertisers and Facebook are choosing how to use that profile for targeted advertising (for an example of Facebook advertisement placement, see Appendix B). Describing a study on Facebook's advertising, Roberts (2010) says:

> To test the effectiveness of Facebook's advertisers' micro-targeting method, Lessin (2008) did an experiment by creating his own Facebook advertisement. It was an ad targeted to his girlfriend, so he typed in her specific demographic (a Wall Street Journal Reporter, 25 years old, living in San Francisco, graduated from Harvard in 2006, majored in history, etc.) and was able to get the ad directly placed on her Facebook website. This test demonstrated advertisers' ability to nano-target their market in a unique way not seen in traditional advertising (p 26).

While Facebook shows ads around a user's profile, LinkedIn is arguably worse. After creating an online resume and sending the link to a potential client or employer, when they view your profile they will see links to your direct competitors placed all around your resume. Even in a best case scenario where these distractions are absent, somebody viewing a user's social media page is not only seeing content the user wishes displayed, but the interface of the actual social media site hosting the content.

Aggregation

A larger void is created if somebody wishes to aggregate the content they post to various social media sites. As mentioned in the introduction, fully 42% of adult internet users utilize multiple social media sites (Duggan & Smith, 2013). They're sending photos to one site and posting thoughts on another, or posting video on one and research on another. What if a user of multiple sites wants one unified identity to present, where the perception of the identity is based on the content from multiple sources, instead of being colored by which site an observer sees a portion of their content on?

Aggregation through Existing Social Media

In 2009, Facebook opened up their service to allow external sites to hook into their system (Schofield, 2009). This allowed other sites to post their content into a user's Facebook newsfeed alongside activity their friends did within Facebook directly. This method does allow users to bring content into a central place – they can simply set up their Flickr or Vine account so photos or videos posted there all show up in their Facebook timeline. However, this places the users' information from not just Facebook, but other sites as well, into Facebook's "walled garden" - their data is now locked into Facebook (McCown, 2009), and the user is ever more tied into the whims of Facebook's needs when presenting their data for all their online activity. Appendix B shows how this might look.

Customized or Manual Aggregation

Someone wishing their content from multiple sources displayed interweaved together could always duplicate the process of posting. Every time they post content or a message on a website, they could post that same content or a link to the original content, into a central location – perhaps a personal website they fully control. Enterprising users may write their own program in order to aggregate it all onto their own website – but that is an effort requiring technical skills that either place it out of reach or not worth the time cost for the vast majority of the internet population.

Existing Solutions

One alternative is ifthisthenthat (https://ifttt.com/). This is a service that allows one to set up 'triggers' for various 'channels', and perform an action when the trigger occurs. It allows you to cross-post through setting up recipes – for example, you have it tweet whenever you blog, or blog whenever you tweet. It provides a very nice option, but it still doesn't let you go to a central service – you just copy content between various others. It also limited to mapping one service to another; it is not possible to one action to multiple sites. There are also sites that exist for site-specific crossposting; for example, Flickstagram (http://flickstagram.org/) imports Instagram photos to Flickr.

Other services work in the opposite direction from the aggregation needed – they set up an interface where you have a single control panel that lets you manage and schedule content postings *to* social networks. Sites such as Hootsuite (http://hootsuite.com) or Buffer (http://buffer.com) are used by businesses to manage their social media presence. On a smaller scale, Tweetdeck (http://tweetdeck.com) lets individuals manage multiple Twitter accounts as well as Facebook and Linkedin – only those 3 sites though.

Perhaps the service that comes closest to what is proposed here is Tint (http://tint.com) - they produce an aggregated timeline from various social media feeds. However, is a closed, proprietary solution. It is also non-expandable; there is no way to add a new service or site to it. It is not free, and is targeted soley at at business users. The timeline they produced is hosted at tint.com – a user cannot integrate it into their existing website.

Justification

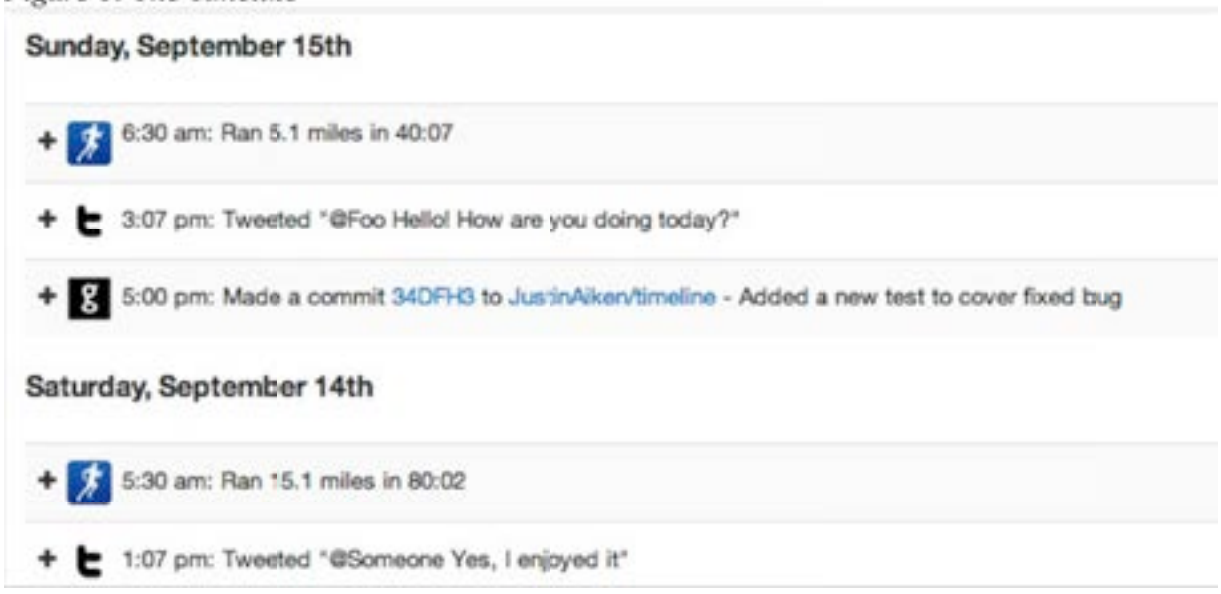
No solution exists that works with an existing personal (or business website), and pulls in content from an infinitely expandable number of social media sites and web services into a single

unified timeline. After initially integrating a solution like this to a website, and adding desired accounts, the user should be done; henceforth any activity carried out normally on the linked sites makes the data available to put on the ever updating timeline, keeping the content on the page fresh and dynamic. This content is presented free of advertisements, competing posts, or third party interfaces – just the user's data displayed in chronological format. Such a solution would keep a user's self-presentation in line with their intended image, and for professional use creates a living portfolio of their web activity.

Method

The project I carried out this semester was building just that solution; writing the software to create a free solution for a social media aggregation timeline. Seeking to create the exact answer to the problems listed above, I programmed a software solution to fulfill the needs experience by the hypothetical users. I called this software simply “my timeline” - see Figure 1 for a screenshot, or Appendix A for more screenshots, showing the same timeline with a handful of different themes or skins applied.

Figure 1: The Timeline



What It Is

The project is an engine built with Ruby on Rails. This is a framework that came about in 2005, and is optimized for programmer happiness and high productivity, letting the programmer write "beautiful code by favoring convention over configuration" (Hansson, 2009, para. 1). It's rapid prototyping took off with the startup scene, and usage exploded; companies like Twitter, Hulu, Shopify, and Scribed pushed it's popularity. Currently over half a million websites are now built using Ruby on Rails (builtwith.com, 2013).

The timeline is written as an *engine* – any existing Ruby on Rails site can incorporate it by simply including a handful of lines of code. The timeline is written to both work with single-user use – such as an individual that already has a personal portfolio website, and desires to integrate an aggregated feed into that – and as multi-user capable, so a large site that has many user accounts can also give a timeline to each user. It could even be used with a skeletal Rails application to create a website that is nothing but the timeline for any number of users. Besides the timeline engine itself, I also built an application that worked just like this; the site existed to host the plugin. This served to both give me a way to demonstrate how the timeline engine works to people, and the source code of the demo app helps developers see how to incorporate timelines into their sites.

Open Source

Key to this project is the notion of "open source." Open source can either mean that the source code used to create a piece of software is freely available and distributed with the project, or on a grander scale that it is built by "developers at many different locations and organizations sharing code to develop and refine programs" (Lerner and Tirole, 2002, p. 197). Much of the technology that powers the internet is built on open source technologies; web server software, operating systems, and frameworks are more often than not freely available community projects.

Github is the largest repository of open source work (Finley, 2013); many of the tools that used to create this project are already hosted there (including the central framework, Ruby on Rails.) Github makes it easy for interested users to clone the source code for a project, made modifications (such as bug fixes or feature enhancements), and push the changes back to the project owner for easy review and inclusion in future releases.

The source code for my timeline engine, the demonstration application, and all associated plugins are all published on Github, were it is free for anyone to download, copy, clone, or send bug fixes and feature enhancements. Besides the current state of the source code, the full history of all changes is also found there. The url for the timeline is http://www.github.com/JustinAiken/my_timeline and the url for the demonstration wrapper is http://www.github.com/JustinAiken/my_timeline-demo. The source code for these is also available in Appendix D. All the code is available under the MIT license (http://opensource.org/licenses/MIT), a more permissive open source license than the common GPL3 public license (2007) - basically it permits users to use the code or program however they like, so long as they don't sue the author.

Expendable

Central to this project is an expendable system. As mentioned earlier in this proposal, the amount of websites, services, and networks is dizzying and ever growing. For this reason, rather than hardcoding any particular service in, the core engine contains only what is needed for central functionality: aggregation, storage, display, authentication, and the ability to create posts outside of any third party site. To actually import or scrape data from other services, it has a flexible add-on system, so that each the import process for each service can have it's own separately maintained codebase. I built several plugins to demonstrate this.

First, I built a plugin for twitter. This connects the users timeline to their twitter account –

anything they tweet then appears on their timeline. Next, I added health graph, which is the service behind Runkeeper. This means that if a user goes for a run while their phone tracks their path, as soon as they finish and save the run, besides pushing the run to Runkeeper, they can also have it pop up on their timeline.

Finally, I built a plugin for Github, letting a user's open source contributions also post on the aggregated timeline. For programmers, this creates a truly useful living portfolio; open source contributors are highly employable: “The number one way of getting a job in any programming company right now is to have a GitHub account and show your work” (Begel, Bosch, and Storey, 2013, p. 52). Having a nice timeline of one's programming activity is more accessible and intuitive than trying to click through all projects they've contributed to on Github, and if this timeline also has some intelligent tweets on the subject aggregated in... it is easy to see how this can be used to create a better link to give out professionally than a reference to a Facebook or Twitter account!

Table 1: Addresses of the plugins created for the project

Twitter	https://www.github.com/JustinAiken/my_timeline-twitter
Github	https://www.github.com/JustinAiken/my_timeline-github
Health Graph	https://www.github.com/JustinAiken/my_timeline-health_graph

Outside of service expandability, the appearance of the timeline is also themeable. The engine renders the events as raw HTML with CSS attached – this allows a designer to construct CSS rules to make the timeline appear however they wish. The timeline can thus easily match an existing website it's integrated into, or a catchy theme can be created. Appendix A shows some screenshots of the timeline with different themes applied.

Technical Detail

This section is presented as an examination of some of the logic and concepts in the code; although tangetial to the communication theory applications of the project, the technical discussion in this section presents a closer look at the work involved with the project. Also of note is Appendix B, where an ERD diagram shows the model scheme used by the core plugin and it's interconnection modules. Most of the code should be fairly self-evident to other developers, since it follows standard Rails conventions, but there are a few places of note that I thought called for a more discussion.

First, in order to make the engine accessible to developers, I sought to make it as easy as possible to integrate into an existing application – I wanted no changes to the host application code to be necessary besides adding the engine/plugins to the Gemfile, tweaking the initializer, and mounting the timeline to the preferred route. To this end, I tried to use as much metaprogramming magic as possible to have the Timeline hook into the other parts of the application it needed access/modifications to, rather than requiring developers to put it in their own code. For example, the engine uses the Rails-settings gem to manage per-user settings, such as Oauth tokens and login info for the various sites; it also reprograms some parts of this library to work with the engine. Rather than requiring a developer to set Rails-settings up on their user model, I lazily initialized the engine, with a post-initialization hook that performed a class_eval on their chosen User model for them (Figure 2).

Figure 2: Expanding the user model

```
RailsSettings::SettingObject.class_eval do
  self.table_name = "my_timeline_settings"

  MyTimeline.config_object = ::RailsSettings::Configuration.new(MyTimeline.user_class) do |s|
    s.key :empty_placeholder
  end

  MyTimeline.user_class.class_eval do
    self.send :include, ::RailsSettings::Base
    self.send :extend,  ::RailsSettings::Scopes

    MyTimeline.config_object.key :twitter, defaults: {foo: "bar"}
    MyTimeline.config_object.key :github, defaults: {foo: "bar"}
  end unless MyTimeline.user_class == MyTimeline::UserStub
end
```

One other aspect calls for particular note. Most route-based mountable engines tend to use their own layouts – Spree, ActiveAdmin, etc. Since this engine is meant to be integrated more tightly with a public view, it uses the application layout; this means that path helpers to application routes would be broken when viewing a my_timeline route. To fix this, the following code was added to handle delegating missing path_to and link_to helpers back to main application:

```
module MyTimeline
  module ApplicationHelper
    def method_missing(meth, *args, &block)
      if meth.to_s =~ /_path$|_url$/
        if main_app.respond_to? meth
          main_app.send meth, *args
        else
          super
        end
      else
        super
      end
    end
```

```
    def respond_to?(meth)
      if meth.to_s =~ /_path$|_url$/
        if main_app.respond_to? meth
          true
        else
          super
        end
      else
        super
      end
    end
  end
end
```

Evaluation

Work Completed

The final work consisted of 6,222 lines of code split across five separate projects (the core engine, the demo app, and the three plugins). There is also ample documentation included in each of the projects to assist other developers that want to work on the project. To help guard against bugs, the timeline project was hooked into a service called Travis-CI (http://travis-ci.org), which checks build status by building the project and running automated tests each time a change was made, to help ensure that a change did not break anything.

To get an objective quantifiable review of the code quality, I hooked each of the five projects up to a service called CodeClimate. Code Climate (http://www.codeclimate.org) is a site that runs automated scans through each of the objects, classes, and constructs used to build the software, and analyses the programming for best practices, the presence of patterns that result in disaster later, and

overly complex or unreadable code. It assigns a numeric grade based on GPA's; 4.0 is the best possible score for code to achieve. Table 2 shows the scores for each of the projects in this work.

Table 2: Code Climate GPA

Project	Core Engine	Demo App	Twitter plugin	Github plugin	Runkeeper plugin
Score	4.0	4.0	4.0	3.6	3.6

The core engine has currently been downloaded 550 times. On the source code page, several people have starred and subscribed to updates. So far, one outside contribution has been made to the code – a user submitted a change to the github plugin to improve the formatting of some documentation. At the time of writing this, one user appears to have started writing an adapter for Pinterest, but it is not yet complete.

Conclusion

For users of social media sites that want to present a specific image online, or at least shape their image – especially users of multiple social media sites – a new option is available that lets them maintain control of their own timeline. This is especially useful for those users that want a portfolio based site for professional reasons, that stays up to date with their activity automatically. As earlier quoted from Facebook: “Imagine if there was an easy way to rediscover the things you shared, and collect all your best moments in a single place” (Lessin, 2001, para 6).

Limitations and Future Development

This is a piece of software that has infinite possibilities for growth. The initial version was focused on filling a void found looking for something for aggregated self-presentation of a few chosen sites. The immediate next step is creating plugins for a plethora of services – I will probably do 5-10 more to cover sites I personally use – time will tell if others start creating plugins to serve their needs

for sites I do not personally user. Past that though, I see the scope of the project expanding beyond pulling in events for display on a public timeline. I would like to grow it into an ever-increasing catalog of all personal data available online. This means expanding it beyond events, to scrape, aggregate, and display non-evented data in a meaningful way. This also means using it for uses outside public presentation. Currently it does has privacy options – a user could for example, hide a Tweet they didn't want to show on their timeline. However, it's primary purpose is public display, and that's how it's tooled. With future expansion, it could be used for self-quantification and personal tracking – a topic moving beyond the realm of communication.

References

Begel, A., Bosch, J., & Storey, M. A. (2013). Social Networking Meets Software Development: Perspectives from GitHub, MSDN, Stack Exchange, and TopCoder. *Software, IEEE, 30*(1), 52-66. doi:10.1109/MS.2013.13

Bullingham, L., & Vasconcelos, A. C. (2013). 'The presentation of self in the online world': Goffman and the study of online identities. *Journal of Information Science, 39*(1), 101-112. doi:10.1177/0165551512470051

Dominick, J. R. (1999). Who do you think you are? Personal home pages and self-presentation on the World Wide Web. *Journalism & Mass Communication Quarterly, 76*(4), 646-658. doi:10.1177/107769909907600403

Duggan, M. (2013, October 28). Photo and video sharing grow online. *Pew Internet.* Retrieved from http://www.pewinternet.org/Reports/2013/Photos-and-videos.aspx

Duggan, M., & Smith, A. (2013, December 30). Social media update 2013. *Pew Internet.* Retrieved from http://www.pewinternet.org/Reports/2013/Social-Media-Update/Main-Findings.aspx

Ferri, F., Grifoni, P., & Guzzo, T. (2012). New forms of social and professional digital relationships: the case of Facebook. *Social network analysis and mining, 2*(2), 121-137. doi:10.1007/s13278-011-0038-4

Finley, K. (2013). Github has surpassed sourceforge and google code in popularity. *ReadWriteWeb.* Retrieved form http://www.readwriteweb.com/hack/2011/06/github-has-passed-sourceforge.php

GNU General Public License. (2013, June 29). *Version 3*. Retrieved from http://www.gnu.org/licenses/gpl-3.0.html

Goffman, E. (1959). *The presentation of self in everyday life*. London: Penguin.

Groner, R., Weibel, D., & Wissmath, B.(2010). Motives for creating a private website and personality of personal homepage owners in terms of extraversion and heuristic orientation. *Cyberpsychology: Journal of Psychosocial Research on Cyberspace, 4*(1).

Hackman, M. (2012, Feburary 02). [Web log message]. *Facebook used by half of the world's internet users, save asia.* Retrieved from http://www.pcmag.com/article2/0,2817,2399732,00.asp

Hansson, D. H. (2009). Ruby on rails. Website. Retrieved from: http://www.rubyonrails.org.

Henkel, D. (2012, July 16). [Web log message]. *Signal to noise – the demise of facebook.* Retrieved from http://devinhenkel.com/compelling/signal-to-noise-the-demise-of-facebook/

Herzog, H. (1940). Professor quiz: A gratification study. *Radio and the printed page*, 64-93.

Hogan, B. (2010). The presentation of self in the age of social media: distinguishing performances and exhibitions online. *Bulletin of Science, Technology & Society, 30*(6), 377-386. doi:10.1177/0270467610385893

Johnson, P. R., & Yang, S. (2009, August). Uses and gratifications of Twitter: An examination of user motives and satisfaction of Twitter use. In *Communication Technology Division of the annual convention of the Association for Education in Journalism and Mass Communication in Boston, MA.*

Joinson, A. N. (2008, April). Looking at, looking up or keeping up with people?: motives and use of facebook. *In Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (pp. 1027-1036). ACM. doi:10.1145/1357054.1357213

Kaste, M. (2011, June 29). Facebook's newest challenger: Google Plus. *NPR*. Retrieved from

http://www.npr.org/2011/06/29/137507567/facebooks-newest-challenger-google-plus

Katz, E., Blumler, J. G., & Gurevitch, M. (1973). Uses and Gratifications Research. *The Public Opinion Quarterly*, *37*(4), 509–523. doi:10.1086/268109

Kim, W., Jeong, O. R., & Lee, S. W. (2010). On social Web sites. *Information Systems, 35*(2), 215-236. doi:10.1016/j.is.2009.08.003

LaRose, R., & Eastin, M. S. (2004). A Social Cognitive Theory of Internet Uses and Gratifications: Toward a New Model of Media Attendance. *Journal of Broadcasting & Electronic Media*, *48*(3), 358–377. doi:10.1207/s15506878jobem4803_2

Lessin, S. (2008). Welcome to the world of nanotargeted ads. *Advertising Age*, *79*(42).

Lessen, S. (2011, September 22). Tell Your Story with Timeline. *Facebook.* Retrieved October 5, 2013, from www.facebook.com/notes/facebook/tell-your-story-with-timeline/10150289612087131

Levy, M. R., Windahl, S. (1984). Audience activity and gratifications: A conceptual clarification and exploration. *Communication Research, 11*, 51–78. doi:10.1177/009365084011001003

Levy, M. R., Windahl, S. (1985). The concept of audience activity. *Media gratifications research: Current perspectives*: 109–122. doi:10.1177/009365084011001003

Lerner, J., & Tirole, J. (2002). Some simple economics of open source. *The journal of industrial economics*, *50*(2), 197-234. doi:10.1111/1467-6451.00174

Lunden, I. (2013, March 14). Early vine use sees video app rising on ios while cinemagram, viddy, socialcam all decline. *Techcrunch.com*. Retrieved from http://techcrunch.com/2013/03/14/early-vine-use-sees-video-app-rising-on-ios-while-cinemagram-viddy-socialcam-all-decline/

Lyle, J., Parker, E. B., Schramm, W. (1961). Television in the lives of our children.

Stanford. Stanford University Press.

McCown, F., & Nelson, M. L. (2009, June). What happens when facebook is gone?. *Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries* (pp. 251-254). ACM. doi:10.1145/1555400.1555440

Mehdizadeh, S. (2010). Self-presentation 2.0: Narcissism and self-esteem on Facebook. *Cyberpsychology, Behavior, and Social Networking*, *13*(4), 357-364. doi:doi:10.1089/cyber.2009.0257

Miller, B. (2013, December 3). Social networking site brings neighbors together. *Star Telegram*. Retrieved from http://www.star-telegram.com/2013/11/30/5386755/social-networking-site-brings.html

Van Mieghem, P. (2011). Human psychology of common appraisal: the reddit score. *Multimedia, IEEE Transactions on*, *13*(6), 1404-1406. doi:10.1109/TMM.2011.2165054

Pariser, E. (2011). Beware Online" Filter Bubbles"(Ted Talk).

Papacharissi, Z. (2009). The virtual geographies of social networks: a comparative analysis of Facebook, LinkedIn and ASmallWorld. *New Media & Society, 11*(1-2), 199-220. doi:10.1177/1461444808099577

Pempek, T. A., Yermolayeva, Y. A., & Calvert, S. L. (2009). College students' social networking experiences on Facebook. *Journal of Applied Developmental Psychology, 30*(3), 227-238. doi:10.1016/j.appdev.2008.12.010

Pew Internet (2012, February). What Internet Users Do On A Typical Day. Retrieved from http://www.pewinternet.org/Static-Pages/Trend-Data-(Adults)/Online-Activities-Daily.aspx

Porter, J. (2010). Designing for the social web. Peachpit Press.

Raacke, J., & Bonds-Raacke, J. (2008). MySpace and Facebook: Applying the uses and gratifications theory to exploring friend-networking sites. *Cyberpsychology & behavior, 11*(2), 169-174. doi:10.1089/cpb.2007.0056

Roberts, K. K. (2010). Privacy and perceptions: How Facebook advertising affects its users. *The Elon Journal of Undergraduate Research in Communications, 1*(1), 24-34.

Schonfeld, E. (2009, April 27). Facebook opens up its stream API to developers. *Techcrunch.com.* Retrieved from http://techcrunch.com/2009/04/27/facebook-opens-up-its-stream-api-to-developers/

Shannon, C. E. (1998). Communication in the presence of noise. *Proceedings of the IEEE, 86*(2), 447-457. doi:10.1109/JRPROC.1949.232969

Signorini, A., Segre, A. M., & Polgreen, P. M. (2011). The use of Twitter to track levels of disease activity and public concern in the US during the influenza A H1N1 pandemic. *PloS one, 6*(5). doi:10.1371/journal.pone.0019467

Smith, C. (2013). Instagram users continue drop, flickr on rise. *Hypebot.com*. Retrieved from: http://www.hypebot.com/hypebot/2013/01/instagram-users-continue-drop-flickr-on-rise-does-it-matter-for-music-marketing.html.

Smith, J. (2009, February 3). Facebook turns 5 years old – a look at facebook through the years. *Insidefacebook.com.* Retrieved from http://www.insidefacebook.com/2009/02/03/facebook-turns-5-years-old-a-look-at-facebook-through-the-years/

Sittig, A., & Zuckerberg, M. (2010). U*.S. Patent No. 7,725,492*. Washington, DC: U.S. Patent and Trademark Office.

Mckee, S. (2009 Janurary 1). Why social media is worth small business owners' time. *Business Week*

Online. Retrieved from http://www.businessweek.com/stories/2009-01-16/why-social-media-is-worth-small-business-owners-timebusinessweek-business-news-stock-market-and-financial-advice

Social Network. (2013). In *OxfordDictionaries.com*. Retrieved from http://www.oxforddictionaries.com/us/definition/american_english/social-network

Urista, M. A., Dong, Q., & Day, K. D. (2009). Explaining why young adults use MySpace and Facebook through uses and gratifications theory. *Human Communication, 12*(2), 215-229.

Wang, Q., & Zhu, Y. (2012, July). Research on relationship of gratification sought, gratification obtained and microblogging user behavior. In *Service Systems and Service Management (ICSSSM), 2012 9th International Conference on* (pp. 325-330). IEEE. 2012. doi:10.1109/ICSSSM.2012.6252246

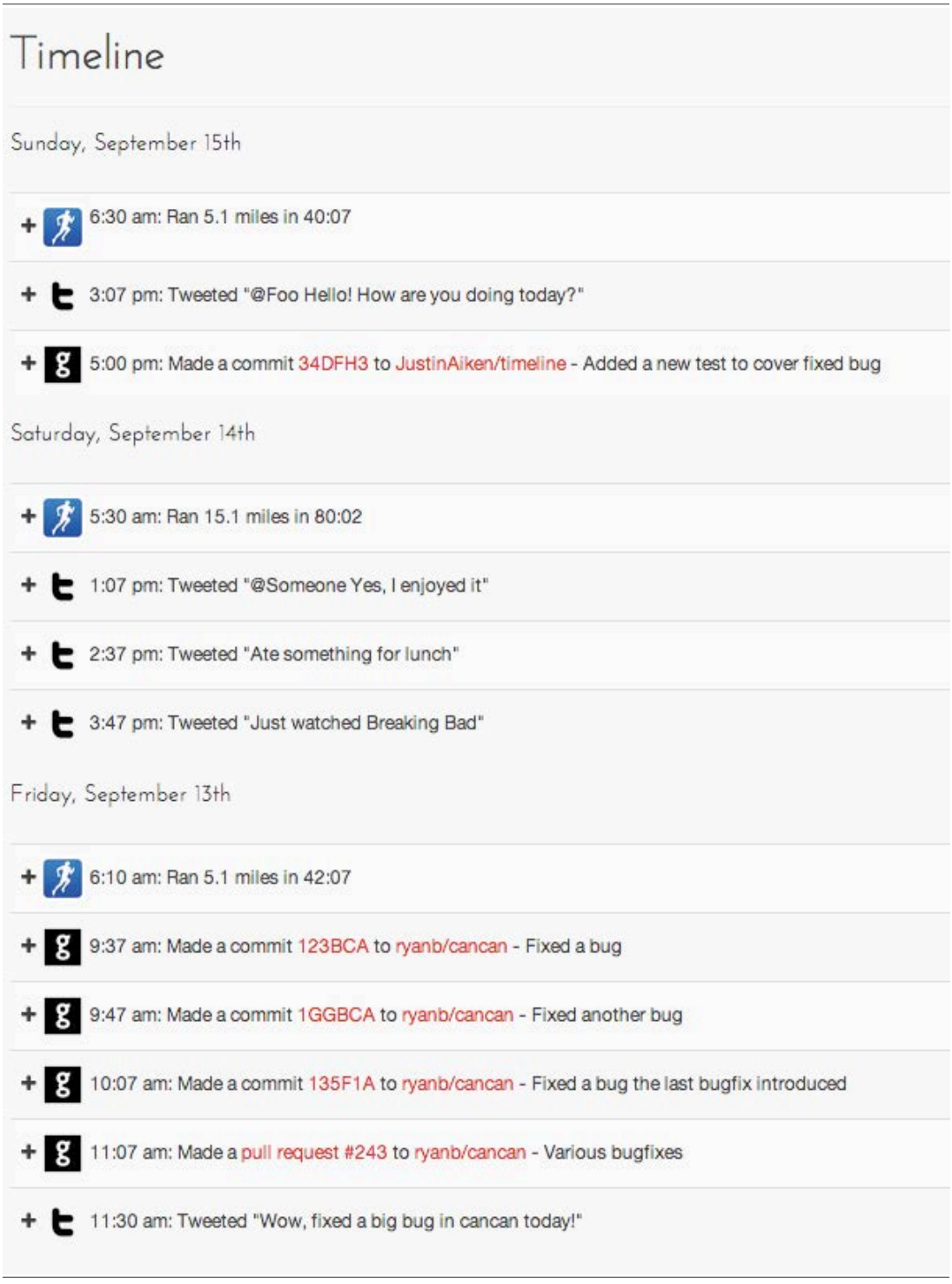
Williams, I. (2013, December 5). The gender identify of facebook: is it a boy or a girl, and has that affected the mobilizing of its gender? Retrieved from https://onlineacademiccommunity.uvic.ca/iwanwilliams/2013/12/05/the-gender-identify-of-facebook-is-it-a-boy-or-a-girl-and-has-that-affected-the-mobilizing-of-its-gender/

Zeldman, L. J. (2008 April 27). *The vanishing personal site.* Retrieved from http://www.zeldman.com/2008/04/27/content-outsourcing-and-the-disappearing-personal-site

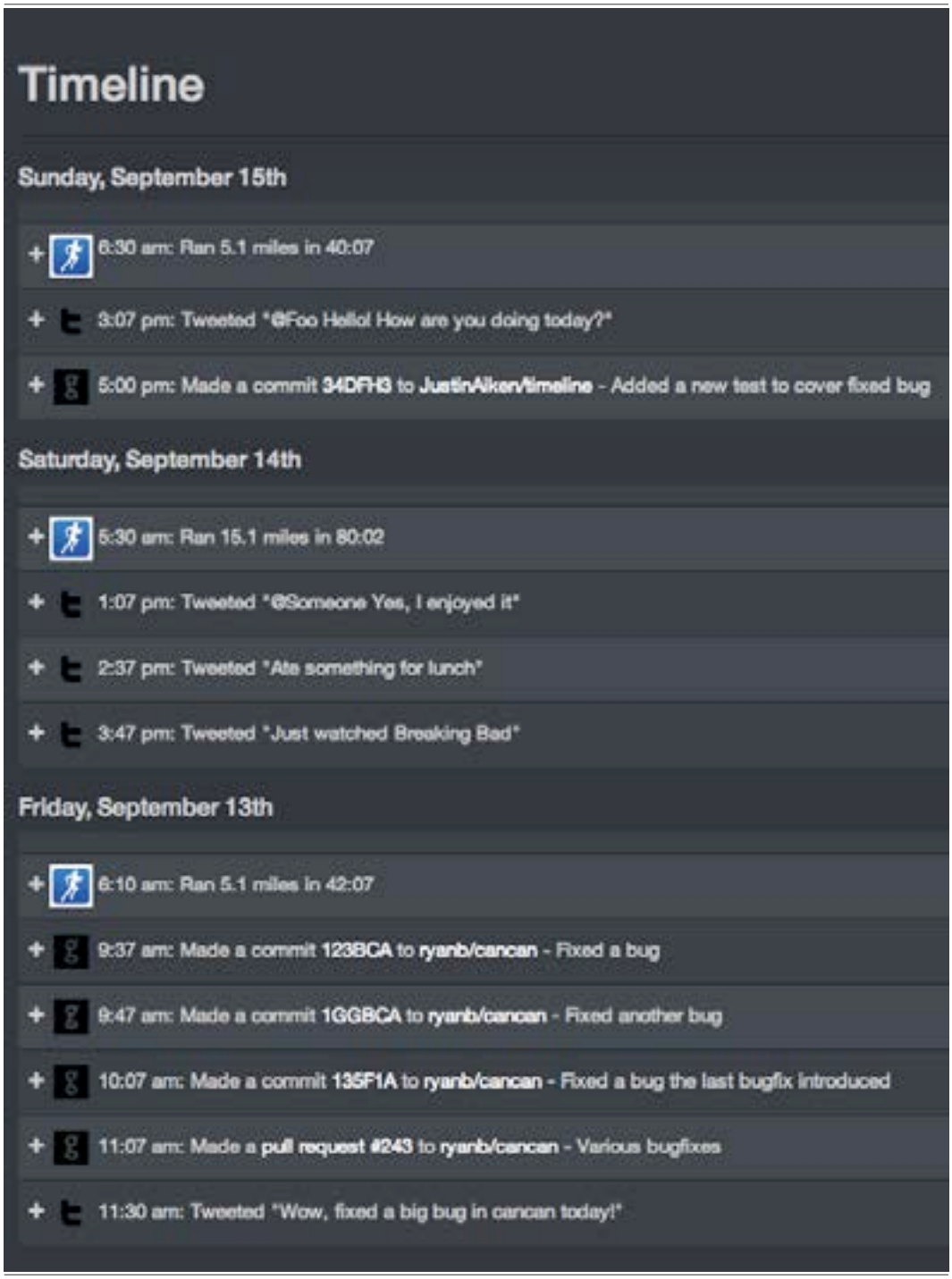
Zhao, S., Grasmuck, S., & Martin, J. (2008). Identity construction on Facebook: Digital empowerment in anchored relationships. *Computers in human behavior, 24*(5), 1816-1836. doi:10.1016/j.chb.2008.02.012

Appendix A: Various Screenshots of the Timeline

Screenshot 1: Plain Theme



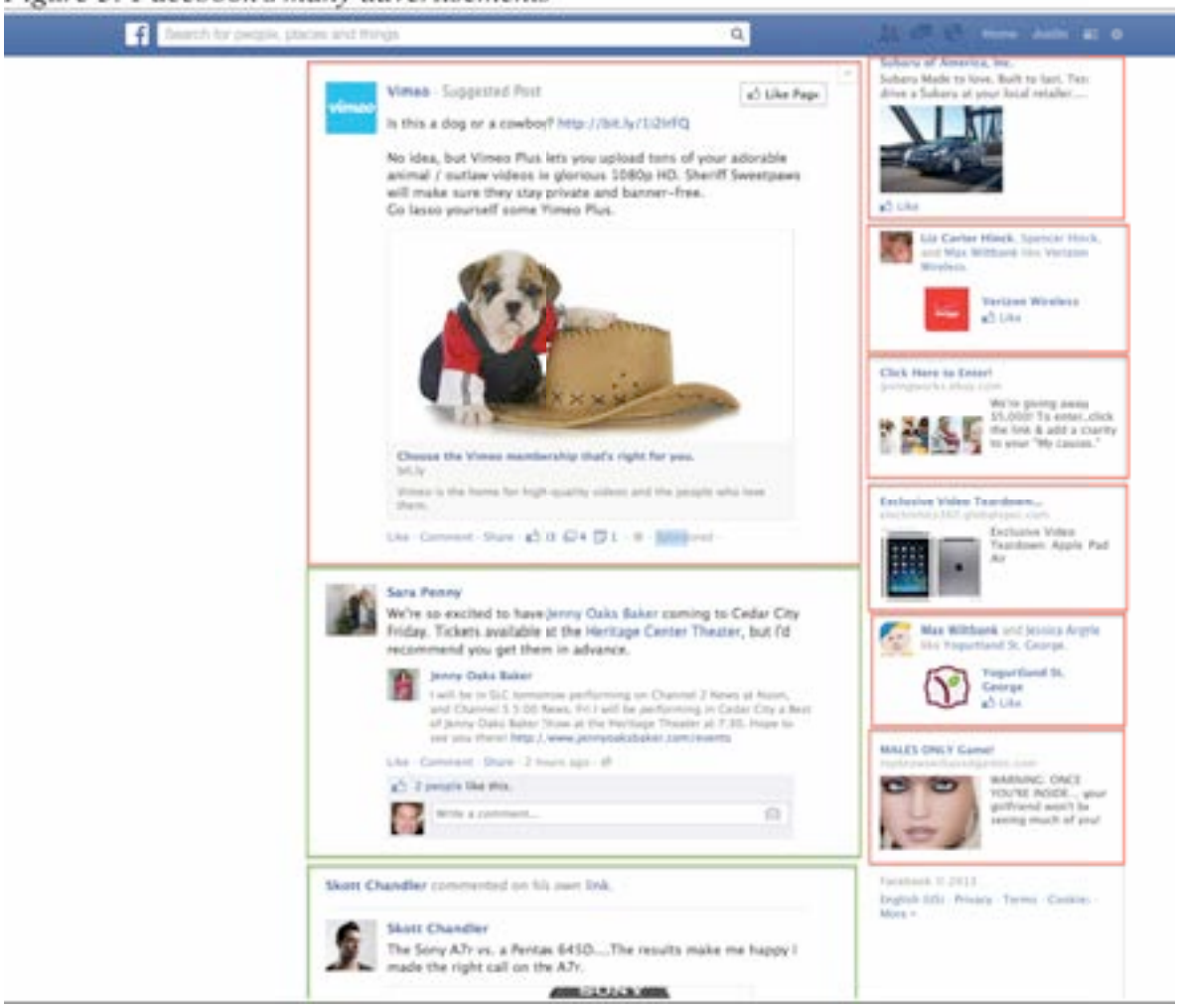
Screenshot 2: Black Theme



Screenshot 3: Green Theme

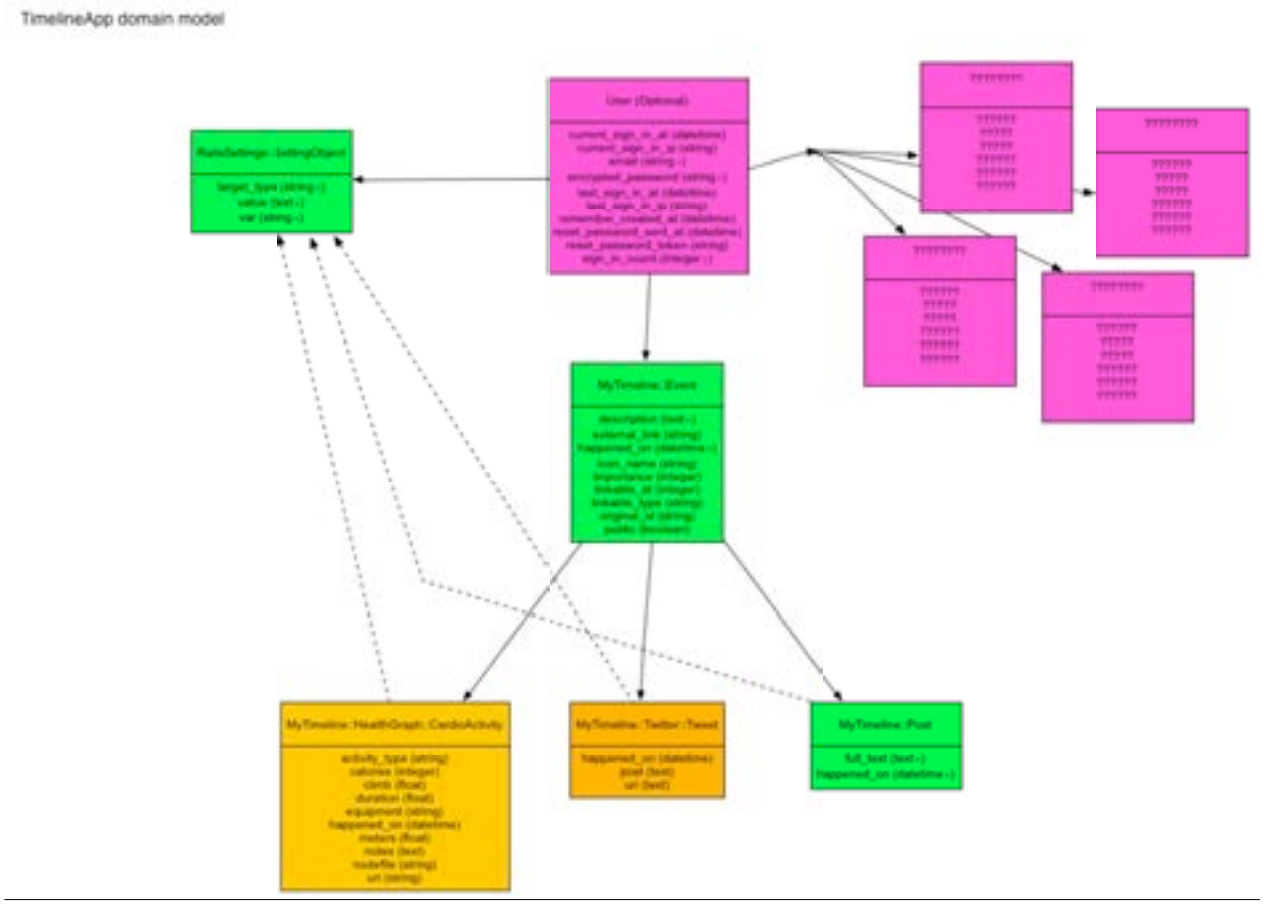
Appendix B: Examples of Noise Present in Facebook

Figure 3: Facebook's many advertisements



Appendix C: ERD Diagram

Figure 4: Entity Relationship between Timeline models



APPENDIX D

The remainder of this document contains the source code for each of the my_timeline repositories. A handful of files have been removed from the primary plugin (namely some of the boilerplate for the dummy application used for tests) as well as the demo application (mostly the stylesheets used to create the theme). The other three plugins are presented in their entirety. These version are 0.1.0 of each – future updates will be available at:

- https://www.github.com/JustinAiken/my_timeline
- https://www.github.com/JustinAiken/my_timeline-demo
- https://www.github.com/JustinAiken/my_timeline-twitter
- https://www.github.com/JustinAiken/my_timeline-github
- https://www.github.com/JustinAiken/my_timeline-health_graph

/my_timeline/README.markdown

````
# My Timeline [![Code Climate](https://codeclimate.com/github/JustinAiken/my_timeline.png)](https://codeclimate.com/github/JustinAiken/my_timeline) [![Build Status](https://secure.travis-ci.org/JustinAiken/my_timeline.png?branch=master)](http://travis-ci.org/JustinAiken/my_timeline)
#### A social-media aggregation/display plugin

This is a Rails Engine to help pull in content from any number of social media sites, services, or websites.
The aggregated information is displayed in a unified timeline.

It is being developed with extensibility in mind - each service will have it's own plugin.

### What it looks like:
![Screenshot](doc/screenshot.png)

### Requirements:
- Ruby 1.9.3 or 2.x
- Rails 3.1.x or 3.2.x or 4.x
- Bootstrap (or bootstrap-named classes) - For the markup.  Just stuff like `table.table-striped`, no stuctural markup from Bootstrap is needed
- Any standard ActiveRecord-compatible database should work

### Supported services:

- [Runkeeper](https://github.com/JustinAiken/my_timeline-health_graph)
- [Twitter](https://github.com/JustinAiken/my_timeline-twitter)
- [Github](https://github.com/JustinAiken/my_timeline-github)
- If you develop another, let me know and I'll add it here!

### Demonstration

There is a [small demo app](https://github.com/JustinAiken/my_timeline-demo) available to show how it looks inside a fresh Rails application with a Devise User system.

### Usage:

1. Add the gem to your Gemfile: `gem 'my_timeline'` and `bundle install`
2. Install the config file: `rails g my_timeline:install`
3. Edit `config/initializers/my_timeline.rb` to taste
4. Mount the engine in your routes:
  ```ruby
  # A timeline belongs_to User
  resources :users do
    mount MyTimeline::Engine => '/timeline', as: :my_timeline
  end
  ```

  or

  ```ruby
  # No Users, just a dedicated timeline route
    mount MyTimeline::Engine => '/timeline', as: :my_timeline
  ```
5. Add a gem for any service you'd like to add on.

## Credits

Original author: [Justin Aiken](https://github.com/JustinAiken)

## Links

* [Source](https://github.com/JustinAiken/my_timeline)
* [Bug Tracker](https://github.com/JustinAiken/my_timeline/issues)
* [Rubygem](https://rubygems.org/gems/my_timeline)

## Note on Patches/Pull Requests

* Fork the project.
````

/my_timeline/README.markdown

```
63  * Make your feature addition or bug fix.
64  * Add tests for it. This is important so I don't break it in a future version unintentionally.
65  * Commit, do not mess with rakefile, version, or history.
66    * If you want to have your own version, that is fine but bump version in a commit by itself so I can i
    gnore when I pull
67  * Send me a pull request. Bonus points for topic branches.
68
69  ## Copyright
70
71  Copyright (c) 2013 Justin Aiken Inc. MIT license (see LICENSE for details).
72
```

/my_timeline/LICENSE

```
Copyright 2013 Justin Aiken

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

```

/my_timeline/CHANGELOG.markdown

```
1  # 0.1.0
2    * [BUGFIX] Fix the settings to reload in development mode
3    * [BUGFIX] Fix the user stub
4    * [FEATURE] Display event's times with the User's timezone, and make the time formatting configura
   ble
5    * [BUGFIX] Make the plugin registry a Set instead of an Array to avoid duplicates
6
7  # 0.0.5
8    * [FEATURE] Added a detail view that expands below the summary post, if that model supports an ex
   pandable view (defaults to not.)
9    * [BUGFIX] Dehardcode header text
10
11 # 0.0.4
12   * [FEATURE] Added Rails 4 compatibility
13   * [FEATURE] Added more tests
14
15 # 0.0.3
16   * [BUGFIX] Fixed external event link
17
18 # 0.0.2
19   * Many, many bugfixes
20
21 # 0.0.1
22   * Initial release
23
```

/my_timeline/my_timeline.gemspec

```
1  $:.push File.expand_path("../lib", __FILE__)
2
3  require "my_timeline/version"
4
5  Gem::Specification.new do |s|
6    s.name        = "my_timeline"
7    s.version     = MyTimeline::VERSION
8    s.authors     = ["Justin Aiken"]
9    s.email       = ["60tonangel@gmail.com"]
10   s.homepage    = "https://www.github.com/JustinAiken/my_timeline"
11   s.summary     = "Social Media Aggregation Timeline"
12   s.description = "Social Media Aggregation Timeline"
13
14   s.license     = 'MIT'
15
16   s.files         = `git ls-files`.split("\n")
17   s.test_files    = `git ls-files -- {spec}/*`.split("\n")
18
19   s.add_runtime_dependency "rails", ['> 2.0']
20   s.add_runtime_dependency "kaminari"
21   s.add_runtime_dependency 'ledermann-rails-settings'
22
23   s.add_development_dependency 'sqlite3'
24   s.add_development_dependency 'rspec-rails'
25   s.add_development_dependency 'capybara'
26   s.add_development_dependency 'factory_girl_rails'
27   s.add_development_dependency 'guard-rspec'
28 end
29
```

/my_timeline/Gemfile

```
1  source "https://rubygems.org"
2
3  gemspec
4
5  # jquery-rails is used by the dummy application
6  gem "jquery-rails"
7
```

/my_timeline/Rakefile

```
1  #!/usr/bin/env rake
2  begin
3    require 'bundler/setup'
4  rescue LoadError
5    puts 'You must `gem install bundler` and `bundle install` to run rake tasks'
6  end
7
8  begin
9    require 'rdoc/task'
10 rescue LoadError
11   require 'rdoc/rdoc'
12   require 'rake/rdoctask'
13   RDoc::Task = Rake::RDocTask
14 end
15
16 RDoc::Task.new(:rdoc) do |rdoc|
17   rdoc.rdoc_dir = 'rdoc'
18   rdoc.title    = 'MyTimeline'
19   rdoc.options << '--line-numbers'
20   rdoc.rdoc_files.include('README.rdoc')
21   rdoc.rdoc_files.include('lib/**/*.rb')
22 end
23
24 APP_RAKEFILE = File.expand_path("../spec/dummy/Rakefile", __FILE__)
25 load 'rails/tasks/engine.rake'
26
27 Bundler::GemHelper.install_tasks
28
29 Dir[File.join(File.dirname(__FILE__), 'tasks/**/*.rake')].each {|f| load f }
30
31 require 'rspec/core'
32 require 'rspec/core/rake_task'
33
34 desc "Run all specs in spec directory (excluding plugin specs)"
35 RSpec::Core::RakeTask.new spec: 'app:db:test:prepare'
36
37 task default: :spec
38
```

/my_timeline/Guardfile

```
guard :rspec, cmd: 'zeus rspec --color --format nested --fail-fast', all_after_pass: false, all_on_start: false do
  watch(%r{^spec/.+_spec\.rb$})
  watch(%r{^lib/(.+)\.rb$})     { |m| "spec/lib/#{m[1]}_spec.rb" }
  watch('spec/spec_helper.rb')  { "spec" }

  # Rails
  watch(%r{^app/(.+)\.rb$})                           { |m| "spec/#{m[1]}_spec.rb" }
  watch(%r{^app/(.*)(\.erb|\.haml|\.slim)$})          { |m| "spec/#{m[1]}#{m[2]}_spec.rb" }
  watch(%r{^app/controllers/(.+)_(controller)\.rb$})  { |m| ["spec/routing/#{m[1]}_routing_spec.rb", [
  watch(%r{^spec/support/(.+)\.rb$})                  { "spec" }
  watch('config/routes.rb')                           { "spec/routing" }
  watch('app/controllers/application_controller.rb')  { "spec/controllers" }

  # Capybara features
  watch(%r{^app/views/(.+)/.*\.(erb)$})      { |m| "spec/features/#{m[1]}_spec.rb" }
end
```

/my_timeline/zeus.json

```
{
  "command": "ruby -rubygems -r./engine_plan -eZeus.go",

  "plan": {
    "boot": {
      "default_bundle": {
        "development_environment": {
          "prerake": {"rake": []},
          "runner":   ["r"],
          "console":  ["c"],
          "generate": ["g"],
          "destroy":  ["d"]
        },
        "test_environment": {
          "test_helper": {"test": ["rspec"]}
        }
      }
    }
  }
}

```

/my_timeline/.gitignore

```
1  .bundle/
2  log/*.log
3  pkg/
4  spec/dummy/db/*.sqlite3
5  spec/dummy/log/*.log
6  spec/dummy/tmp/
7  spec/dummy/.sass-cache
8  Gemfile.lock
9  .ruby-version
10 .ruby-gemset
11
```

/my_timeline/.travis.yml

```
1  language: ruby
2  rvm:
3    - 1.9.3
4    - 2.0.0
5  env:
6    - DB=sqlite
7  gemfile:
8    - gemfiles/Gemfile.rails-3.x
9    - gemfiles/Gemfile.rails-4.x
10 script:
11   - RAILS_ENV=test bundle exec rake db:migrate
12   - bundle exec rake
13 before_script:
14
```

/my_timeline/engine_plan.rb

```
1  require 'zeus/rails'
2
3  ROOT_PATH   = File.expand_path(Dir.pwd)
4  ENV_PATH    = File.expand_path('spec/dummy/config/environment', ROOT_PATH)
5  BOOT_PATH   = File.expand_path('spec/dummy/config/boot', ROOT_PATH)
6  APP_PATH    = File.expand_path('spec/dummy/config/application', ROOT_PATH)
7  ENGINE_ROOT = File.expand_path(Dir.pwd)
8  ENGINE_PATH = File.expand_path('lib/my_timeline/engine', ENGINE_ROOT)
9
10 class EnginePlan < Zeus::Rails
11 end
12
13 Zeus.plan = EnginePlan.new
14
```

/my_timeline/db/migrate/20131103135539_create_my_timeline_posts.rb

```
1  class CreateMyTimelinePosts < ActiveRecord::Migration
2    def change
3      create_table :my_timeline_posts do |t|
4        t.text      :full_text
5        t.datetime :happened_on
6
7        t.references :event
8        t.timestamps
9      end
10   end
11 end
12
```

/my_timeline/db/migrate/20131027171920_create_my_timeline_events.rb

```
1  class CreateMyTimelineEvents < ActiveRecord::Migration
2    def change
3      create_table :my_timeline_events do |t|
4        t.text :description
5        t.datetime :happened_on
6        t.string :icon_name
7        t.string :external_link
8        t.string :original_id
9        t.boolean :public, default: true
10       t.integer :importance, default: 5
11
12       t.references :user
13       t.references :linkable, :polymorphic => true
14
15       t.timestamps
16     end
17   end
18 end
19
```

/my_timeline/db/migrate/20131103000200_create_my_timeline_settings.rb

```
1  class CreateMyTimelineSettings < ActiveRecord::Migration
2
3    def change
4      create_table :my_timeline_settings do |t|
5        t.string      :var,     :null => false
6        t.text        :value
7        t.references :target, :null => false, :polymorphic => true
8
9        t.timestamps
10     end
11
12     add_index :my_timeline_settings, [ :target_type, :target_id, :var ], :unique => true, :name => "
13   end
14 end
15
```

/my_timeline/app/views/my_timeline/posts/new.html.erb

```
1  <h3> New Post </h3>
2
3  <%= render partial: "form"  %>
4
```

/my_timeline/app/views/my_timeline/posts/_form.html.erb

```
1  <%= form_for @post do |f| %>
2    <% if @post.errors.any? %>
3      <div class="alert alert-error">
4        <h2><%= pluralize(@post.errors.count, "error") %> prohibited
5          this post from being saved:</h2>
6        <ul>
7        <% @post.errors.full_messages.each do |msg| %>
8          <li><%= msg %></li>
9        <% end %>
10       </ul>
11     </div>
12   <% end %>
13
14   <%= f.label :happened_on, 'Happened On' %>
15   <%= f.datetime_select :happened_on, {prompt: { day: 'Select day', month: 'Select month', year:
16   <br />
17
18   <%= f.fields_for :event do |ff| %>
19     <div>
20       <%= ff.label :description, 'Description' %>
21       <%= ff.text_field :description %>
22
23       <%= ff.label :public, 'Publically viewable' %>
24       <%= ff.check_box :public %>
25     </div>
26   <% end %>
27
28   <%= f.label :full_text, 'Text:' %>
29   <%= f.text_area :full_text %>
30   <br />
31
32   <br />
33
34   <%= f.submit %>
35 <% end %>
36
```

/my_timeline/app/views/my_timeline/events/edit.html.erb

```
1  <%= form_for @event do |f| %>
2    <% if @event.errors.any? %>
3      <div class="alert alert-error">
4        <h2><%= pluralize(@event.errors.count, "error") %> prohibited
5          this event from being saved:</h2>
6        <ul>
7        <% @event.errors.full_messages.each do |msg| %>
8          <li><%= msg %></li>
9        <% end %>
10       </ul>
11     </div>
12   <% end %>
13
14   <%= f.label :happened_on, 'Happened On' %>
15   <%= f.text_field :happened_on %>
16   <br />
17
18   <%= f.label :description, 'Description' %>
19   <%= f.text_field :description %>
20
21   <%= f.label :public, 'Publically viewable' %>
22   <%= f.check_box :public %>
23
24   <br />
25
26   <%= f.submit %>
27 <% end %>
28
```

/my_timeline/app/views/my_timeline/events/index.html.erb

```
1  <%= javascript_include_tag "my_timeline/events" %>
2
3  <% if t "my_timeline.timeline_header", default: nil %>
4    <div class="page-header">
5      <small><%= t "my_timeline.timeline_header" %></small>
6    </div>
7  <% end %>
8
9  <% @dates_with_events.each do |day| %>
10   <h4><%= date_header_string day[:date] %></h4>
11
12   <%= render partial: "my_timeline/events/day_with_events_#{MyTimeline.render_method}", loca
     ls: {events: day[:events]} %>
13 <% end %>
14
15 <%= paginate @events %>
16
```

/my_timeline/app/views/my_timeline/events/_event.html.erb

```
<p>
  <% if event.linkable.class.respond_to? :is_exandable? %>
    <%= link_to "#", class: "event_expand", id: "event_#{event.id}" do %>
      <%= glyph 'plus-sign' %>
    <% end %>
  <% end %>
  <%= link_to event.external_link do %>
  <%= image_tag event.icon_path, size: "32x32" %>
<% end %>
<%= event.happened_on %>
<%= raw event.description %>
<% if @owner_viewing %>
  <%= link_to edit_event_path(event.id) do %>
    <%= glyph 'pencil' %>
  <% end %>
  <%= link_to event_path(event.event), method: :delete do %>
    <%= glyph 'remove-sign' %>
  <% end %>
<% end %>
</p>
<% if event.linkable.class.respond_to? :is_exandable? %>
  <div class="event_details" id="event_<%= event.id %>" style="display:none;">
  </div>
  <span class="event_url" id="event_url_<%= event.id %>" style="display:none;">
    <%= polymorphic_url [my_timeline, event.linkable] %>
  </span>
<% end %>

```

/my_timeline/app/views/my_timeline/events/_day_with_events_list.html.erb

```
1  <ul>
2    <% events.each do |event| %>
3      <li>
4        <%= render partial: 'my_timeline/events/event', object: MyTimeline::EventPresenter.new(event
5      </li>
6    <% end %>
7  </ul>
8
```

/my_timeline/app/views/my_timeline/events/_day_with_events_table.html.erb

```
1  <table class="<%= MyTimeline.table_class %>">
2    <thead>
3      <th></th>
4    </thead>
5    <tbody>
6      <% events.each do |event| %>
7        <tr>
8          <td style="vertical-align:middle;">
9            <%= render partial: 'my_timeline/events/event', object: MyTimeline::EventPresenter.new(ev
   ent) %>
10         </td>
11       </tr>
12     <% end %>
13   </tbody>
14 </table>
15
```

/my_timeline/app/views/my_timeline/control_panel/index.html.erb

```
1  <div class="page-header">
2    <h1><%= I18n.t("my_timeline.control_panel.header") %></h1>
3  </div>
4
5  <div>
6    <h3>My Settings</h2>
7    <%= render partial: "my_timeline/control_panel/time_zone" %>
8  </div>
9
10 <hr>
11
12 <% @enabled_plugins.each do |plug| %>
13   <%= render partial: "my_timeline/#{plug}/control_panel" %>
14   <hr>
15 <% end %>
16
17 <h3> New Post </h3>
18 <%= link_to "Add", new_post_path %> a self-contained post.
19
```

/my_timeline/app/views/my_timeline/control_panel/_time_zone.html.erb

```
1  <%= form_for @user, url: control_panel_timezone_path, method: :post do |f| %>
2    <%= f.time_zone_select :time_zone %>
3    <br>
4    <%= f.submit 'Save', class: "btn btn-primary" %>
5  <% end %>
6
```

/my_timeline/app/assets/javascripts/my_timeline/events.js.coffee

```
$ ->
  $("a.event_expand").click (event) ->
    event.preventDefault()

    event_id = $(this).attr("id").replace /[A-Z_a-z$-]/g, ""
    div_id   = "div#event_" + event_id
    url      = $("span#event_url_" + event_id).text()

    if $(div_id).css('display') == 'none'
      $.ajax url,
        type: 'GET'
        dataType: 'html'
        error: (jqXHR, textStatus, errorThrown) ->
          $(div_id).html errorThrown
          $(div_id).toggle()

        success: (data, textStatus, jqXHR) ->
          $(div_id).html data
          $(div_id).toggle()
    else
      $(div_id).toggle()


```

/my_timeline/app/assets/stylesheets/my_timeline/application.css

```
1  /*
2   * This is a manifest file that'll be compiled into application.css, which will include all the files
3   * listed below.
4   *
5   * Any CSS and SCSS file within this directory, lib/assets/stylesheets, vendor/assets/stylesheets,
6   * or vendor/assets/stylesheets of plugins, if any, can be referenced here using a relative path.
7   *
8   * You're free to add application-wide styles to this file and they'll appear at the top of the
9   * compiled file, but it's generally better to create a new file per style scope.
10  *
11  *= require_self
12  *= require_tree .
13  */
14
```

/my_timeline/app/models/my_timeline/post.rb

```
1  module MyTimeline
2    class Post < ActiveRecord::Base
3
4      unless rails4?
5        attr_accessible :happened_on, :full_text
6        attr_accessible :event, :event_id, :event_attributes
7      end
8
9      belongs_to :event, dependent: :destroy
10
11     validates :happened_on, presence: true
12     validates :full_text, presence: true
13
14     accepts_nested_attributes_for :event
15
16     def self.is_exandable?
17       true
18     end
19   end
20 end
21
```

/my_timeline/app/models/my_timeline/event.rb

```
1  module MyTimeline
2    class Event < ActiveRecord::Base
3
4      unless rails4?
5        attr_accessible :description, :happened_on, :icon_name, :external_link, :original_id, :public, :
   importance
6        attr_accessible :user, :linkable, :user_id, :linkable_type, :linkable_id
7      end
8
9      belongs_to :linkable, polymorphic: true, dependent: :delete
10     belongs_to :user, class_name: MyTimeline.user_class.to_s
11
12     validates :description, presence: true
13     validates :happened_on, presence: true
14     validates :importance, inclusion: {in: 1..10, allow_blank: true, message: "%{value} is not
   between 1-10." }
15
16     scope :desc, order("my_timeline_events.happened_on DESC")
17   end
18 end
19
```

/my_timeline/app/helpers/my_timeline/events_helper.rb

```
1  module MyTimeline
2    module EventsHelper
3      def date_header_string(date)
4        date.strftime "%b #{date.day.ordinalize}, %Y"
5      end
6    end
7  end
8
```

/my_timeline/app/helpers/my_timeline/application_helper.rb

```
 1  module MyTimeline
 2    module ApplicationHelper
 3      def method_missing(meth, *args, &block)
 4        if meth.to_s =~ /_path$|_url$/
 5          if main_app.respond_to? meth
 6            main_app.send meth, *args
 7          else
 8            super
 9          end
10        else
11          super
12        end
13      end
14
15      def respond_to?(meth)
16        if meth.to_s =~ /_path$|_url$/
17          if main_app.respond_to? meth
18            true
19          else
20            super
21          end
22        else
23          super
24        end
25      end
26    end
27  end
28
```

/my_timeline/app/presenters/my_timeline/event_presenter.rb

```
1  module MyTimeline
2    class EventPresenter
3
4      attr_accessor :event
5
6      def initialize(event)
7        @event = event
8      end
9
10     def icon_path
11       "my_timeline/icons/#{event.icon_name}"
12     end
13
14     def happened_on
15       time.strftime MyTimeline.time_formatter
16     end
17
18     def id
19       event.id
20     end
21
22   private
23
24     def time
25       return event.happened_on if event.linkable.class.respond_to? :keep_original_time_zone?
26       return event.happened_on unless user.time_zone.present?
27
28       event.happened_on.in_time_zone user.time_zone
29     end
30
31     def user
32       if MyTimeline.user_class == MyTimeline::UserStub
33         MyTimeline::UserStub
34       else
35         event.user
36       end
37     end
38
39     def method_missing(meth, *args, &blk)
40       if event.respond_to?(meth)
41         event.send meth, *args
42       else
43         super
44       end
45     end
46   end
47 end
48
```

/my_timeline/app/controllers/my_timeline/posts_controller.rb

```
1  module MyTimeline
2    class PostsController < MyTimeline::ApplicationController
3
4      def new
5        @event = Event.new
6        @post  = Post.new(event: @event)
7      end
8
9      def create
10       @post  = Post.new(rails4? ? post_params : params[:post])
11
12       @post.event.happened_on = @post.happened_on
13       @post.event.user_id     = @user.id
14       @post.event.icon_name   = "notes.png"
15
16       if @post.save
17         @post.event.linkable = @post
18         @post.event.save
19
20         redirect_to root_path, notice: "Post saved."
21       else
22         render :new
23       end
24     end
25
26     def show
27       @post = Post.find_by_id params[:id]
28       render text: @post.full_text
29     end
30
31   private
32
33     if rails4?
34       define_method :post_params do
35         params.required(:post).permit :happened_on, :full_text, event_attributes: [:description, :public]
36       end
37     end
38   end
39 end
40
```

/my_timeline/app/controllers/my_timeline/events_controller.rb

```ruby
module MyTimeline
  class EventsController < MyTimeline::ApplicationController

    def index
      if @show_hidden
        q = {}
      else
        q = {public: true}
      end

      @events = @user.events.where(q).desc.page params[:page]

      @events_by_day      = @events.all.to_a.group_by { |e| e.happened_on.to_date }
      @dates_with_events = build_dates
    end

    def show
      #
    end

    def edit
      @event = Event.find_by_id params[:id]
    end

    def update
      @event = Event.find_by_id params[:id]
      if @event.update_attributes(rails4? ? event_params : params[:event])
        redirect_to root_path, notice: "Edit successful."
      else
        render 'edit'
      end
    end

    def destroy
      @event = Event.find_by_id(params[:id])
      @event.destroy

      redirect_to root_path
    end

  private

    DateWithEvents = Struct.new(:date, :events)

    def build_dates
      [].tap do |array|
        @events_by_day.each do |date, events|
          array << DateWithEvents.new(date, events.reverse)
        end
      end
    end

    if rails4?
      define_method :event_params do
        params.required(:event).permit :description, :happened_on, :public
      end
    end
  end
end

```

/my_timeline/app/controllers/my_timeline/application_controller.rb

```
1  class MyTimeline::ApplicationController < ApplicationController
2
3    before_filter :find_user
4
5  private
6
7    def find_user
8      @user = MyTimeline.user_class.send "find_by_#{MyTimeline.user_slug}", params[:user_id]
9
10     if @user == current_user
11       @owner_viewing = true
12       @show_hidden   = true
13     else
14       #
15     end
16
17     params.delete :user_id
18   end
19 end
20
```

/my_timeline/app/controllers/my_timeline/control_panel_controller.rb

```
1  module MyTimeline
2    class ControlPanelController < MyTimeline::ApplicationController
3
4      before_filter :user_only
5
6      def index
7        @enabled_plugins = MyTimeline.enabled_plugins
8      end
9
10     def timezone
11       if rails4?
12         @user.time_zone = user_params[:time_zone]
13       else
14         @user.time_zone = params[user_param][:time_zone]
15       end
16
17       @user.save!
18       redirect_to :back, notice: "Time zone setting saved."
19     end
20
21   private
22
23     if rails4?
24       define_method :user_params do
25         params.required(user_param).permit :time_zone
26       end
27     end
28
29     def user_param
30       MyTimeline.user_class.model_name.param_key.to_sym
31     end
32
33     def user_only
34       unless @owner_viewing
35         redirect_to root_path, notice: "Can't see that!"
36       end
37     end
38   end
39 end
40
```

/my_timeline/lib/my_timeline.rb

```
1  require 'kaminari'
2
3  require "my_timeline/user_stub"
4  require "my_timeline/engine"
5
6  module MyTimeline
7
8    mattr_accessor :user_class, :user_slug, :render_method, :table_class, :config_object, :
   enabled_plugins, :time_formatter
9
10   @@user_class = 'MyTimeline::UserStub'
11   def self.user_class
12     @@user_class.constantize
13   end
14
15   @@user_slug = :id
16
17   @@render_method = 'table'
18   @@use_bootstrap = true
19
20   @@table_class = "table table-striped"
21
22   @@time_formatter = "%-l:%M %P - "
23
24   @@enabled_plugins = Set.new
25   def self.register_plugin(plugin_name, options = {})
26     MyTimeline.config_object.key plugin_name, options
27     @@enabled_plugins << plugin_name
28   end
29
30   def self.setup
31     yield self
32   end
33 end
34
```

/my_timeline/lib/tasks/my_timeline_tasks.rake

```
1  # desc "Explaining what the task does"
2  # task :my_timeline do
3  #   # Task goes here
4  # end
5
```

/my_timeline/lib/generators/templates/README

```
1  ===============================================================================
2
3  Please add my_timeline to your routes file:
4
5    1. For a multiple-user instance:
6
7    resources :users do
8      mount MyTimeline::Engine => '/timeline', as: :my_timeline
9    end
10
11   2. For a single timeline:
12
13   mount MyTimeline::Engine => '/timeline', as: :my_timeline
14
15 ===============================================================================
16
```

/my_timeline/lib/generators/templates/my_timeline.rb

```
1  MyTimeline.setup do |config|
2    # The User class to use... Default is "User".
3    # Set to nil to not use per-user timelines,
4    # or put a constant in a string to use that class
5    config.user_class = 'User'
6
7    # By default, looks for the user by id, but if you want to use a name or a slug,
8    # set it here.  I.E., config.user_slug = :nick_name would result in User.find_by_nick_name
9    #config.user_slug = :id
10
11   # How to render the events - in a :table, or in a :list
12   # config.render_method = :table
13
14   # What classes to style the table with
15   # config.table_class = "table table-striped"
16
17   # How to format the time of the event; default looks like "3:33 pm - "
18   # config.time_formatter = %-l:%M %P - "
19 end
20
```

/my_timeline/lib/generators/my_timeline/install_generator.rb

```
1  module MyTimeline
2    module Generators
3      class InstallGenerator < Rails::Generators::Base
4        source_root File.expand_path("../../templates", __FILE__)
5
6        desc "Creates a MyTimeline initializer and copy locale files to your application."
7        class_option :orm
8
9        def copy_initializer
10         template "my_timeline.rb", "config/initializers/my_timeline.rb"
11       end
12
13       def show_readme
14         readme "README" if behavior == :invoke
15       end
16     end
17   end
18 end
19
```

/my_timeline/lib/my_timeline/engine.rb

```
1  require 'my_timeline/settings_ext'
2  require 'my_timeline/core_ext/rails4'
3
4  module MyTimeline
5    class Engine < ::Rails::Engine
6
7      isolate_namespace MyTimeline
8
9      config.autoload_paths << File.expand_path("../../app/classes/**", __FILE__)
10     config.autoload_paths << File.expand_path("../../app/scrapers/**", __FILE__)
11
12     config.generators do |g|
13       g.test_framework      :rspec,        fixture: false
14       g.fixture_replacement :factory_girl, dir: 'spec/factories'
15       g.assets false
16       g.helper false
17     end
18
19     config.after_initialize do |app|
20       MyTimeline::SettingsExt.extend_rails_settings
21     end
22
23     config.to_prepare do |app|
24       MyTimeline::SettingsExt.extend_rails_settings
25     end if Rails.env.development?
26   end
27 end
28
```

/my_timeline/lib/my_timeline/version.rb

```
1  module MyTimeline
2    VERSION = "0.1.0"
3  end
4
```

/my_timeline/lib/my_timeline/user_stub.rb

```ruby
require 'singleton'

module MyTimeline
  class UserStub
    include Singleton

    include ActiveModel::Validations
    include ActiveModel::Conversion
    extend ActiveModel::Naming

    def events
      Event
    end

    def settings(var = :core)
      RailsSettings::SettingObject.find_by_var var
    end

    def id
      nil
    end

    def save!
      true
    end

    def persisted?
      false
    end

    def self.method_missing(meth, *args, &blk)
      if meth.to_s =~ /^find_by/
        UserStub
      else
        instance.send meth, *args, &blk
        # super
      end
    end

    def self.settings_attr_accessor(*args)
      args.each do |method_name|
        eval "
          def self.#{method_name.to_s}
            RailsSettings::SettingObject.find_by_var(:core).send('#{method_name}')
          end
          def self.#{method_name.to_s}=(value)
            RailsSettings::SettingObject.find_by_var(:core).send('#{method_name}=', value)
          end
        "
      end
    end

    settings_attr_accessor :time_zone
  end
end

```

/my_timeline/lib/my_timeline/settings_ext.rb

```
1  require 'rails-settings'
2
3  module MyTimeline
4    module SettingsExt
5      def self.extend_rails_settings
6
7        RailsSettings::SettingObject.class_eval do
8          self.table_name = "my_timeline_settings"
9
10         MyTimeline.config_object = ::RailsSettings::Configuration.new(MyTimeline.user_class) do |s|
11           s.key :core
12         end
13
14         MyTimeline.user_class.class_eval do
15           self.send :include, ::RailsSettings::Base
16           self.send :extend,  ::RailsSettings::Scopes
17
18           def self.settings_attr_accessor(*args)
19             args.each do |method_name|
20               eval "
21                 def #{method_name}
22                   self.settings(:core).send(:#{method_name})
23                 end
24                 def #{method_name}=(value)
25                   self.settings(:core).send(:#{method_name}=, value)
26                 end
27               "
28             end
29           end
30
31           settings_attr_accessor :time_zone
32
33         end unless MyTimeline.user_class == MyTimeline::UserStub
34       end
35     end
36   end
37 end
38
39
```

/my_timeline/lib/my_timeline/core_ext/rails4.rb

```
1  module Kernel
2    define_method :rails4? do
3      Rails::VERSION::MAJOR >= 4
4    end
5  end
6
```

/my_timeline/spec/spec_helper.rb

```
1  ENV['RAILS_ENV'] ||= 'test'
2
3  require File.expand_path "../dummy/config/environment.rb",  __FILE__
4  require 'rspec/rails'
5  require 'factory_girl_rails'
6
7  Rails.backtrace_cleaner.remove_silencers!
8
9  # Load support files
10 Dir["#{File.dirname(__FILE__)}/support/**/*.rb"].each { |f| require f }
11
12 RSpec.configure do |config|
13   config.include MyTimeline::Engine.routes.url_helpers
14   config.before(:each) { @routes = MyTimeline::Engine.routes }
15
16   config.mock_with :rspec
17   config.use_transactional_fixtures = true
18   config.infer_base_class_for_anonymous_controllers = false
19   config.order = "random"
20   config.filter_run focus: true
21   config.run_all_when_everything_filtered = true
22 end
23
```

/my_timeline/spec/dummy/Rakefile

```
1  #!/usr/bin/env rake
2  # Add your own tasks in files placed in lib/tasks ending in .rake,
3  # for example lib/tasks/capistrano.rake, and they will automatically be available to Rake.
4
5  require File.expand_path('../config/application', __FILE__)
6
7  Dummy::Application.load_tasks
8
```

/my_timeline/spec/dummy/config.ru

```
1  # This file is used by Rack-based servers to start the application.
2
3  require ::File.expand_path('../config/environment',  __FILE__)
4  run Dummy::Application
5
```

/my_timeline/spec/dummy/db/schema.rb

```
1  # encoding: UTF-8
2  # This file is auto-generated from the current state of the database. Instead
3  # of editing this file, please use the migrations feature of Active Record to
4  # incrementally modify your database, and then regenerate this schema definition.
5  #
6  # Note that this schema.rb definition is the authoritative source for your
7  # database schema. If you need to create the application database on another
8  # system, you should be using db:schema:load, not running all the migrations
9  # from scratch. The latter is a flawed and unsustainable approach (the more migrations
10 # you'll amass, the slower it'll run and the greater likelihood for issues).
11 #
12 # It's strongly recommended to check this file into your version control system.
13
14 ActiveRecord::Schema.define(:version => 20131103135539) do
15
16   create_table "my_timeline_events", :force => true do |t|
17     t.string   "description"
18     t.datetime "happened_on"
19     t.string   "icon_name"
20     t.string   "external_link"
21     t.string   "original_id"
22     t.boolean  "public",        :default => true
23     t.integer  "importance",    :default => 5
24     t.integer  "user_id"
25     t.integer  "linkable_id"
26     t.string   "linkable_type"
27     t.datetime "created_at",                    :null => false
28     t.datetime "updated_at",                    :null => false
29   end
30
31   create_table "my_timeline_foos", :force => true do |t|
32     t.string   "name"
33     t.datetime "created_at", :null => false
34     t.datetime "updated_at", :null => false
35   end
36
37   create_table "my_timeline_posts", :force => true do |t|
38     t.text     "full_text"
39     t.datetime "happened_on"
40     t.integer  "event_id"
41     t.datetime "created_at",  :null => false
42     t.datetime "updated_at",  :null => false
43   end
44
45   create_table "my_timeline_settings", :force => true do |t|
46     t.string   "var",         :null => false
47     t.text     "value"
48     t.integer  "target_id",   :null => false
49     t.string   "target_type", :null => false
50     t.datetime "created_at",  :null => false
51     t.datetime "updated_at",  :null => false
52   end
53
54   add_index "my_timeline_settings", ["target_type", "target_id", "var"], :name => "index_my_ti
   meline_settings_on_user", :unique => true
55 end
56
```

/my_timeline/spec/dummy/config/boot.rb

```
1  require 'rubygems'
2  gemfile = File.expand_path('../../../../Gemfile', __FILE__)
3
4  if File.exist?(gemfile)
5    ENV['BUNDLE_GEMFILE'] = gemfile
6    require 'bundler'
7    Bundler.setup
8  end
9
10 $:.unshift File.expand_path('../../../../lib', __FILE__)
```

/my_timeline/spec/dummy/config/routes.rb

```
1  Rails.application.routes.draw do
2
3    mount MyTimeline::Engine => "/my_timeline"
4  end
5
```

/my_timeline/spec/dummy/config/database.yml

```
1  test:
2    adapter: sqlite3
3    database: db/test.sqlite3
4    pool: 5
5    timeout: 5000
6  development:
7    adapter: sqlite3
8    database: db/development.sqlite3
9    pool: 5
10   timeout: 5000
11
```

/my_timeline/spec/dummy/config/application.rb

```
require File.expand_path('../boot', __FILE__)

require "active_record/railtie"
require "action_controller/railtie"
require "action_mailer/railtie"
require "sprockets/railtie"

begin
  require "active_resource/railtie"
rescue LoadError
end

Bundler.require(*Rails.groups)
require "my_timeline"

module Dummy
  class Application < Rails::Application
    config.encoding = "utf-8"
    config.filter_parameters += [:password]
    config.active_support.escape_html_entities_in_json = true
    config.active_record.whitelist_attributes = true unless rails4?
    config.assets.enabled = true
    config.assets.version = '1.0'
  end
end


```

/my_timeline/spec/dummy/config/environment.rb

```
1  # Load the rails application
2  require File.expand_path('../application', __FILE__)
3
4  # Initialize the rails application
5  Dummy::Application.initialize!
6
```

/my_timeline/spec/dummy/config/environments/test.rb

```
1  Dummy::Application.configure do
2    # Settings specified here will take precedence over those in config/application.rb
3
4    # The test environment is used exclusively to run your application's
5    # test suite. You never need to work with it otherwise. Remember that
6    # your test database is "scratch space" for the test suite and is wiped
7    # and recreated between test runs. Don't rely on the data there!
8    config.cache_classes = true
9
10   # Configure static asset server for tests with Cache-Control for performance
11   config.serve_static_assets = true
12   config.static_cache_control = "public, max-age=3600"
13
14   # Log error messages when you accidentally call methods on nil
15   config.whiny_nils = true unless rails4?
16
17   # Show full error reports and disable caching
18   config.consider_all_requests_local       = true
19   config.action_controller.perform_caching = false
20
21   # Raise exceptions instead of rendering exception templates
22   config.action_dispatch.show_exceptions = false
23
24   # Disable request forgery protection in test environment
25   config.action_controller.allow_forgery_protection    = false
26
27   # Tell Action Mailer not to deliver emails to the real world.
28   # The :test delivery method accumulates sent emails in the
29   # ActionMailer::Base.deliveries array.
30   config.action_mailer.delivery_method = :test
31
32   # Raise exception on mass assignment protection for Active Record models
33   config.active_record.mass_assignment_sanitizer = :strict unless rails4?
34
35   # Print deprecation notices to the stderr
36   config.active_support.deprecation = :stderr
37 end
38
```

/my_timeline/spec/dummy/config/environments/development.rb

```
1  Dummy::Application.configure do
2    # Settings specified here will take precedence over those in config/application.rb
3
4    # In the development environment your application's code is reloaded on
5    # every request. This slows down response time but is perfect for development
6    # since you don't have to restart the web server when you make code changes.
7    config.cache_classes = false
8
9    # Log error messages when you accidentally call methods on nil.
10   config.whiny_nils = true unless rails4?
11
12   # Show full error reports and disable caching
13   config.consider_all_requests_local       = true
14   config.action_controller.perform_caching = false
15
16   # Don't care if the mailer can't send
17   config.action_mailer.raise_delivery_errors = false
18
19   # Print deprecation notices to the Rails logger
20   config.active_support.deprecation = :log
21
22   # Only use best-standards-support built into browsers
23   config.action_dispatch.best_standards_support = :builtin
24
25   # Raise exception on mass assignment protection for Active Record models
26   config.active_record.mass_assignment_sanitizer = :strict unless rails4?
27
28   # Log the query plan for queries taking more than this (works
29   # with SQLite, MySQL, and PostgreSQL)
30   config.active_record.auto_explain_threshold_in_seconds = 0.5 unless rails4?
31
32   # Do not compress assets
33   config.assets.compress = false
34
35   # Expands the lines which load the assets
36   config.assets.debug = true
37 end
38
```

/my_timeline/spec/dummy/script/rails

```
1  #!/usr/bin/env ruby
2  # This command will automatically be run when you run "rails" with Rails 3 gems installed from the r
   oot of your application.
3
4  APP_PATH = File.expand_path('../../config/application',  __FILE__)
5  require File.expand_path('../../config/boot',  __FILE__)
6  require 'rails/commands'
7
```

/my_timeline/spec/models/my_timeline/event_spec.rb

```
 1  require 'spec_helper'
 2
 3  module MyTimeline
 4    describe Event do
 5      it "is an event" do
 6        e = Event.new
 7        e.save.should be_false
 8
 9        f = Event.new(
10          happened_on: Time.now,
11          description: "foo",
12          original_id: "6"
13        )
14        f.save.should be_true
15
16        e = FactoryGirl.create(:my_timeline_event)
17        e.save.should be_true
18      end
19    end
20  end
21
```

/my_timeline/spec/helpers/my_timeline/application_helper_spec.rb

```
require 'spec_helper'

module MyTimeline
  describe ApplicationHelper do

    class HelperTester
      include MyTimeline::ApplicationHelper

      class MainApp
        def self.bar_path
          "/bar"
        end
      end

      def main_app
        MainApp
      end
    end

    subject { HelperTester.new }

    describe "#method_missing" do

      it "with a non url/path method, herps and dies" do
        expect { subject.foo_and_bar }.to raise_error NoMethodError
      end

      it "with a url/path method not found in the main app, herps and dies" do
        expect { subject.foo_path }.to raise_error NoMethodError
      end

      it "passes a valid url/path method back to the main app" do
        expect(subject.bar_path).to eq "/bar"
      end
    end
  end
end

```

/my_timeline/spec/factories/my_timeline_events.rb

```
1  FactoryGirl.define do
2    factory :my_timeline_event, :class => MyTimeline::Event do
3      happened_on {Time.now - 1.year}
4      description "Foo"
5      original_id "1"
6    end
7  end
8
```

/my_timeline/spec/controllers/my_timeline/posts_controller_spec.rb

```
require 'spec_helper'

describe MyTimeline::PostsController do
  routes { MyTimeline::Engine.routes }

  before { ApplicationController.any_instance.stub :current_user }

  describe "GET #new" do
    it "news" do
      get "new"
      subject.instance_variable_get(:@post).should  be_a MyTimeline::Post
      subject.instance_variable_get(:@event).should be_a MyTimeline::Event
    end
  end

  describe "POST #create" do
    it "creates the event/post" do
      post "create", post: {happened_on: Time.now, full_text: "foo", event_attributes: {description: "b
ar", public: "true"}}
      MyTimeline::Post.last.should_not be_nil
      response.should redirect_to root_path
    end
  end
end

```

/my_timeline/spec/controllers/my_timeline/events_controller_spec.rb

```
require 'spec_helper'

describe MyTimeline::EventsController do
  routes { MyTimeline::Engine.routes }

  before { ApplicationController.any_instance.stub :current_user }

  describe "GET #index" do
    it "gets okay" do
      get "index"
      response.code.should == "200"
    end
  end

  describe "GET #show" do
    xit "shows stuff" do
    end
  end

  describe "GET #edit" do
    xit "edits" do
      get "edit"
    end
  end

  describe "#update" do
    xit "updates" do
    end
  end

  describe "#destroy" do
    xit "kills it"
  end
end

```

/my_timeline/spec/controllers/my_timeline/application_controller_spec.rb

```
1  require 'spec_helper'
2
3  describe MyTimeline::ApplicationController do
4    describe "#find_user" do
5
6      before { subject.params = {user_id: "7"} }
7      after do
8        subject.params.should == {}
9      end
10
11     context "when the user exists" do
12       it "sets some variables" do
13         MyTimeline::UserStub.should_receive(:find_by_id).with("7").and_return "foo"
14         subject.stub(:current_user).and_return "foo"
15         subject.send :find_user
16         subject.instance_variable_get(:@owner_viewing).should be_true
17         subject.instance_variable_get(:@show_hidden).should    be_true
18       end
19     end
20
21     context "when the user is not found" do
22       it "doesn't do much" do
23         subject.stub(:current_user).and_return nil
24         subject.send :find_user
25         subject.instance_variable_get(:@owner_viewing).should be_false
26         subject.instance_variable_get(:@show_hidden).should    be_false
27       end
28     end
29   end
30 end
31
```

/my_timeline/spec/controllers/my_timeline/control_panel_controller_spec.rb

```
 1  require 'spec_helper'
 2
 3  describe MyTimeline::ControlPanelController do
 4    routes { MyTimeline::Engine.routes }
 5
 6    before { ApplicationController.any_instance.stub :current_user }
 7
 8    describe "GET #index" do
 9      it "gets okay" do
10        get "index"
11        response.code.should == "200"
12      end
13    end
14  end
15
```

/my_timeline/config/routes.rb

```
1  MyTimeline::Engine.routes.draw do
2    root to: "events#index"
3
4    resources :events
5    resources :posts
6
7    get  "control_panel" => "control_panel#index",     as: "control_panel"
8    post "control_panel" => "control_panel#timezone", as: "control_panel_timezone"
9  end
10
```

/my_timeline/config/locales/en.yml

```
1  en:
2    my_timeline:
3      timeline_header: A demonstration timeline
4      control_panel:
5        header: Control Panel
6
```

/my_timeline/script/rails

```
1  #!/usr/bin/env ruby
2  # This command will automatically be run when you run "rails" with Rails 3 gems installed from the r
   oot of your application.
3
4  ENGINE_ROOT = File.expand_path('../..', __FILE__)
5  ENGINE_PATH = File.expand_path('../../lib/my_timeline/engine', __FILE__)
6
7  require 'rails/all'
8  require 'rails/engine/commands'
9
```

/my_timeline/gemfiles/Gemfile.rails-3.x

```
1  source "https://rubygems.org"
2
3  gemspec :path => '..'
4
5  gem "jquery-rails"
6  gem "rails", "~>3.2"
7
```

/my_timeline/gemfiles/Gemfile.rails-4.x

```
1  source "https://rubygems.org"
2
3  gemspec :path => '..'
4
5  gem "jquery-rails"
6  gem "rails", ">4.0"
7
```

/my_timeline-demo/README.markdown

```
1  [![Code Climate](https://codeclimate.com/github/JustinAiken/my_timeline-demo.png)](https://
   codeclimate.com/github/JustinAiken/my_timeline-demo)
2
3  # My Timeline-Demo
4  #### A social-media aggregation/display plugin display application
5
6  This is a small demo app to host the [My Timeline](https://github.com/JustinAiken/my_timeline) Rails
7
8  It lets you quickly see what it looks like, including when it's styled with a myriad of [free bootstrap the
   mes](http://bootswatch.com/).
9
10 ### What it looks like:
11 ![Screenshot](doc/screenshot.png)
12
13 ### Usage:
14
15 1. Clone the project (`git clone git@github.com:JustinAiken/my_timeline-demo.git`)
16 2. Bundle the gems (`cd my_timeline-demo && bundle install`)
17 3. Prepare the database (`db:create && db:migrate && db:seed`)
18 4. Put in your own keys in [`config/initializers/my_timeline.rb`](config/initializers/my_timeline.rb)
19 5. Start the rails server (`rails s`)
20 6. Visit [the website](http://127.0.0.1:3000) in a browser
21 7. Login with the user created during the seed step
22 8. ???
23 9. Profit!
24
25 ## Credits
26
27 Original author: [Justin Aiken](https://github.com/JustinAiken)
28
29 ## Links
30
31 * [Source](https://github.com/JustinAiken/my_timeline-demo)
32 * [Bug Tracker](https://github.com/JustinAiken/my_timeline-demo/issues)
33
34 ## Note on Patches/Pull Requests
35
36 * Fork the project.
37 * Make your feature addition or bug fix.
38 * Add tests for it. This is important so I don't break it in a future version unintentionally.
39 * Commit, do not mess with rakefile, version, or history.
40   * If you want to have your own version, that is fine but bump version in a commit by itself so I can ig
   nore when I pull
41 * Send me a pull request. Bonus points for topic branches.
42
43 ## Copyright
44
45 Copyright (c) 2013 Justin Aiken MIT license (see LICENSE for details).
46
```

/my_timeline-demo/LICENSE

```
Copyright 2013 Justin Aiken

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

```

/my_timeline-demo/Gemfile

```
1  source 'https://rubygems.org'
2
3  #Rails Base
4  gem 'rails', '3.2.15'
5  gem 'mysql2'
6  gem 'devise'
7  gem 'thin'
8
9  #My Timeline and My Timeline addons
10 gem 'my_timeline',               path: "/Users/jaiken/projects/timeline/my_timeline"
11 gem 'my_timeline-health_graph', path: "/Users/jaiken/projects/timeline/my_timeline-health_graph
12 gem 'my_timeline-twitter'      , path: "/Users/jaiken/projects/timeline/my_timeline-twitter"
13 gem 'my_timeline-github'       , path: "/Users/jaiken/projects/timeline/my_timeline-github"
14
15
16 #Gems needed by My Timeline Addons
17 gem 'health_graph', git: 'git://github.com/jupp0r/health_graph.git'
18
19 group :assets do
20   gem 'sass-rails',   '~> 3.2.3'
21   gem 'coffee-rails', '~> 3.2.1'
22   gem 'uglifier', '>= 1.0.3'
23   gem 'twitter-bootstrap-rails'
24 end
25
26 group :development do
27   gem 'rails-erd'
28   gem 'quiet_assets'
29   gem 'better_errors'
30   gem 'binding_of_caller'
31 end
32
33 gem 'jquery-rails'
34
35
```

/my_timeline-demo/Rakefile

```
1  #!/usr/bin/env rake
2  # Add your own tasks in files placed in lib/tasks ending in .rake,
3  # for example lib/tasks/capistrano.rake, and they will automatically be available to Rake.
4
5  require File.expand_path('../config/application', __FILE__)
6  TimelineApp::Application.load_tasks
7
```

/my_timeline-demo/config.ru

```
1  # This file is used by Rack-based servers to start the application.
2
3  require ::File.expand_path('../config/environment',  __FILE__)
4  run TimelineApp::Application
5
```

/my_timeline-demo/zeus.json

```
{
  "command": "ruby -rubygems -r./custom_plan -eZeus.go",

  "plan": {
    "boot": {
      "default_bundle": {
        "development_environment": {
          "prerake": {"rake": []},
          "runner":   ["r"],
          "console":  ["c"],
          "server":   ["s"],
          "generate": ["g"],
          "destroy":  ["d"]
        }
      }
    }
  }
}

```

/my_timeline-demo/.gitignore

```
# See http://help.github.com/ignore-files/ for more about ignoring files.
#
# If you find yourself ignoring temporary files generated by your text editor
# or operating system, you probably want to add a global ignore instead:
#   git config --global core.excludesfile ~/.gitignore_global

# Ignore bundler config
/.bundle

# Ignore the default SQLite database.
/db/*.sqlite3

# Ignore all logfiles and tempfiles.
/log/*.log
/tmp

.rake_tasks
.ruby-gemset
.ruby-version
.zeus.sock

```

/my_timeline-demo/Gemfile.lock

```
GIT
  remote: git://github.com/jupp0r/health_graph.git
  revision: aac3be6dc4ec870d0dff7445408982e6174e3992
  specs:
    health_graph (0.5.7)
      faraday (>= 0.7.4)
      faraday_middleware (>= 0.7.8)
      hashie (>= 1.2)
      oauth2 (>= 0.5.2)
      webmock (>= 1.7.6)

PATH
  remote: /Users/jaiken/projects/timeline/my_timeline
  specs:
    my_timeline (0.1.0)
      kaminari
      ledermann-rails-settings
      rails (> 2.0)

PATH
  remote: /Users/jaiken/projects/timeline/my_timeline-github
  specs:
    my_timeline-github (0.1.0)
      my_timeline (>= 0.1.0)
      octokit

PATH
  remote: /Users/jaiken/projects/timeline/my_timeline-health_graph
  specs:
    my_timeline-health_graph (0.1.0)
      health_graph
      my_timeline (>= 0.1.0)

PATH
  remote: /Users/jaiken/projects/timeline/my_timeline-twitter
  specs:
    my_timeline-twitter (0.1.0)
      my_timeline (>= 0.1.0)
      twitter (>= 5.0, < 6.0)

GEM
  remote: https://rubygems.org/
  specs:
    actionmailer (3.2.15)
      actionpack (= 3.2.15)
      mail (~> 2.5.4)
    actionpack (3.2.15)
      activemodel (= 3.2.15)
      activesupport (= 3.2.15)
      builder (~> 3.0.0)
      erubis (~> 2.7.0)
      journey (~> 1.0.4)
      rack (~> 1.4.5)
      rack-cache (~> 1.2)
      rack-test (~> 0.6.1)
      sprockets (~> 2.2.1)
    activemodel (3.2.15)
      activesupport (= 3.2.15)
      builder (~> 3.0.0)
    activerecord (3.2.15)
      activemodel (= 3.2.15)
      activesupport (= 3.2.15)
      arel (~> 3.0.2)
      tzinfo (~> 0.3.29)
    activeresource (3.2.15)
      activemodel (= 3.2.15)
      activesupport (= 3.2.15)
```

/my_timeline-demo/Gemfile.lock

```
    activesupport (3.2.15)
      i18n (~> 0.6, >= 0.6.4)
      multi_json (~> 1.0)
    addressable (2.3.5)
    arel (3.0.2)
    atomic (1.1.14)
    bcrypt-ruby (3.1.2)
    better_errors (1.0.1)
      coderay (>= 1.0.0)
      erubis (>= 2.6.6)
    binding_of_caller (0.7.2)
      debug_inspector (>= 0.0.1)
    buftok (0.2.0)
    builder (3.0.4)
    choice (0.1.6)
    coderay (1.1.0)
    coffee-rails (3.2.2)
      coffee-script (>= 2.2.0)
      railties (~> 3.2.0)
    coffee-script (2.2.0)
      coffee-script-source
      execjs
    coffee-script-source (1.6.3)
    crack (0.4.1)
      safe_yaml (~> 0.9.0)
    daemons (1.1.9)
    debug_inspector (0.0.2)
    descendants_tracker (0.0.3)
    devise (3.1.1)
      bcrypt-ruby (~> 3.0)
      orm_adapter (~> 0.1)
      railties (>= 3.2.6, < 5)
      thread_safe (~> 0.1)
      warden (~> 1.2.3)
    equalizer (0.0.9)
    erubis (2.7.0)
    eventmachine (1.0.3)
    execjs (2.0.2)
    faraday (0.8.8)
      multipart-post (~> 1.2.0)
    faraday_middleware (0.9.0)
      faraday (>= 0.7.4, < 0.9)
    hashie (2.0.5)
    hike (1.2.3)
    http (0.5.0)
      http_parser.rb
    http_parser.rb (0.6.0)
    httpauth (0.2.0)
    i18n (0.6.5)
    journey (1.0.4)
    jquery-rails (3.0.4)
      railties (>= 3.0, < 5.0)
      thor (>= 0.14, < 2.0)
    json (1.8.1)
    jwt (0.1.8)
      multi_json (>= 1.5)
    kaminari (0.15.1)
      actionpack (>= 3.0.0)
      activesupport (>= 3.0.0)
    ledermann-rails-settings (2.2.0)
      activerecord (>= 3.1)
    mail (2.5.4)
      mime-types (~> 1.16)
      treetop (~> 1.4.8)
    memoizable (0.4.0)
      thread_safe (~> 0.1.3)
    mime-types (1.25)
```

/my_timeline-demo/Gemfile.lock

```
135     multi_json (1.8.2)
136     multi_xml (0.5.5)
137     multipart-post (1.2.0)
138     mysql2 (0.3.13)
139     oauth2 (0.9.2)
140       faraday (~> 0.8)
141       httpauth (~> 0.2)
142       jwt (~> 0.1.4)
143       multi_json (~> 1.0)
144       multi_xml (~> 0.5)
145       rack (~> 1.2)
146     octokit (2.7.0)
147       sawyer (~> 0.5.2)
148     orm_adapter (0.4.0)
149     polyglot (0.3.3)
150     quiet_assets (1.0.2)
151       railties (>= 3.1, < 5.0)
152     rack (1.4.5)
153     rack-cache (1.2)
154       rack (>= 0.4)
155     rack-ssl (1.3.3)
156       rack
157     rack-test (0.6.2)
158       rack (>= 1.0)
159     rails (3.2.15)
160       actionmailer (= 3.2.15)
161       actionpack (= 3.2.15)
162       activerecord (= 3.2.15)
163       activeresource (= 3.2.15)
164       activesupport (= 3.2.15)
165       bundler (~> 1.0)
166       railties (= 3.2.15)
167     rails-erd (1.1.0)
168       activerecord (>= 3.0)
169       activesupport (>= 3.0)
170       choice (~> 0.1.6)
171       ruby-graphviz (~> 1.0.4)
172     railties (3.2.15)
173       actionpack (= 3.2.15)
174       activesupport (= 3.2.15)
175       rack-ssl (~> 1.3.2)
176       rake (>= 0.8.7)
177       rdoc (~> 3.4)
178       thor (>= 0.14.6, < 2.0)
179     rake (10.1.0)
180     rdoc (3.12.2)
181       json (~> 1.4)
182     ruby-graphviz (1.0.9)
183     safe_yaml (0.9.7)
184     sass (3.2.12)
185     sass-rails (3.2.6)
186       railties (~> 3.2.0)
187       sass (>= 3.1.10)
188       tilt (~> 1.3)
189     sawyer (0.5.3)
190       addressable (~> 2.3.5)
191       faraday (~> 0.8, < 0.10)
192     simple_oauth (0.2.0)
193     sprockets (2.2.2)
194       hike (~> 1.2)
195       multi_json (~> 1.0)
196       rack (~> 1.0)
197       tilt (~> 1.1, != 1.3.0)
198     thin (1.6.1)
199       daemons (>= 1.0.9)
200       eventmachine (>= 1.0.0)
201       rack (>= 1.0.0)
```

/my_timeline-demo/Gemfile.lock

```
202       thor (0.18.1)
203       thread_safe (0.1.3)
204         atomic
205       tilt (1.4.1)
206       treetop (1.4.15)
207         polyglot
208         polyglot (>= 0.3.1)
209       twitter (5.5.1)
210         addressable (~> 2.3)
211         buftok (~> 0.2.0)
212         descendants_tracker (~> 0.0.3)
213         equalizer (~> 0.0.9)
214         faraday (>= 0.8, < 0.10)
215         http (~> 0.5.0)
216         http_parser.rb (~> 0.6.0)
217         json (~> 1.8)
218         memoizable (~> 0.4.0)
219         simple_oauth (~> 0.2.0)
220       twitter-bootstrap-rails (2.2.8)
221         actionpack (>= 3.1)
222         execjs
223         rails (>= 3.1)
224         railties (>= 3.1)
225       tzinfo (0.3.38)
226       uglifier (2.3.0)
227         execjs (>= 0.3.0)
228         json (>= 1.8.0)
229       warden (1.2.3)
230         rack (>= 1.0)
231       webmock (1.15.2)
232         addressable (>= 2.2.7)
233         crack (>= 0.3.2)
234
235   PLATFORMS
236     ruby
237
238   DEPENDENCIES
239     better_errors
240     binding_of_caller
241     coffee-rails (~> 3.2.1)
242     devise
243     health_graph!
244     jquery-rails
245     my_timeline!
246     my_timeline-github!
247     my_timeline-health_graph!
248     my_timeline-twitter!
249     mysql2
250     quiet_assets
251     rails (= 3.2.15)
252     rails-erd
253     sass-rails (~> 3.2.3)
254     thin
255     twitter-bootstrap-rails
256     uglifier (>= 1.0.3)
257
```

/my_timeline-demo/custom_plan.rb

```
1  require 'zeus/rails'
2
3  class CustomPlan < Zeus::Rails
4
5    # def my_custom_command
6    #   # see https://github.com/burke/zeus/blob/master/docs/ruby/modifying.md
7    # end
8
9  end
10
11 Zeus.plan = CustomPlan.new
12
```

/my_timeline-demo/db/seeds.rb

```
1  def get_input(display_name, default_val)
2    puts "Please enter #{display_name} (or press enter to use #{default_val}):"
3    input = STDIN.gets.chomp
4    input = default_val if input.blank?
5
6    input
7  end
8
9  puts "Creating Sample User....."
10
11 first_name = get_input("first name", "Foo")
12 last_name  = get_input("last name", "Bar")
13 email      = get_input("email", "foo@bar.com")
14 password   = get_input("password", "foobar")
15
16 u = User.create(
17   email:                 email,
18   password:              password,
19   password_confirmation: password,
20   first_name:            first_name,
21   last_name:             last_name
22 )
23
24 puts "User = #{u.inspect}"
25 u.save!
26 puts "Saved!"
27
```

/my_timeline-demo/db/schema.rb

```
1  # encoding: UTF-8
2  # This file is auto-generated from the current state of the database. Instead
3  # of editing this file, please use the migrations feature of Active Record to
4  # incrementally modify your database, and then regenerate this schema definition.
5  #
6  # Note that this schema.rb definition is the authoritative source for your
7  # database schema. If you need to create the application database on another
8  # system, you should be using db:schema:load, not running all the migrations
9  # from scratch. The latter is a flawed and unsustainable approach (the more migrations
10 # you'll amass, the slower it'll run and the greater likelihood for issues).
11 #
12 # It's strongly recommended to check this file into your version control system.
13
14 ActiveRecord::Schema.define(:version => 20131222224425) do
15
16   create_table "my_timeline_events", :force => true do |t|
17     t.text     "description"
18     t.datetime "happened_on"
19     t.string   "icon_name"
20     t.string   "external_link"
21     t.string   "original_id"
22     t.boolean  "public",        :default => true
23     t.integer  "importance",    :default => 5
24     t.integer  "user_id"
25     t.integer  "linkable_id"
26     t.string   "linkable_type"
27     t.datetime "created_at",                    :null => false
28     t.datetime "updated_at",                    :null => false
29   end
30
31   create_table "my_timeline_github_fork_events", :force => true do |t|
32     t.datetime "happened_on"
33     t.string   "original_id"
34     t.string   "repo"
35     t.integer  "event_id"
36     t.datetime "created_at",  :null => false
37     t.datetime "updated_at",  :null => false
38   end
39
40   create_table "my_timeline_health_graph_cardio_activities", :force => true do |t|
41     t.datetime "happened_on"
42     t.float    "meters"
43     t.float    "duration"
44     t.integer  "calories"
45     t.string   "routefile"
46     t.string   "uri"
47     t.text     "notes"
48     t.string   "equipment"
49     t.float    "climb"
50     t.string   "activity_type"
51     t.integer  "event_id"
52     t.datetime "created_at",    :null => false
53     t.datetime "updated_at",    :null => false
54   end
55
56   create_table "my_timeline_posts", :force => true do |t|
57     t.text     "full_text"
58     t.datetime "happened_on"
59     t.integer  "event_id"
60     t.datetime "created_at",  :null => false
61     t.datetime "updated_at",  :null => false
62   end
63
64   create_table "my_timeline_settings", :force => true do |t|
65     t.string   "var",         :null => false
66     t.text     "value"
67     t.integer  "target_id",   :null => false
```

/my_timeline-demo/db/schema.rb

```
68      t.string   "target_type", :null => false
69      t.datetime "created_at",  :null => false
70      t.datetime "updated_at",  :null => false
71    end
72
73    add_index "my_timeline_settings", ["target_type", "target_id", "var"], :name => "index_my_t
    imeline_settings_on_target_type_and_target_id_and_var", :unique => true
74
75    create_table "my_timeline_twitter_tweets", :force => true do |t|
76      t.datetime "happened_on"
77      t.text     "uri"
78      t.text     "post"
79      t.integer  "event_id"
80      t.datetime "created_at",  :null => false
81      t.datetime "updated_at",  :null => false
82    end
83
84    create_table "users", :force => true do |t|
85      t.datetime "created_at",                           :null => false
86      t.datetime "updated_at",                           :null => false
87      t.string   "email",                :default => "", :null => false
88      t.string   "encrypted_password",   :default => "", :null => false
89      t.string   "first_name"
90      t.string   "last_name"
91      t.string   "reset_password_token"
92      t.datetime "reset_password_sent_at"
93      t.datetime "remember_created_at"
94      t.integer  "sign_in_count",        :default => 0,  :null => false
95      t.datetime "current_sign_in_at"
96      t.datetime "last_sign_in_at"
97      t.string   "current_sign_in_ip"
98      t.string   "last_sign_in_ip"
99    end
100
101   add_index "users", ["email"], :name => "index_users_on_email", :unique => true
102   add_index "users", ["reset_password_token"], :name => "index_users_on_reset_password_tok
    en", :unique => true
103
104 end
105
```

/my_timeline-demo/db/migrate/20131117001829_create_users.rb

```
1  class CreateUsers < ActiveRecord::Migration
2    def change
3      create_table :users do |t|
4        t.string :first_name
5        t.string :last_name
6
7        t.timestamps
8      end
9    end
10 end
11
```

/my_timeline-demo/db/migrate/20131117001910_add_devise_to_users.rb

```
1  class AddDeviseToUsers < ActiveRecord::Migration
2    def self.up
3      change_table(:users) do |t|
4        ## Database authenticatable
5        t.string :email,              :null => false, :default => ""
6        t.string :encrypted_password, :null => false, :default => ""
7
8        ## Recoverable
9        t.string   :reset_password_token
10       t.datetime :reset_password_sent_at
11
12       ## Rememberable
13       t.datetime :remember_created_at
14
15       ## Trackable
16       t.integer  :sign_in_count, :default => 0, :null => false
17       t.datetime :current_sign_in_at
18       t.datetime :last_sign_in_at
19       t.string   :current_sign_in_ip
20       t.string   :last_sign_in_ip
21
22       ## Confirmable
23       # t.string   :confirmation_token
24       # t.datetime :confirmed_at
25       # t.datetime :confirmation_sent_at
26       # t.string   :unconfirmed_email # Only if using reconfirmable
27
28       ## Lockable
29       # t.integer  :failed_attempts, :default => 0, :null => false # Only if lock strategy is :
   failed_attempts
30       # t.string   :unlock_token # Only if unlock strategy is :email or :both
31       # t.datetime :locked_at
32
33
34       # Uncomment below if timestamps were not included in your original model.
35       # t.timestamps
36     end
37
38     add_index :users, :email,                :unique => true
39     add_index :users, :reset_password_token, :unique => true
40     # add_index :users, :confirmation_token,   :unique => true
41     # add_index :users, :unlock_token,         :unique => true
42   end
43
44   def self.down
45     # By default, we don't want to make any assumption about how to roll back a migration when your
46     # model already existed. Please edit below which fields you would like to remove in this migration.
47     raise ActiveRecord::IrreversibleMigration
48   end
49 end
50
```

/my_timeline-demo/db/migrate/20131222121619_create_tweets.my_timeline.rb

```
1  class CreateTweets < ActiveRecord::Migration
2    def change
3      create_table :my_timeline_twitter_tweets do |t|
4        t.datetime :happened_on
5
6        t.text :uri
7        t.text :post
8
9        t.references :event
10       t.timestamps
11     end
12   end
13 end
14
```

/my_timeline-demo/db/migrate/20131116235053_create_my_timeline_posts.my_timeline.r

```
 1  # This migration comes from my_timeline (originally 20131103135539)
 2  class CreateMyTimelinePosts < ActiveRecord::Migration
 3    def change
 4      create_table :my_timeline_posts do |t|
 5        t.text      :full_text
 6        t.datetime :happened_on
 7
 8        t.references :event
 9        t.timestamps
10      end
11    end
12  end
13
```

/my_timeline-demo/db/migrate/20131116235051_create_my_timeline_events.my_timeline.

```
1  # This migration comes from my_timeline (originally 20131027171920)
2  class CreateMyTimelineEvents < ActiveRecord::Migration
3    def change
4      create_table :my_timeline_events do |t|
5        t.text :description
6        t.datetime :happened_on
7        t.string :icon_name
8        t.string :external_link
9        t.string :original_id
10       t.boolean :public, default: true
11       t.integer :importance, default: 5
12
13       t.references :user
14       t.references :linkable, :polymorphic => true
15
16       t.timestamps
17     end
18   end
19 end
20
```

/my_timeline-demo/db/migrate/20131116235052_create_my_timeline_settings.my_timelin

```
1  # This migration comes from my_timeline (originally 20131103000200)
2  class CreateMyTimelineSettings < ActiveRecord::Migration
3
4    def change
5      create_table :my_timeline_settings do |t|
6        t.string      :var,     :null => false
7        t.text        :value
8        t.references :target, :null => false, :polymorphic => true
9
10       t.timestamps
11     end
12
13     add_index :my_timeline_settings, [ :target_type, :target_id, :var ], :unique => true
14   end
15 end
16
```

/my_timeline-demo/db/migrate/20131222224425_create_github_fork_events.my_timeline_

```ruby
# This migration comes from my_timeline_github (originally 20131222224040)
class CreateGithubForkEvents < ActiveRecord::Migration
  def change
    create_table :my_timeline_github_fork_events do |t|
      t.datetime :happened_on

      t.string :original_id
      t.string :repo

      t.references :event
      t.timestamps
    end
  end
end

```

/my_timeline-demo/db/migrate/20131116165219_create_health_graph_cardio_activities.r

```
1  class CreateHealthGraphCardioActivities < ActiveRecord::Migration
2    def change
3      create_table :my_timeline_health_graph_cardio_activities do |t|
4        t.datetime :happened_on
5        t.float    :meters
6        t.float    :duration
7        t.integer  :calories
8        t.string   :routefile
9        t.string   :uri
10       t.text     :notes
11       t.string   :equipment
12       t.float    :climb
13       t.string   :activity_type
14
15       t.references :event
16
17       t.timestamps
18     end
19   end
20 end
21
```

/my_timeline-demo/app/views/home/index.html.erb

```
1 Hello!
2
```

/my_timeline-demo/app/views/layouts/application.html.erb

```
1  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/
   TR/xhtml1/DTD/xhtml1-transitional.dtd">
2  <html lang="en">
3    <head>
4      <title> Timeline</title>
5      <%= stylesheet_link_tag "application", media: "all" %>
6      <%= stylesheet_link_tag "bootswatch/#{$current_theme}", media: "all" %>
7      <%= javascript_include_tag "application" %>
8    </head>
9
10   <body>
11     <%= nav_bar brand: "My Timeline Demo-App", responsive: true do %>
12       <%= menu_group pull: :center do %>
13         <%= menu_item "Home", "/" %>
14         <% if current_user %>
15           <%= menu_item "My Timeline", "/users/#{current_user.id}/timeline/" %>
16           <%= menu_item "Settings",    "/users/#{current_user.id}/timeline/control_panel" %>
17         <% end %>
18         <%= menu_divider %>
19         <%= drop_down "Theme" do %>
20           <%= menu_item "Amelia"   , "/home/theme/amelia" %>
21           <%= menu_item "Cerulean" , "/home/theme/cerulean" %>
22           <%= menu_item "Cosmo"    , "/home/theme/cosmo" %>
23           <%= menu_item "Cyborg"   , "/home/theme/cyborg" %>
24           <%= menu_item "Flatly"   , "/home/theme/flatly" %>
25           <%= menu_item "Journal"  , "/home/theme/journal" %>
26           <%= menu_item "Readable" , "/home/theme/readable" %>
27           <%= menu_item "Simplex"  , "/home/theme/simplex" %>
28           <%= menu_item "Slate"    , "/home/theme/slate" %>
29           <%= menu_item "Spacelab" , "/home/theme/spacelab" %>
30           <%= menu_item "Superhero", "/home/theme/superhero" %>
31           <%= menu_item "United"   , "/home/theme/united" %>
32           <%= menu_item "None"     , '/home/theme/nil' %>
33         <% end %>
34
35         <%= menu_item "#{User.first.first_name}'s Timeline", "/users/1/timeline/" %>
36       <% end %>
37       <%= menu_group pull: :right do %>
38         <% if current_user %>
39           <%= menu_item "Log Out", destroy_user_session_path, method: :delete %>
40         <% else %>
41           <%= menu_item "Log In" , new_user_session_path %>
42           <%= menu_item "Sign Up", new_user_registration_path %>
43         <% end %>
44       <% end %>
45     <% end %>
46
47     <div class="container-fluid">
48       <div class="row-fluid">
49         <div class="span10">
50           <%= bootstrap_flash %>
51           <%= yield %>
52         </div>
53       </div>
54       <footer></footer>
55     </div>
56   </body>
57 </html>
58
```

/my_timeline-demo/app/views/kaminari/_gap.html.erb

```
1  <li class="disabled">
2    <%= link_to raw(t 'views.pagination.truncate'), '#' %>
3  </li>
4
```

/my_timeline-demo/app/views/kaminari/_page.html.erb

```
1 <li class="<%= 'active' if page.current? %>">
2   <%= link_to page, page.current? ? '#' : url, {:remote => remote, :rel => page.next? ? 'next' : pa
  ge.prev? ? 'prev' : nil} %>
3 </li>
4
```

/my_timeline-demo/app/views/kaminari/_last_page.html.erb

```
1 <li>
2   <%= link_to_unless current_page.last?, raw(t 'views.pagination.last'), url, {:remote => remote}
3 </li>
4
```

/my_timeline-demo/app/views/kaminari/_next_page.html.erb

```
1 <li>
2   <%= link_to_unless current_page.last?, raw(t 'views.pagination.next'), url, :rel => 'next', :
  remote => remote %>
3 </li>
4
```

/my_timeline-demo/app/views/kaminari/_paginator.html.erb

```
1  <%= paginator.render do -%>
2    <div class="pagination pagination">
3      <ul>
4        <%= first_page_tag unless current_page.first? %>
5        <%= prev_page_tag unless current_page.first? %>
6        <% each_page do |page| -%>
7          <% if page.left_outer? || page.right_outer? || page.inside_window? -%>
8            <%= page_tag page %>
9          <% elsif !page.was_truncated? -%>
10           <%= gap_tag %>
11         <% end -%>
12       <% end -%>
13       <%= next_page_tag unless current_page.last? %>
14       <%= last_page_tag unless current_page.last? %>
15     </ul>
16   </div>
17 <% end -%>
18
```

/my_timeline-demo/app/views/kaminari/_prev_page.html.erb

```
1 <li>
2   <%= link_to_unless current_page.first?, raw(t 'views.pagination.previous'), url, :rel => 'prev', :
  remote => remote %>
3 </li>
4
```

/my_timeline-demo/app/views/kaminari/_first_page.html.erb

```
1 <li>
2   <%= link_to_unless current_page.first?, raw(t 'views.pagination.first'), url, :remote => remote
3 </li>
4
```

/my_timeline-demo/app/assets/javascripts/application.js

```
1  // This is a manifest file that'll be compiled into application.js, which will include all the files
2  // listed below.
3  //
4  // Any JavaScript/Coffee file within this directory, lib/assets/javascripts, vendor/assets/javascripts,
5  // or vendor/assets/javascripts of plugins, if any, can be referenced here using a relative path.
6  //
7  // It's not advisable to add code directly here, but if you do, it'll appear at the bottom of the
8  // the compiled file.
9  //
10 // WARNING: THE FIRST BLANK LINE MARKS THE END OF WHAT'S TO BE PROCESSED, ANY
   BLANK LINE SHOULD
11 // GO AFTER THE REQUIRES BELOW.
12 //
13 //= require jquery
14 //= require jquery_ujs
15 //= require twitter/bootstrap
16 //= require_tree .
17
```

/my_timeline-demo/app/assets/javascripts/bootstrap.js.coffee

```
1  jQuery ->
2    $("a[rel~=popover], .has-popover").popover()
3    $("a[rel~=tooltip], .has-tooltip").tooltip()
4
```

/my_timeline-demo/app/assets/stylesheets/application.css

```
1  /*
2   * This is a manifest file that'll be compiled into application.css, which will include all the files
3   * listed below.
4   *
5   * Any CSS and SCSS file within this directory, lib/assets/stylesheets, vendor/assets/stylesheets,
6   * or vendor/assets/stylesheets of plugins, if any, can be referenced here using a relative path.
7   *
8   * You're free to add application-wide styles to this file and they'll appear at the top of the
9   * compiled file, but it's generally better to create a new file per style scope.
10  *
11  *= require_self
12  *= require_tree .
13  */
14
```

/my_timeline-demo/app/assets/stylesheets/bootstrap_and_overrides.css

```
1  /*
2   =require twitter-bootstrap-static/bootstrap
3
4  require bootswatch/cerulean
5
6  require twitter-bootstrap-static/sprites
7   =require twitter-bootstrap-static/fontawesome
8   */
9
10 select.date-select {
11     width:auto;
12     border: 1px solid #cccccc;
13     background-color: #ffffff;
14 }
15
16 select.datetime_select {
17     width:auto;
18     border: 1px solid #cccccc;
19     background-color: #ffffff;
20 }
21
```

/my_timeline-demo/app/models/user.rb

```
1  class User < ActiveRecord::Base
2    devise :database_authenticatable, :registerable,
3           :recoverable, :rememberable, :trackable, :validatable
4
5    attr_accessible :email, :password, :password_confirmation, :remember_me, :first_name, :
   last_name
6
7    has_many :events, class_name: MyTimeline::Event
8  end
9
```

/my_timeline-demo/app/helpers/application_helper.rb

```
1  module ApplicationHelper
2  end
3
```

/my_timeline-demo/app/controllers/home_controller.rb

```
1  class HomeController < ApplicationController
2    def index
3    end
4
5    def theme
6      $current_theme = params[:theme]
7      redirect_to :back
8    end
9  end
10
```

/my_timeline-demo/app/controllers/application_controller.rb

```
1  class ApplicationController < ActionController::Base
2    # protect_from_forgery
3
4    # skip_before_filter :verify_authenticity_token, :only => [:destroy]
5  end
6
```

/my_timeline-demo/config/boot.rb

```
1  require 'rubygems'
2
3  # Set up gems listed in the Gemfile.
4  ENV['BUNDLE_GEMFILE'] ||= File.expand_path('../../Gemfile', __FILE__)
5
6  require 'bundler/setup' if File.exists?(ENV['BUNDLE_GEMFILE'])
7
```

/my_timeline-demo/config/routes.rb

```
1  TimelineApp::Application.routes.draw do
2    devise_for :users
3
4    root to: "home#index"
5    get "home/theme/:theme", to: "home#theme", as: :change_theme
6
7    resources :users do
8      mount MyTimeline::Engine => '/timeline', as: :my_timeline
9    end
10 end
11
```

/my_timeline-demo/config/database.yml

```
1  test:
2    adapter: mysql2
3    database: timeline_test
4    username: root
5    password:
6  development:
7    adapter: mysql2
8    database: timeline_dev
9    username: root
10   password:
11
```

/my_timeline-demo/config/application.rb

```
require File.expand_path('../boot', __FILE__)

# Pick the frameworks you want:
require "active_record/railtie"
require "action_controller/railtie"
require "action_mailer/railtie"
require "active_resource/railtie"
require "sprockets/railtie"
# require "rails/test_unit/railtie"

if defined?(Bundler)
  # If you precompile assets before deploying to production, use this line
  Bundler.require(*Rails.groups(:assets => %w(development test)))
  # If you want your assets lazily compiled in production, use this line
  # Bundler.require(:default, :assets, Rails.env)
end

module TimelineApp
  class Application < Rails::Application
    # Settings in config/environments/* take precedence over those specified here.
    # Application configuration should go into files in config/initializers
    # -- all .rb files in that directory are automatically loaded.

    # Custom directories with classes and modules you want to be autoloadable.
    # config.autoload_paths += %W(#{config.root}/extras)

    # Only load the plugins named here, in the order given (default is alphabetical).
    # :all can be used as a placeholder for all plugins not explicitly named.
    # config.plugins = [ :exception_notification, :ssl_requirement, :all ]

    # Activate observers that should always be running.
    # config.active_record.observers = :cacher, :garbage_collector, :forum_observer

    # Set Time.zone default to the specified zone and make Active Record auto-convert to this zone.
    # Run "rake -D time" for a list of tasks for finding time zone names. Default is UTC.
    # config.time_zone = 'Central Time (US & Canada)'

    # The default locale is :en and all translations from config/locales/*.rb,yml are auto loaded.
    # config.i18n.load_path += Dir[Rails.root.join('my', 'locales', '*.{rb,yml}').to_s]
    # config.i18n.default_locale = :de

    # Configure the default encoding used in templates for Ruby 1.9.
    config.encoding = "utf-8"

    # Configure sensitive parameters which will be filtered from the log file.
    config.filter_parameters += [:password]

    # Enable escaping HTML in JSON.
    config.active_support.escape_html_entities_in_json = true

    # Use SQL instead of Active Record's schema dumper when creating the database.
    # This is necessary if your schema can't be completely dumped by the schema dumper,
    # like if you have constraints or database-specific column types
    # config.active_record.schema_format = :sql

    # Enforce whitelist mode for mass assignment.
    # This will create an empty whitelist of attributes available for mass-assignment for all models
    # in your app. As such, your models will need to explicitly whitelist or blacklist accessible
    # parameters by using an attr_accessible or attr_protected declaration.
    config.active_record.whitelist_attributes = true

    # Enable the asset pipeline
    config.assets.enabled = true

    # Version of your assets, change this if you want to expire all your assets
    config.assets.version = '1.0'
  end
```

/my_timeline-demo/config/application.rb

```
68 end
69
```

/my_timeline-demo/config/environment.rb

```
1  # Load the rails application
2  require File.expand_path('../application', __FILE__)
3
4  # Initialize the rails application
5  TimelineApp::Application.initialize!
6
```

/my_timeline-demo/config/locales/en.yml

```
1  # Sample localization file for English. Add more files in this directory for other locales.
2  # See https://github.com/svenfuchs/rails-i18n/tree/master/rails%2Flocale for starting points.
3
4  en:
5    hello: "Hello world"
6
```

/my_timeline-demo/config/locales/devise.en.yml

```
# Additional translations at https://github.com/plataformatec/devise/wiki/I18n

en:
  devise:
    confirmations:
      confirmed: "Your account was successfully confirmed."
      confirmed_and_signed_in: "Your account was successfully confirmed. You are now signed in."
      send_instructions: "You will receive an email with instructions about how to confirm your account in a few minutes."
      send_paranoid_instructions: "If your email address exists in our database, you will receive an email with instructions about how to confirm your account in a few minutes."
    failure:
      already_authenticated: "You are already signed in."
      inactive: "Your account is not activated yet."
      invalid: "Invalid email or password."
      invalid_token: "Invalid authentication token."
      locked: "Your account is locked."
      not_found_in_database: "Invalid email or password."
      timeout: "Your session expired. Please sign in again to continue."
      unauthenticated: "You need to sign in or sign up before continuing."
      unconfirmed: "You have to confirm your account before continuing."
    mailer:
      confirmation_instructions:
        subject: "Confirmation instructions"
      reset_password_instructions:
        subject: "Reset password instructions"
      unlock_instructions:
        subject: "Unlock Instructions"
    omniauth_callbacks:
      failure: "Could not authenticate you from %{kind} because \"%{reason}\"."
      success: "Successfully authenticated from %{kind} account."
    passwords:
      no_token: "You can't access this page without coming from a password reset email. If you do come from a password reset email, please make sure you used the full URL provided."
      send_instructions: "You will receive an email with instructions about how to reset your password in a few minutes."
      send_paranoid_instructions: "If your email address exists in our database, you will receive a password recovery link at your email address in a few minutes."
      updated: "Your password was changed successfully. You are now signed in."
      updated_not_active: "Your password was changed successfully."
    registrations:
      destroyed: "Bye! Your account was successfully cancelled. We hope to see you again soon."
      signed_up: "Welcome! You have signed up successfully."
      signed_up_but_inactive: "You have signed up successfully. However, we could not sign you in because your account is not yet activated."
      signed_up_but_locked: "You have signed up successfully. However, we could not sign you in because your account is locked."
      signed_up_but_unconfirmed: "A message with a confirmation link has been sent to your email address. Please open the link to activate your account."
      update_needs_confirmation: "You updated your account successfully, but we need to verify your new email address. Please check your email and click on the confirm link to finalize confirming your new email address."
      updated: "You updated your account successfully."
    sessions:
      signed_in: "Signed in successfully."
      signed_out: "Signed out successfully."
    unlocks:
      send_instructions: "You will receive an email with instructions about how to unlock your account in a few minutes."
      send_paranoid_instructions: "If your account exists, you will receive an email with instructions about how to unlock it in a few minutes."
      unlocked: "Your account has been unlocked successfully. Please sign in to continue."
  errors:
    messages:
      already_confirmed: "was already confirmed, please try signing in"
      confirmation_period_expired: "needs to be confirmed within %{period}, please request a new
```

/my_timeline-demo/config/locales/devise.en.yml

```
54    one"
55        expired: "has expired, please request a new one"
56        not_found: "not found"
57        not_locked: "was not locked"
58        not_saved:
59          one: "1 error prohibited this %{resource} from being saved:"
60          other: "%{count} errors prohibited this %{resource} from being saved:"
61
```

/my_timeline-demo/config/environments/test.rb

```
1  TimelineApp::Application.configure do
2    # Settings specified here will take precedence over those in config/application.rb
3
4    # The test environment is used exclusively to run your application's
5    # test suite. You never need to work with it otherwise. Remember that
6    # your test database is "scratch space" for the test suite and is wiped
7    # and recreated between test runs. Don't rely on the data there!
8    config.cache_classes = true
9
10   # Configure static asset server for tests with Cache-Control for performance
11   config.serve_static_assets = true
12   config.static_cache_control = "public, max-age=3600"
13
14   # Log error messages when you accidentally call methods on nil
15   config.whiny_nils = true
16
17   # Show full error reports and disable caching
18   config.consider_all_requests_local       = true
19   config.action_controller.perform_caching = false
20
21   # Raise exceptions instead of rendering exception templates
22   config.action_dispatch.show_exceptions = false
23
24   # Disable request forgery protection in test environment
25   config.action_controller.allow_forgery_protection    = false
26
27   # Tell Action Mailer not to deliver emails to the real world.
28   # The :test delivery method accumulates sent emails in the
29   # ActionMailer::Base.deliveries array.
30   config.action_mailer.delivery_method = :test
31
32   # Raise exception on mass assignment protection for Active Record models
33   config.active_record.mass_assignment_sanitizer = :strict
34
35   # Print deprecation notices to the stderr
36   config.active_support.deprecation = :stderr
37 end
38
```

/my_timeline-demo/config/environments/production.rb

```
1  TimelineApp::Application.configure do
2    # Settings specified here will take precedence over those in config/application.rb
3
4    # Code is not reloaded between requests
5    config.cache_classes = true
6
7    # Full error reports are disabled and caching is turned on
8    config.consider_all_requests_local       = false
9    config.action_controller.perform_caching = true
10
11   # Disable Rails's static asset server (Apache or nginx will already do this)
12   config.serve_static_assets = false
13
14   # Compress JavaScripts and CSS
15   config.assets.compress = true
16
17   # Don't fallback to assets pipeline if a precompiled asset is missed
18   config.assets.compile = false
19
20   # Generate digests for assets URLs
21   config.assets.digest = true
22
23   # Defaults to nil and saved in location specified by config.assets.prefix
24   # config.assets.manifest = YOUR_PATH
25
26   # Specifies the header that your server uses for sending files
27   # config.action_dispatch.x_sendfile_header = "X-Sendfile" # for apache
28   # config.action_dispatch.x_sendfile_header = 'X-Accel-Redirect' # for nginx
29
30   # Force all access to the app over SSL, use Strict-Transport-Security, and use secure cookies.
31   # config.force_ssl = true
32
33   # See everything in the log (default is :info)
34   # config.log_level = :debug
35
36   # Prepend all log lines with the following tags
37   # config.log_tags = [ :subdomain, :uuid ]
38
39   # Use a different logger for distributed setups
40   # config.logger = ActiveSupport::TaggedLogging.new(SyslogLogger.new)
41
42   # Use a different cache store in production
43   # config.cache_store = :mem_cache_store
44
45   # Enable serving of images, stylesheets, and JavaScripts from an asset server
46   # config.action_controller.asset_host = "http://assets.example.com"
47
48   # Precompile additional assets (application.js, application.css, and all non-JS/CSS are already added
49   # config.assets.precompile += %w( search.js )
50
51   # Disable delivery errors, bad email addresses will be ignored
52   # config.action_mailer.raise_delivery_errors = false
53
54   # Enable threaded mode
55   # config.threadsafe!
56
57   # Enable locale fallbacks for I18n (makes lookups for any locale fall back to
58   # the I18n.default_locale when a translation can not be found)
59   config.i18n.fallbacks = true
60
61   # Send deprecation notices to registered listeners
62   config.active_support.deprecation = :notify
63
64   # Log the query plan for queries taking more than this (works
65   # with SQLite, MySQL, and PostgreSQL)
66   # config.active_record.auto_explain_threshold_in_seconds = 0.5
67 end
```

/my_timeline-demo/config/environments/development.rb

```
1  TimelineApp::Application.configure do
2    # Settings specified here will take precedence over those in config/application.rb
3
4    # In the development environment your application's code is reloaded on
5    # every request. This slows down response time but is perfect for development
6    # since you don't have to restart the web server when you make code changes.
7    config.cache_classes = false
8
9    # Log error messages when you accidentally call methods on nil.
10   config.whiny_nils = true
11
12   # Show full error reports and disable caching
13   config.consider_all_requests_local       = true
14   config.action_controller.perform_caching = false
15
16   # Don't care if the mailer can't send
17   config.action_mailer.raise_delivery_errors = false
18
19   # Print deprecation notices to the Rails logger
20   config.active_support.deprecation = :log
21
22   # Only use best-standards-support built into browsers
23   config.action_dispatch.best_standards_support = :builtin
24
25   # Raise exception on mass assignment protection for Active Record models
26   config.active_record.mass_assignment_sanitizer = :strict
27
28   # Log the query plan for queries taking more than this (works
29   # with SQLite, MySQL, and PostgreSQL)
30   config.active_record.auto_explain_threshold_in_seconds = 0.5
31
32   # Do not compress assets
33   config.assets.compress = false
34
35   # Expands the lines which load the assets
36   config.assets.debug = true
37 end
38
```

/my_timeline-demo/config/initializers/devise.rb

```
1  # Use this hook to configure devise mailer, warden hooks and so forth.
2  # Many of these configuration options can be set straight in your model.
3  Devise.setup do |config|
4    # The secret key used by Devise. Devise uses this key to generate
5    # random tokens. Changing this key will render invalid all existing
6    # confirmation, reset password and unlock tokens in the database.
7    config.secret_key = '23f2f2b73d75958ca7002dbbeaebf913d27373bafbf6d6f75b995bc0396b9b785ed
     1b1dc61daca46e0bdd5f37296fc1484449b40bedc5ae92ad0dcea5c6bd86c'
8
9    # ==> Mailer Configuration
10   # Configure the e-mail address which will be shown in Devise::Mailer,
11   # note that it will be overwritten if you use your own mailer class
12   # with default "from" parameter.
13   config.mailer_sender = 'please-change-me-at-config-initializers-devise@example.com'
14
15   # Configure the class responsible to send e-mails.
16   # config.mailer = 'Devise::Mailer'
17
18   # ==> ORM configuration
19   # Load and configure the ORM. Supports :active_record (default) and
20   # :mongoid (bson_ext recommended) by default. Other ORMs may be
21   # available as additional gems.
22   require 'devise/orm/active_record'
23
24   # ==> Configuration for any authentication mechanism
25   # Configure which keys are used when authenticating a user. The default is
26   # just :email. You can configure it to use [:username, :subdomain], so for
27   # authenticating a user, both parameters are required. Remember that those
28   # parameters are used only when authenticating and not when retrieving from
29   # session. If you need permissions, you should implement that in a before filter.
30   # You can also supply a hash where the value is a boolean determining whether
31   # or not authentication should be aborted when the value is not present.
32   # config.authentication_keys = [ :email ]
33
34   # Configure parameters from the request object used for authentication. Each entry
35   # given should be a request method and it will automatically be passed to the
36   # find_for_authentication method and considered in your model lookup. For instance,
37   # if you set :request_keys to [:subdomain], :subdomain will be used on authentication.
38   # The same considerations mentioned for authentication_keys also apply to request_keys.
39   # config.request_keys = []
40
41   # Configure which authentication keys should be case-insensitive.
42   # These keys will be downcased upon creating or modifying a user and when used
43   # to authenticate or find a user. Default is :email.
44   config.case_insensitive_keys = [ :email ]
45
46   # Configure which authentication keys should have whitespace stripped.
47   # These keys will have whitespace before and after removed upon creating or
48   # modifying a user and when used to authenticate or find a user. Default is :email.
49   config.strip_whitespace_keys = [ :email ]
50
51   # Tell if authentication through request.params is enabled. True by default.
52   # It can be set to an array that will enable params authentication only for the
53   # given strategies, for example, `config.params_authenticatable = [:database]` will
54   # enable it only for database (email + password) authentication.
55   # config.params_authenticatable = true
56
57   # Tell if authentication through HTTP Auth is enabled. False by default.
58   # It can be set to an array that will enable http authentication only for the
59   # given strategies, for example, `config.http_authenticatable = [:token]` will
60   # enable it only for token authentication. The supported strategies are:
61   # :database      = Support basic authentication with authentication key + password
62   # :token         = Support basic authentication with token authentication key
63   # :token_options = Support token authentication with options as defined in
64   #                  http://api.rubyonrails.org/classes/ActionController/HttpAuthentication/Token.html
65   # config.http_authenticatable = false
66
```

/my_timeline-demo/config/initializers/devise.rb

```
67  # If http headers should be returned for AJAX requests. True by default.
68  # config.http_authenticatable_on_xhr = true
69
70  # The realm used in Http Basic Authentication. 'Application' by default.
71  # config.http_authentication_realm = 'Application'
72
73  # It will change confirmation, password recovery and other workflows
74  # to behave the same regardless if the e-mail provided was right or wrong.
75  # Does not affect registerable.
76  # config.paranoid = true
77
78  # By default Devise will store the user in session. You can skip storage for
79  # :http_auth and :token_auth by adding those symbols to the array below.
80  # Notice that if you are skipping storage for all authentication paths, you
81  # may want to disable generating routes to Devise's sessions controller by
82  # passing :skip => :sessions to `devise_for` in your config/routes.rb
83  config.skip_session_storage = [:http_auth]
84
85  # By default, Devise cleans up the CSRF token on authentication to
86  # avoid CSRF token fixation attacks. This means that, when using AJAX
87  # requests for sign in and sign up, you need to get a new CSRF token
88  # from the server. You can disable this option at your own risk.
89  # config.clean_up_csrf_token_on_authentication = true
90
91  # ==> Configuration for :database_authenticatable
92  # For bcrypt, this is the cost for hashing the password and defaults to 10. If
93  # using other encryptors, it sets how many times you want the password re-encrypted.
94  #
95  # Limiting the stretches to just one in testing will increase the performance of
96  # your test suite dramatically. However, it is STRONGLY RECOMMENDED to not use
97  # a value less than 10 in other environments.
98  config.stretches = Rails.env.test? ? 1 : 10
99
100 # Setup a pepper to generate the encrypted password.
101 # config.pepper = '
    036a9b1d2a1161d6a69f39a8c56f3290497bb8315b6f291411fba4efa66b7882d225f7f089d45284c21ef37
    0a35a058e21153a99930d6c314a9a37633ff8193b'
102
103 # ==> Configuration for :confirmable
104 # A period that the user is allowed to access the website even without
105 # confirming his account. For instance, if set to 2.days, the user will be
106 # able to access the website for two days without confirming his account,
107 # access will be blocked just in the third day. Default is 0.days, meaning
108 # the user cannot access the website without confirming his account.
109 # config.allow_unconfirmed_access_for = 2.days
110
111 # A period that the user is allowed to confirm their account before their
112 # token becomes invalid. For example, if set to 3.days, the user can confirm
113 # their account within 3 days after the mail was sent, but on the fourth day
114 # their account can't be confirmed with the token any more.
115 # Default is nil, meaning there is no restriction on how long a user can take
116 # before confirming their account.
117 # config.confirm_within = 3.days
118
119 # If true, requires any email changes to be confirmed (exactly the same way as
120 # initial account confirmation) to be applied. Requires additional unconfirmed_email
121 # db field (see migrations). Until confirmed new email is stored in
122 # unconfirmed email column, and copied to email column on successful confirmation.
123 config.reconfirmable = true
124
125 # Defines which key will be used when confirming an account
126 # config.confirmation_keys = [ :email ]
127
128 # ==> Configuration for :rememberable
129 # The time the user will be remembered without asking for credentials again.
130 # config.remember_for = 2.weeks
131
```

/my_timeline-demo/config/initializers/devise.rb

```
132  # If true, extends the user's remember period when remembered via cookie.
133  # config.extend_remember_period = false
134
135  # Options to be passed to the created cookie. For instance, you can set
136  # :secure => true in order to force SSL only cookies.
137  # config.rememberable_options = {}
138
139  # ==> Configuration for :validatable
140  # Range for password length. Default is 8..128.
141  config.password_length = 4..128
142
143  # Email regex used to validate email formats. It simply asserts that
144  # one (and only one) @ exists in the given string. This is mainly
145  # to give user feedback and not to assert the e-mail validity.
146  # config.email_regexp = /\A[^@]+@[^@]+\z/
147
148  # ==> Configuration for :timeoutable
149  # The time you want to timeout the user session without activity. After this
150  # time the user will be asked for credentials again. Default is 30 minutes.
151  # config.timeout_in = 30.minutes
152
153  # If true, expires auth token on session timeout.
154  # config.expire_auth_token_on_timeout = false
155
156  # ==> Configuration for :lockable
157  # Defines which strategy will be used to lock an account.
158  # :failed_attempts = Locks an account after a number of failed attempts to sign in.
159  # :none            = No lock strategy. You should handle locking by yourself.
160  # config.lock_strategy = :failed_attempts
161
162  # Defines which key will be used when locking and unlocking an account
163  # config.unlock_keys = [ :email ]
164
165  # Defines which strategy will be used to unlock an account.
166  # :email = Sends an unlock link to the user email
167  # :time  = Re-enables login after a certain amount of time (see :unlock_in below)
168  # :both  = Enables both strategies
169  # :none  = No unlock strategy. You should handle unlocking by yourself.
170  # config.unlock_strategy = :both
171
172  # Number of authentication tries before locking an account if lock_strategy
173  # is failed attempts.
174  # config.maximum_attempts = 20
175
176  # Time interval to unlock the account if :time is enabled as unlock_strategy.
177  # config.unlock_in = 1.hour
178
179  # ==> Configuration for :recoverable
180  #
181  # Defines which key will be used when recovering the password for an account
182  # config.reset_password_keys = [ :email ]
183
184  # Time interval you can reset your password with a reset password key.
185  # Don't put a too small interval or your users won't have the time to
186  # change their passwords.
187  config.reset_password_within = 6.hours
188
189  # ==> Configuration for :encryptable
190  # Allow you to use another encryption algorithm besides bcrypt (default). You can use
191  # :sha1, :sha512 or encryptors from others authentication tools as :clearance_sha1,
192  # :authlogic_sha512 (then you should set stretches above to 20 for default behavior)
193  # and :restful_authentication_sha1 (then you should set stretches to 10, and copy
194  # REST_AUTH_SITE_KEY to pepper).
195  #
196  # Require the `devise-encryptable` gem when using anything other than bcrypt
197  # config.encryptor = :sha512
198
```

/my_timeline-demo/config/initializers/devise.rb

```
199   # ==> Configuration for :token_authenticatable
200   # Defines name of the authentication token params key
201   # config.token_authentication_key = :auth_token
202
203   # ==> Scopes configuration
204   # Turn scoped views on. Before rendering "sessions/new", it will first check for
205   # "users/sessions/new". It's turned off by default because it's slower if you
206   # are using only default views.
207   # config.scoped_views = false
208
209   # Configure the default scope given to Warden. By default it's the first
210   # devise role declared in your routes (usually :user).
211   # config.default_scope = :user
212
213   # Set this configuration to false if you want /users/sign_out to sign out
214   # only the current scope. By default, Devise signs out all scopes.
215   # config.sign_out_all_scopes = true
216
217   # ==> Navigation configuration
218   # Lists the formats that should be treated as navigational. Formats like
219   # :html, should redirect to the sign in page when the user does not have
220   # access, but formats like :xml or :json, should return 401.
221   #
222   # If you have any extra navigational formats, like :iphone or :mobile, you
223   # should add them to the navigational formats lists.
224   #
225   # The "*/*" below is required to match Internet Explorer requests.
226   # config.navigational_formats = ['*/*', :html]
227
228   # The default HTTP method used to sign out a resource. Default is :delete.
229   config.sign_out_via = :delete
230
231   # ==> OmniAuth
232   # Add a new OmniAuth provider. Check the wiki for more information on setting
233   # up on your models and hooks.
234   # config.omniauth :github, 'APP_ID', 'APP_SECRET', :scope => 'user,public_repo'
235
236   # ==> Warden configuration
237   # If you want to use other strategies, that are not supported by Devise, or
238   # change the failure app, you can configure them inside the config.warden block.
239   #
240   # config.warden do |manager|
241   #   manager.intercept_401 = false
242   #   manager.default_strategies(:scope => :user).unshift :some_external_strategy
243   # end
244
245   # ==> Mountable engine configurations
246   # When using Devise inside an engine, let's call it `MyEngine`, and this engine
247   # is mountable, there are some extra configurations to be taken into account.
248   # The following options are available, assuming the engine is mounted as:
249   #
250   #     mount MyEngine, at: '/my_engine'
251   #
252   # The router that invoked `devise_for`, in the example above, would be:
253   # config.router_name = :my_engine
254   #
255   # When using omniauth, Devise cannot automatically set Omniauth path,
256   # so you need to do it manually. For the users scope, it would be:
257   # config.omniauth_path_prefix = '/my_engine/users/auth'
258 end
259
```

/my_timeline-demo/config/initializers/mime_types.rb

```
1  # Be sure to restart your server when you modify this file.
2
3  # Add new mime types for use in respond_to blocks:
4  # Mime::Type.register "text/richtext", :rtf
5  # Mime::Type.register_alias "text/html", :iphone
6
```

/my_timeline-demo/config/initializers/inflections.rb

```
1  # Be sure to restart your server when you modify this file.
2
3  # Add new inflection rules using the following format
4  # (all these examples are active by default):
5  # ActiveSupport::Inflector.inflections do |inflect|
6  #   inflect.plural /^(ox)$/i, '\1en'
7  #   inflect.singular /^(ox)en/i, '\1'
8  #   inflect.irregular 'person', 'people'
9  #   inflect.uncountable %w( fish sheep )
10 # end
11 #
12 # These inflection rules are supported but not enabled by default:
13 # ActiveSupport::Inflector.inflections do |inflect|
14 #   inflect.acronym 'RESTful'
15 # end
16
```

/my_timeline-demo/config/initializers/my_timeline.rb

```
1  MyTimeline.setup do |config|
2    # The User class to use... Default is "User".
3    # Set to nil to not use per-user timelines,
4    # or put a constant in a string to use that class
5    config.user_class = 'User'
6
7    # By default, looks for the user by id, but if you want to use a name or a slug,
8    # set it here.  I.E., config.user_slug = :nick_name would result in User.find_by_nick_name
9    #config.user_slug = :id
10
11   # How to render the events - in a :table, or in a :list
12   # config.render_method = :table
13
14   # What classes to style the table with
15   # config.table_class = "table table-striped"
16 end
17
18 MyTimeline::HealthGraph.setup do |config|
19   config.client_id     = "secret"
20   config.client_secret = "secret"
21 end
22
23 MyTimeline::Twitter.setup do |config|
24   config.consumer_key        = "secret"
25   config.consumer_secret     = "secret"
26   config.access_token        = "secret"
27   config.access_token_secret = "secret"
28 end
29
30 MyTimeline::Github.setup do |config|
31   config.client_id     = "secret"
32   config.client_secret = "secret"
33 end
34
```

/my_timeline-demo/config/initializers/secret_token.rb

```
1 # Be sure to restart your server when you modify this file.
2
3 # Your secret key for verifying the integrity of signed cookies.
4 # If you change this key, all old signed cookies will become invalid!
5 # Make sure the secret is at least 30 characters and all random,
6 # no regular words or you'll be exposed to dictionary attacks.
7 TimelineApp::Application.config.secret_token = '18d5d35e89c7c4fa4c959fd5c60196b83a1cb03d0db7
  bfead0092f20c9bd7b6332006d7a1c69c4f6d7d41cd5ddaa274dc75387f9a78e7e01e7304fd4c66114aa'
8
```

/my_timeline-demo/config/initializers/session_store.rb

```
1  # Be sure to restart your server when you modify this file.
2
3  TimelineApp::Application.config.session_store :cookie_store, key: '_timeline_app_session'
4
5  # Use the database for sessions instead of the cookie-based default,
6  # which shouldn't be used to store highly confidential information
7  # (create the session table with "rails generate session_migration")
8  # TimelineApp::Application.config.session_store :active_record_store
9
```

/my_timeline-demo/config/initializers/wrap_parameters.rb

```
1  # Be sure to restart your server when you modify this file.
2  #
3  # This file contains settings for ActionController::ParamsWrapper which
4  # is enabled by default.
5
6  # Enable parameter wrapping for JSON. You can disable this by setting :format to an empty array.
7  ActiveSupport.on_load(:action_controller) do
8    wrap_parameters format: [:json]
9  end
10
11 # Disable root element in JSON by default.
12 ActiveSupport.on_load(:active_record) do
13   self.include_root_in_json = false
14 end
15
```

/my_timeline-demo/config/initializers/backtrace_silencers.rb

```
1 # Be sure to restart your server when you modify this file.
2
3 # You can add backtrace silencers for libraries that you're using but don't wish to see in your backtrace
  s.
4 # Rails.backtrace_cleaner.add_silencer { |line| line =~ /my_noisy_library/ }
5
6 # You can also remove all the silencers if you're trying to debug a problem that might stem from frame
  work code.
7 # Rails.backtrace_cleaner.remove_silencers!
8
```

/my_timeline-demo/public/robots.txt

```
1  # See http://www.robotstxt.org/wc/norobots.html for documentation on how to use the robots.txt file
2  #
3  # To ban all spiders from the entire site uncomment the next two lines:
4  # User-Agent: *
5  # Disallow: /
6
```

/my_timeline-demo/script/rails

```
1  #!/usr/bin/env ruby
2  # This command will automatically be run when you run "rails" with Rails 3 gems installed from the r
   oot of your application.
3
4  APP_PATH = File.expand_path('../../config/application',  __FILE__)
5  require File.expand_path('../../config/boot',  __FILE__)
6  require 'rails/commands'
7
```

/my_timeline-github/README.markdown

```
1  # My Timeline - Github Plugin [![Code Climate](https://codeclimate.com/github/JustinAiken/
   my_timeline-github.png)](https://codeclimate.com/github/JustinAiken/my_timeline-github)
2  #### Github Integration with My Timeline
3
4  ## Requirements:
5
6  - [My Timeline](https://github.com/JustinAiken/my_timeline)
7  - [Octokit](https://github.com/octokit/octokit.rb) gem
8
9  ## Usage:
10
11 1. Add this gem to your Gemfile:
12 `gem 'my_timeline-github'` and `bundle install`
13 2. [Register your application](https://github.com/settings/applications/new) with Github to get `client_id
   ` and `client_secret` keys.
14 3. Edit `config/initializers/my_timeline.rb` to include your Github OAUTH keys:
15
16 ```ruby
17 MyTimeline.setup do |config|
18   ...
19 end
20
21 ...
22
23 MyTimeline::Github.setup do |config|
24   config.client_id     = "YOURKEY"
25   config.client_secret = "YOURKEY"
26 end
27
28 ```
29
30 ## Current State
31
32 Github provides a wide variety of user activities.  This plugin will eventually include them all, but they'
   ll be coming one at a time...
33
34 `[ ]` [CommitCommentEvent](http://developer.github.com/v3/activity/events/types/#
   commitcommentevent) <br>
35 `[ ]` [CreateEvent](http://developer.github.com/v3/activity/events/types/#createevent) <br>
36 `[ ]` [DeleteEvent](http://developer.github.com/v3/activity/events/types/#deleteevent) <br>
37 `[ ]` [DownloadEvent](http://developer.github.com/v3/activity/events/types/#downloadevent) <br>
38 `[ ]` [FollowEvent](http://developer.github.com/v3/activity/events/types/#followevent) <br>
39 `[x]` [ForkEvent](http://developer.github.com/v3/activity/events/types/#forkevent) <br>
40 `[ ]` [ForkApplyEvent](http://developer.github.com/v3/activity/events/types/#forkapplyevent) <br>
41 `[ ]` [GistEvent](http://developer.github.com/v3/activity/events/types/#gistevent) <br>
42 `[ ]` [GollumEvent](http://developer.github.com/v3/activity/events/types/#gollumevent) <br>
43 `[ ]` [IssueCommentEvent](http://developer.github.com/v3/activity/events/types/#issuecommentevent) <b
   r>
44 `[ ]` [IssuesEvent](http://developer.github.com/v3/activity/events/types/#issuesevent) <br>
45 `[ ]` [MemberEvent](http://developer.github.com/v3/activity/events/types/#memberevent) <br>
46 `[ ]` [PublicEvent](http://developer.github.com/v3/activity/events/types/#publicevent) <br>
47 `[X]` [PullRequestEvent](http://developer.github.com/v3/activity/events/types/#pullrequestevent) <br>
48 `[ ]` [PullRequestReviewCommentEvent](http://developer.github.com/v3/activity/events/types/#
   pullrequestreviewcommentevent) <br>
49 `[ ]` [PushEvent](http://developer.github.com/v3/activity/events/types/#pushevent) <br>
50 `[ ]` [ReleaseEvent](http://developer.github.com/v3/activity/events/types/#releaseevent) <br>
51 `[ ]` [StatusEvent](http://developer.github.com/v3/activity/events/types/#statusevent) <br>
52 `[ ]` [TeamAddEvent](http://developer.github.com/v3/activity/events/types/#teamaddevent) <br>
53 `[ ]` [WatchEvent](http://developer.github.com/v3/activity/events/types/#watchevent)
54
55 ## Credits
56
57 Original author: [Justin Aiken](https://github.com/JustinAiken)
58
59 ## Links
60
61 * [Source](https://github.com/JustinAiken/my_timeline-github)
```

/my_timeline-github/README.markdown

```
62  * [Bug Tracker](https://github.com/JustinAiken/my_timeline-github/issues)
63  * [Rubygem](https://rubygems.org/gems/my_timeline-github)
64
65  ## Note on Patches/Pull Requests
66
67  * Fork the project.
68  * Make your feature addition or bug fix.
69  * Add tests for it. This is important so I don't break it in a future version unintentionally.
70  * Commit, do not mess with rakefile, version, or history.
71    * If you want to have your own version, that is fine but bump version in a commit by itself so I can i
    gnore when I pull
72  * Send me a pull request. Bonus points for topic branches.
73
74  ## Copyright
75
76  Copyright (c) 2013 Justin Aiken Inc. MIT license (see LICENSE for details).
77
```

/my_timeline-github/LICENSE

```
Copyright 2013 Justin Aiken

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

```

/my_timeline-github/Gemfile

```
1  source "https://rubygems.org"
2
3  gemspec
4
```

/my_timeline-github/.gitignore

```
1  Gemfile.lock
2  .ruby-version
3  .ruby-gemset
4
```

/my_timeline-github/CHANGELOG.markdown

```
1  # 0.1.0
2    * Register settings through plugin registry
3    * Require my_timeline 0.1.0
4    * Slight refactoring
5
6  # 0.0.2
7    * Added Rails 4 compatibility
8    * Require my_timeline 0.0.4
9
10 # 0.0.1
11   * Initial release
12
```

/my_timeline-github/my_timeline-github.gemspec

```
1  $:.push File.expand_path("../lib", __FILE__)
2
3  require "my_timeline/github/version"
4
5  Gem::Specification.new do |s|
6    s.name        = "my_timeline-github"
7    s.version     = MyTimeline::Github::VERSION
8    s.authors     = ["Justin Aiken"]
9    s.email       = ["60tonangel@gmail.com"]
10   s.homepage    = "https://www.github.com/JustinAiken/my_timeline-github"
11   s.summary     = "Github plugin for MyTimeline"
12   s.description = "Github plugin for MyTimeline"
13   s.license     = 'MIT'
14   s.files       = `git ls-files`.split("\n")
15
16   s.add_runtime_dependency "my_timeline", '>= 0.1.0'
17   s.add_runtime_dependency "octokit"
18 end
19
```

/my_timeline-github/db/migrate/create_github_commits.rb

```
1  class CreateGithubCommits < ActiveRecord::Migration
2    def change
3      create_table :my_timeline_github_commits do |t|
4        t.datetime :happened_on
5
6        t.string :url
7        t.string :sha
8        t.integer :additions
9        t.integer :deletions
10       t.integer :total
11
12       t.timestamps
13     end
14   end
15 end
16
```

/my_timeline-github/db/migrate/create_github_fork_events.rb

```
1  class CreateGithubForkEvents < ActiveRecord::Migration
2    def change
3      create_table :my_timeline_github_fork_events do |t|
4        t.datetime :happened_on
5
6        t.string :original_id
7        t.string :repo
8
9        t.references :event
10       t.timestamps
11     end
12   end
13 end
14
```

/my_timeline-github/db/migrate/create_github_push_events.rb

```
1  class CreateGithubPushEvents < ActiveRecord::Migration
2    def change
3      create_table :my_timeline_github_push_events do |t|
4        t.datetime :happened_on
5
6        t.string  :head
7        t.string  :ref
8        t.integer :size
9
10       t.references :event
11       t.timestamps
12     end
13   end
14 end
15
```

/my_timeline-github/db/migrate/create_github_pull_request_events.rb

```
1  class CreateGithubPullRequestEvents < ActiveRecord::Migration
2    def change
3      create_table :my_timeline_github_pull_request_events do |t|
4        t.datetime :happened_on
5
6        t.text     :title
7        t.text     :body
8        t.string   :url
9        t.integer  :commits
10       t.integer  :additions
11       t.integer  :deletions
12       t.integer  :changed_files
13       t.string   :repo
14
15       t.references :event
16       t.timestamps
17     end
18   end
19 end
20
```

/my_timeline-github/app/views/my_timeline/github/_control_panel.html.erb

```
1  <h3> Github Settings </h3>
2  <% if @user.settings(:github).user_name %>
3    <%= button_to "Scrape activities from #{@user.settings(:github).user_name}!", github_scrape_
   path, method: :get, class: "btn btn-primary" %>
4  <% else %>
5    <%= form_tag new_github_path, method: :post do %>
6      <%= label_tag :user_name %>
7      <%= text_field_tag :user_name %>
8      <br>
9      <%= submit_tag "Save username", class: "btn btn-primary"  %>
10   <% end %>
11 <% end %>
12
```

/my_timeline-github/app/models/my_timeline/github/commit.rb

```
1  module MyTimeline
2    module Github
3      class Commit < ActiveRecord::Base
4        self.table_name = :my_timeline_github_commits
5        belongs_to :push_event #,dependant: :destroy
6
7        attr_protected unless rails4?
8      end
9    end
10 end
11
```

/my_timeline-github/app/models/my_timeline/github/fork_event.rb

```
1  module MyTimeline
2    module Github
3      class ForkEvent < ActiveRecord::Base
4        self.table_name = :my_timeline_github_fork_events
5        belongs_to :event #,dependant: :destroy
6
7        attr_protected unless rails4?
8      end
9    end
10 end
11
```

/my_timeline-github/app/models/my_timeline/github/push_event.rb

```
 1 module MyTimeline
 2   module Github
 3     class PushEvent < ActiveRecord::Base
 4       self.table_name = :my_timeline_github_fork_events
 5       belongs_to :event #,dependant: :destroy
 6
 7       has_many :commits
 8
 9       attr_protected unless rails4?
10     end
11   end
12 end
13
```

/my_timeline-github/app/models/my_timeline/github/pull_request_event.rb

```
1  module MyTimeline
2    module Github
3      class PullRequestEvent < ActiveRecord::Base
4        self.table_name = :my_timeline_github_pull_request_events
5        belongs_to :event #,dependant: :destroy
6
7        attr_protected unless rails4?
8      end
9    end
10 end
11
```

/my_timeline-github/app/scrapers/my_timeline/github/commit_builder.rb

```ruby
module MyTimeline
  module Github
    class CommitBuilder

      attr_accessor :event

      def build_event(event)
        @event = event
      end

      def build(commit)
        @new_commit = MyTimeline::Github::Commit.create(
          event:     event,
          url:       commit.FIX_THIS,
          sha:       commit.FIX_THIS,
          additions: commit.FIX_THIS,
          deletions: commit.FIX_THIS,
          total:     commit.FIX_THIS
        )
      end

    private

      def foo
        #ADD SHIT HERE
      end
    end
  end
end

```

/my_timeline-github/app/scrapers/my_timeline/github/github_builder.rb

```
1  module MyTimeline
2    module Github
3      class GithubBuilder
4
5        attr_reader :user, :github_event
6
7        def initialize(user, github_event)
8          @user         = user
9          @github_event = github_event
10       end
11
12       def build_event
13         #Children should define this!
14       end
15
16     private
17
18       def already_exists_in_db?
19         event = MyTimeline::Event.find_by_original_id(github_event.id)
20         event && event.icon_name =~ /github/
21       end
22
23       def link_user_repo(user_repo)
24         "<a href=https://www.github.com/#{user_repo}>#{user_repo}</a>"
25       end
26     end
27   end
28 end
29
```

/my_timeline-github/app/scrapers/my_timeline/github/github_scraper.rb

```ruby
module MyTimeline
  module Github
    class GithubScraper

      # ALL VALID GITHUB API EVENTS:
      #   CommitCommentEvent CreateEvent    DeleteEvent DownloadEvent    FollowEvent ForkEvent
      #   ForkApplyEvent     GistEvent      GollumEvent IssueCommentEvent IssuesEvent MemberEvent
      #   PublicEvent        PullRequestEvent PullRequestReviewCommentEvent PushEvent  ReleaseEvent
      #   StatusEvent        TeamAddEvent   WatchEvent

      # IMPLEMENTED EVENTS ONLY:
      VALID_EVENTS = %w{ForkEvent PushEvent PullRequestEvent}.freeze

      attr_accessor :user

      def initialize(user)
        @user = user
      end

      def scrape
        @count = 0

        events.each do |event|
          if should_build? event
            @count = @count + 1 if "MyTimeline::Github::#{event.type}Builder".constantize.new(use
r, event).build_event
          end
        end

        @count
      end

    private

      def events
        [].tap do |big_array|
          while items_on_current_page? do
            big_array << @current_page
          end
        end.flatten
      end

      def items_on_current_page?
        @current_page = github.user_public_events(username, page: page_number)
        @current_page.length > 0
      end

      def page_number
        @page_number ||= 0
        @page_number = @page_number + 1
      end

      def username
        user.settings(:github).user_name
      end

      def github
        @github ||= Octokit::Client.new(
          client_id:     MyTimeline::Github.client_id,
          client_secret:  MyTimeline::Github.client_secret,
          auto_traversal: true
        )
      end

      def should_build?(event)
        VALID_EVENTS.include?(event.type) &&
        user #TODO - @user has this option set??
```

/my_timeline-github/app/scrapers/my_timeline/github/github_scraper.rb

```
67         end
68       end
69     end
70   end
71
```

/my_timeline-github/app/scrapers/my_timeline/github/fork_event_builder.rb

```ruby
module MyTimeline
  module Github
    class ForkEventBuilder < GithubBuilder

      def build_event
        return false if already_exists_in_db?

        event.linkable = fork_event
        event.user = user if MyTimeline.user_class
        event.save

        fork_event.event = event
        fork_event.save
      end

    private

      def event
        @event ||= MyTimeline::Event.create(
          happened_on:    github_event.created_at,
          original_id:    github_event.id,
          external_link: "https://www.github.com/#{user.settings(:github).user_name}/#{github_event.repo.name}",
          icon_name:      "github.png",
          importance:     5,
          public:         1,
          description:    "Forked #{link_user_repo(github_event.repo.name)}"
        )
      end

      def fork_event
        @fork_event ||= MyTimeline::Github::ForkEvent.new(
         happened_on: github_event.created_at,
         original_id: github_event.id,
         repo:        github_event.repo.name
        )
      end
    end
  end
end

```

/my_timeline-github/app/scrapers/my_timeline/github/push_event_builder.rb

```ruby
module MyTimeline
  module Github
    class PushEventBuilder < GithubBuilder

      def build_event
        return false if already_exists_in_db?

        event.linkable = push_event
        event.user = user if MyTimeline.user_class
        event.save

        push_event.event = event
        push_event.save

        #FIX THIS:
        github_event.commits.each do |commit|
          commit_builder.build commit
        end
      end

    private

      def event
        @event ||= MyTimeline::Event.create(
          happened_on:   github_event.created_at,
          original_id:   github_event.id,
          external_link: github_event.FIXTHIS,
          icon_name:     "github.png",
          importance:    5,
          public:        1,
          description:   "#{github_event.FIXTHIS}"
        )
      end

      def push_event
        @push_event ||= MyTimeline::Github::PushEvent.new(
          happened_on:  github_event.created_at,
          head:         github_event.FIXTHIS,
          ref:          github_event.FIXTHIS,
          size:         github_event.FIXTHIS
        )
      end

      def commit_builder
        @commit_builder ||= CommitBuilder.new(github_event)
      end
    end
  end
end
```

/my_timeline-github/app/scrapers/my_timeline/github/pull_request_event_builder.rb

```ruby
module MyTimeline
  module Github
    class PullRequestEventBuilder < GithubBuilder

      def build_event
        return false if already_exists_in_db?

        event.linkable = pull_request_event
        event.user     = user if MyTimeline.user_class
        event.save

        pull_request_event.event = event
        pull_request_event.save
      end

    private

      def event
        @event ||= MyTimeline::Event.create(
          happened_on:   happened_on,
          original_id:   github_event.id,
          external_link: github_event.payload.pull_request.rels[:html].href,
          icon_name:     "github.png",
          importance:    5,
          public:        1,
          description:   "#{github_event.payload.action.capitalize} Pull Request #{link_pr} on #{link_user_repo(github_event.repo.name)}: #{github_event.payload.pull_request.title}"
        )
      end

      def pull_request_event
        @pull_request_event ||= MyTimeline::Github::PullRequestEvent.new(
          happened_on:   happened_on,
          title:         github_event.payload.pull_request.title,
          body:          github_event.payload.pull_request.body,
          url:           github_event.payload.pull_request.rels[:html].href,
          commits:       github_event.payload.pull_request.commits,
          additions:     github_event.payload.pull_request.additions,
          deletions:     github_event.payload.pull_request.deletions,
          changed_files: github_event.payload.pull_request.changed_files,
          repo:          github_event.repo.name
        )
      end

      def happened_on
        github_event.created_at
      end

      def link_pr
        "<a href=#{github_event.payload.pull_request.rels[:html].href}>##{github_event.payload.number}</a>"
      end
    end
  end
end
```

/my_timeline-github/app/controllers/my_timeline/github_controller.rb

```
1  module MyTimeline
2    class GithubController < ApplicationController
3      def new
4        @user.settings(:github).user_name = params[:user_name]
5        @user.save!
6        redirect_to control_panel_path, notice: "Github added!"
7      end
8
9      def scrape
10       scrapey = MyTimeline::Github::GithubScraper.new(@user).scrape
11       redirect_to :back, notice: "Added #{scrapey} gits."
12     end
13   end
14 end
15
```

/my_timeline-github/lib/my_timeline-github.rb

```
 1  require 'octokit'
 2
 3  require "my_timeline"
 4  require "my_timeline/github/engine"
 5
 6  module MyTimeline
 7    module Github
 8
 9      mattr_accessor :client_id, :client_secret
10
11      def self.setup
12        yield self
13      end
14    end
15  end
16
```

/my_timeline-github/lib/my_timeline/github/engine.rb

```
1  module MyTimeline
2    module Github
3      class Engine < ::Rails::Engine
4        isolate_namespace MyTimeline::Github
5
6        config.autoload_paths << File.expand_path("../../../app/classes/**", __FILE__)
7        config.autoload_paths << File.expand_path("../../../app/scrapers/**", __FILE__)
8
9        config.generators do |g|
10         g.test_framework      :rspec,        fixture: false
11         g.fixture_replacement :factory_girl, dir: 'spec/factories'
12         g.assets false
13         g.helper false
14       end
15
16       config.after_initialize do |app|
17         MyTimeline.register_plugin :github, defaults: {user_token: nil}
18       end
19
20       config.to_prepare do |app|
21         MyTimeline.register_plugin :github, defaults: {user_token: nil}
22       end
23
24       rake_tasks do
25         load File.expand_path "railties/github_tasks.rake", File.dirname(__FILE__)
26       end
27     end
28   end
29 end
30
```

/my_timeline-github/lib/my_timeline/github/version.rb

```
1  module MyTimeline
2    module Github
3      VERSION = "0.1.0"
4    end
5  end
6
```

/my_timeline-github/lib/my_timeline/github/railties/github_tasks.rake

```
1  namespace "my_timeline-github" do
2
3    def already_copied?(migration_file)
4      `ls db/migrate/*#{migration_file}.my_timeline.rb` != ""
5    end
6
7    def copy_migration(migration_file)
8      return if already_copied? migration_file
9      timestamp = Time.now.strftime("%Y%m%d%H%M%S")
10     source    = File.expand_path "../../../../db/migrate/#{migration_file}.rb", File.dirname(__FILE__)
11     dest      = File.expand_path "db/migrate/#{timestamp}_#{migration_file}.my_timeline.rb"
12     puts "cp #{source} #{dest}"
13     `cp #{source} #{dest}`
14   end
15
16   namespace :install do
17     desc "Copy migrations from my_timeline-github to application"
18     task :migrations do
19       copy_migration "create_github_fork_events"
20       copy_migration "create_github_pull_request_events"
21     end
22   end
23 end
24
```

/my_timeline-github/config/routes.rb

```
1  MyTimeline::Engine.routes.draw do
2    get  'github/scrape' => 'github#scrape', as: "github_scrape"
3    post 'github/new'    => 'github#new',    as: "new_github"
4    resources :github
5  end
6
```

/my_timeline-health_graph/README.markdown

```
[![Code Climate](https://codeclimate.com/github/JustinAiken/my_timeline-health_graph.png)](https://codeclimate.com/github/JustinAiken/my_timeline-health_graph)

# My Timeline - Health Graph Plugin
#### Health Graph Integration with My Timeline

### Requirements:
- [My Timeline](https://github.com/JustinAiken/my_timeline)
- HealthGraph Client API Key/Secret - Get one [here](http://developer.runkeeper.com/healthgraph/registration-authorization)

### Usage:

1. Add this gem to your Gemfile:
`gem 'my_timeline-health_graph'` and `bundle install`
2. Add jupp0r's health_graph library to your Gemfile:
`gem 'health_graph', git: 'git://github.com/jupp0r/health_graph.git'`
3. Edit `config/initializers/my_timeline.rb` to include your API keys:

```ruby
MyTimeline.setup do |config|
  ...
end

MyTimeline::HealthGraph.setup do |config|
  config.client_id     = "lotsofrandomhexchars"
  config.client_secret = "lotsofrandomhexchars"
end
```

Credits

Original author: [Justin Aiken](https://github.com/JustinAiken)

Links

* [Source](https://github.com/JustinAiken/my_timeline-health_graph)
* [Bug Tracker](https://github.com/JustinAiken/my_timeline-health_graph/issues)
* [Rubygem](https://rubygems.org/gems/my_timeline-health_graph)

Note on Patches/Pull Requests

* Fork the project.
* Make your feature addition or bug fix.
* Add tests for it. This is important so I don't break it in a future version unintentionally.
* Commit, do not mess with rakefile, version, or history.
  * If you want to have your own version, that is fine but bump version in a commit by itself so I can ignore when I pull
* Send me a pull request. Bonus points for topic branches.

Copyright

Copyright (c) 2013 Justin Aiken Inc. MIT license (see LICENSE for details).

```

## /my_timeline-health_graph/LICENSE

```
Copyright 2013 Justin Aiken

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

```

**/my_timeline-health_graph/CHANGELOG.md**

```
1 # 0.1.0
2   * Register settings through plugin registry
3   * Require my_timeline 0.1.0
4   * Slight refactoring
5
6 # 0.0.2
7   * Added Rails 4 compatibility
8   * Require my_timeline 0.0.4
9
10 # 0.0.1
11   * Initial release
12
```

/my_timeline-health_graph/Gemfile

```
1  source "https://rubygems.org"
2
3  gem 'health_graph', git: 'git://github.com/jupp0r/health_graph.git'
4  gemspec
5
```

/my_timeline-health_graph/.gitignore

```
1  Gemfile.lock
2  .ruby-version
3  .ruby-gemset
4
```

/my_timeline-health_graph/my_timeline-health_graph.gemspec

```
1  $:.push File.expand_path("../lib", __FILE__)
2
3  require "my_timeline/health_graph/version"
4
5  Gem::Specification.new do |s|
6    s.name        = "my_timeline-health_graph"
7    s.version     = MyTimeline::HealthGraph::VERSION
8    s.authors     = ["Justin Aiken"]
9    s.email       = ["60tonangel@gmail.com"]
10   s.homepage    = "https://www.github.com/JustinAiken/my_timeline-health_graph"
11   s.summary     = "HealthGraph plugin for MyTimeline"
12   s.description = "HealthGraph plugin for MyTimeline"
13
14   s.license     = 'MIT'
15
16   s.files       = `git ls-files`.split("\n")
17
18   s.add_runtime_dependency "my_timeline", '>= 0.1.0'
19   s.add_runtime_dependency "health_graph"
20 end
21
```

/my_timeline-health_graph/db/migrate/20131103010356_create_health_graph_cardio_act

```
1  class CreateHealthGraphCardioActivities < ActiveRecord::Migration
2    def change
3      create_table :my_timeline_health_graph_cardio_activities do |t|
4        t.datetime :happened_on
5        t.float    :meters
6        t.float    :duration
7        t.integer  :calories
8        t.string   :routefile
9        t.string   :uri
10       t.text     :notes
11       t.string   :equipment
12       t.float    :climb
13       t.string   :activity_type
14
15       t.references :event
16
17       t.timestamps
18     end
19   end
20 end
21
```

/my_timeline-health_graph/app/views/my_timeline/health_graph/_control_panel.html.erb

```
1 <h3> Health Graph Settings </h3>
2 <% if @user.settings(:health_graph).user_token %>
3   <%= button_to "Scrape runs!", health_graph_scrape_path, method: :get, class: "btn btn-
  primary" %>
4 <% else %>
5   <%= button_to "Connect to Runkeeper", new_health_graph_path, method: :get, class: "btn
  btn-primary" %>
6 <% end %>
7
```

/my_timeline-health_graph/app/models/my_timeline/health_graph/cardio_activity.rb

```
 1 module MyTimeline
 2   module HealthGraph
 3     class CardioActivity < ActiveRecord::Base
 4       self.table_name = :my_timeline_health_graph_cardio_activities
 5       belongs_to :event, dependent: :destroy
 6
 7       attr_protected unless rails4?
 8
 9       def self.keep_original_time_zone?
10         true
11       end
12     end
13   end
14 end
15
```

/my_timeline-health_graph/app/scrapers/my_timeline/health_graph/health_graph_scrap

```ruby
require_relative 'cardio_activity_builder'

module MyTimeline
  module HealthGraph
    class HealthGraphScraper

      attr_accessor :user, :activities, :health_graph_user, :count

      def initialize(user)
        @user = user

        ::HealthGraph.configure do |config|
          config.client_id                = user.settings(:health_graph).client_id
          config.client_secret            = user.settings(:health_graph).client_secret
          config.authorization_redirect_url = ""
        end
        @health_graph_user = ::HealthGraph::User.new user.settings(:health_graph).user_token
      end

      def scrape(type = :cardio_activities)
        @count      = 0
        @activities = []

        case type
        when :cardio_activities   then scrape_runs
        when :strength_activities then scrape_strength
        end

        "Added #{count} #{type}"
      end

      def scrape_runs
        get_runs_from_rk
        activities.each do |activity|
          @count += 1 if cardio_builder.build_activity(activity)
        end
      end

      def scrape_strength
        #get_strongs_from_rk
        # activities.each do |activity|
        #   @count += 1 if strength_builder.build_activity(activity)
        # end
      end

      def get_runs_from_rk
        @feed      = health_graph_user.fitness_activities
        orig_feed = @feed.dup

        #Loop through the pages to get all activies
        while @feed
          @activities += @feed.items
          @feed = @feed.next_page
        end

        @activities = orig_feed.send :unpack_items, @activities.reverse
      end

      def cardio_builder
        @cardio_builder ||= MyTimeline::HealthGraph::CardioActivityBuilder.new(user, health_graph_
user)
      end

      def strength_builder
        #@strength_builder ||= MyTimeline::HealthGraphScraper::StrengthActivityBuilder.new(user)
      end
    end
```

/my_timeline-health_graph/app/scrapers/my_timeline/health_graph/health_graph_scrap

```
67    end
68  end
69
```

/my_timeline-health_graph/app/scrapers/my_timeline/health_graph/cardio_activity_builc

```ruby
module MyTimeline
  module HealthGraph
    class CardioActivityBuilder

      attr_reader :user, :activity, :summary, :health_graph_user

      def initialize(user, health_graph_user)
        @health_graph_user = health_graph_user
        @user = user
      end

      def build_activity(activity)
        @activity = activity
        return false if already_exists_in_db?

        @summary = activity.summary

        event = MyTimeline::Event.create(
          happened_on:   activity.start_time,
          original_id:   activity.uri[19..26],
          external_link: "#{profile_base_url}/activity/#{activity.uri[19..26]}",
          icon_name:     ICONS[activity.type],
          importance:    5,
          public:        true,
          description:   build_description,
        )

        new_activity = MyTimeline::HealthGraph::CardioActivity.create(
          happened_on:   activity.start_time,
          duration:      activity.duration,
          meters:        activity.total_distance,
          uri:           activity.uri,
          calories:      summary.total_calories,
          climb:         summary.climb,
          equipment:     summary.equipment,
          activity_type: activity.type,
          notes:         summary.notes
        )

        event.linkable = new_activity
        event.user = user if MyTimeline.user_class
        event.save

        new_activity.event = event
        new_activity.save
      end

    private

      def already_exists_in_db?
        MyTimeline::HealthGraph::CardioActivity.find_by_uri(@activity.uri.to_s).present?
      end

      def build_description
        "".tap do |s|
          s << ACTIVITY_TEXT[activity.type]
          s << equip_string
          s << to_miles(activity.total_distance)
          s << " in #{to_time(activity.duration)}"
          s << calories_text
          s << notes
        end
      end

      def calories_text
        summary.total_calories  ? " and burned #{summary.total_calories.to_i} calories" : ""
      end
```

/my_timeline-health_graph/app/scrapers/my_timeline/health_graph/cardio_activity_builc

```
68
69        def notes
70          summary.notes ? " (#{summary.notes})" : ""
71        end
72
73        def to_miles(num)
74          formatted = '%.2f' % (num * 0.000621371192)
75          "#{formatted} miles"
76        end
77
78        def to_time(num)
79          seconds = num % 60
80          minutes = (num / 60) % 60
81          hours   = num / (60 * 60)
82
83          format "%02d:%02d:%02d", hours, minutes, seconds
84        end
85
86        ICONS = {
87          "Running"     => "run.png",
88          "Cycling"     => "cycle.png",
89          "Walking"     => "walk.png",
90          "Elliptical"  => "elliptical.png"
91        }
92
93        ACTIVITY_TEXT = {
94          "Running"     =>  "Ran ",
95          "Cycling"     =>  "Cycled ",
96          "Walking"     =>  "Strolled ",
97          "Elliptical"  =>  "Ran  "
98        }
99
100       def equip_string
101         summary.equipment == "None" ? "" : "(on a #{summary.equipment.downcase}) "
102       end
103
104       def profile_base_url
105         health_graph_user.profile.profile
106       end
107     end
108   end
109 end
110
```

/my_timeline-health_graph/app/controllers/my_timeline/health_graph_controller.rb

```
1  module MyTimeline
2    class HealthGraphController < MyTimeline::ApplicationController
3
4      def new
5        ::HealthGraph.configure do |config|
6          config.client_id            = MyTimeline::HealthGraph.client_id
7          config.client_secret        = MyTimeline::HealthGraph.client_secret
8          config.authorization_redirect_url = "#{root_url}/health_graph/code"
9        end
10
11       redirect_to ::HealthGraph.authorize_url
12     end
13
14     def show
15       access_token = ::HealthGraph.access_token(params[:code])
16
17       @user.settings(:health_graph).user_token = access_token
18       @user.save!
19       redirect_to control_panel_path, notice: "Connection Successful!"
20     end
21
22     def scrape
23       scrape_status = MyTimeline::HealthGraph::HealthGraphScraper.new(@user).scrape
24       redirect_to :back, notice: scrape_status
25     end
26   end
27 end
28
```

/my_timeline-health_graph/lib/my_timeline-health_graph.rb

```
 1  require 'health_graph'
 2
 3  require "my_timeline"
 4  require "my_timeline/health_graph/engine"
 5
 6  module MyTimeline
 7    module HealthGraph
 8
 9      mattr_accessor :client_id, :client_secret
10
11      def self.setup
12        yield self
13      end
14    end
15  end
16
```

/my_timeline-health_graph/lib/my_timeline/health_graph/engine.rb

```
1  module MyTimeline
2    module HealthGraph
3      class Engine < ::Rails::Engine
4        isolate_namespace MyTimeline::HealthGraph
5
6        config.autoload_paths << File.expand_path("../../../app/classes/**", __FILE__)
7        config.autoload_paths << File.expand_path("../../../app/scrapers/**", __FILE__)
8
9        config.generators do |g|
10         g.test_framework      :rspec,        fixture: false
11         g.fixture_replacement :factory_girl, dir: 'spec/factories'
12         g.assets false
13         g.helper false
14       end
15
16       config.after_initialize do |app|
17         MyTimeline.register_plugin :health_graph, defaults: {user_token: nil}
18       end
19
20       config.to_prepare do |app|
21         MyTimeline.register_plugin :health_graph, defaults: {user_token: nil}
22       end
23
24       rake_tasks do
25         load File::expand_path "railties/health_graph_tasks.rake", File.dirname(__FILE__)
26       end
27     end
28   end
29 end
30
```

/my_timeline-health_graph/lib/my_timeline/health_graph/version.rb

```
1  module MyTimeline
2    module HealthGraph
3      VERSION = "0.1.0"
4    end
5  end
6
```

/my_timeline-health_graph/lib/my_timeline/health_graph/railties/health_graph_tasks.rak

```
1  namespace "my_timeline-health_graph" do
2    namespace :install do
3      desc "Copy migrations from my_timeline-health_graph to application"
4      task :migrations do
5        timestamp = Time.now.strftime("%Y%m%d%H%M%S")
6
7        source    = File.expand_path "../../../../db/migrate/
   20131103010356_create_health_graph_cardio_activities.rb", File.dirname(__FILE__)
8        dest      = File.expand_path "db/migrate/#{timestamp}_create_health_graph_cardio_activities.
   my_timeline.rb"
9        puts "cp #{source} #{dest}"
10       `cp #{source} #{dest}`
11     end
12   end
13 end
14
```

/my_timeline-health_graph/config/routes.rb

```
1  MyTimeline::Engine.routes.draw do
2    get 'health_graph/scrape' => 'health_graph#scrape', as: "health_graph_scrape"
3    resources :health_graph
4  end
5
```

/my_timeline-twitter/README.markdown

```
[![Code Climate](https://codeclimate.com/github/JustinAiken/my_timeline-twitter.png)](https://codeclimate.com/github/JustinAiken/my_timeline-twitter)

# My Timeline - Twitter Plugin
#### Twitter Integration with My Timeline

### Requirements:
- [My Timeline](https://github.com/JustinAiken/my_timeline)
- [Twitter gem](https://github.com/sferik/twitter)

### Usage:

1. Add this gem to your Gemfile:
`gem 'my_timeline-twitter'` and `bundle install`
2. Edit `config/initializers/my_timeline.rb` to include your API keys:

```ruby
MyTimeline.setup do |config|
  ...
end

MyTimeline::Twitter.setup do |config|
  config.client_id            = "lotsofrandomhexchars"
  config.client_secret        = "lotsofrandomhexchars"
  config.access_token         = "lotsofrandomhexchars"
  config.access_token_secret = "lotsofrandomhexchars"
end
```

Credits

Original author: [Justin Aiken](https://github.com/JustinAiken)

Links

* [Source](https://github.com/JustinAiken/my_timeline-twitter)
* [Bug Tracker](https://github.com/JustinAiken/my_timeline-twitter/issues)
* [Rubygem](https://rubygems.org/gems/my_timeline-twitter)

Note on Patches/Pull Requests

* Fork the project.
* Make your feature addition or bug fix.
* Add tests for it. This is important so I don't break it in a future version unintentionally.
* Commit, do not mess with rakefile, version, or history.
  * If you want to have your own version, that is fine but bump version in a commit by itself so I can ignore when I pull
* Send me a pull request. Bonus points for topic branches.

Copyright

Copyright (c) 2013 Justin Aiken Inc. MIT license (see LICENSE for details).

```

## /my_timeline-twitter/LICENSE

```
Copyright 2013 Justin Aiken

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

```

**/my_timeline-twitter/CHANGELOG.md**

```
1 # 0.1.0
2   * Register settings through plugin registry
3   * Require my_timeline 0.1.0
4   * Slight refactoring
5
6 # 0.0.2
7   * Added Rails 4 compatibility
8   * Require my_timeline 0.0.4
9
10 # 0.0.1
11   * Initial release
12
```

/my_timeline-twitter/Gemfile

```
1  source "https://rubygems.org"
2
3  gemspec
4
```

/my_timeline-twitter/.gitignore

```
1 Gemfile.lock
2 .ruby-version
3 .ruby-gemset
4
```

/my_timeline-twitter/my_timeline-twitter.gemspec

```
1  $:.push File.expand_path("../lib", __FILE__)
2
3  require "my_timeline/twitter/version"
4
5  Gem::Specification.new do |s|
6    s.name        = "my_timeline-twitter"
7    s.version     = MyTimeline::Twitter::VERSION
8    s.authors     = ["Justin Aiken"]
9    s.email       = ["60tonangel@gmail.com"]
10   s.homepage    = "https://www.github.com/JustinAiken/my_timeline-twitter"
11   s.summary     = "Twitter plugin for MyTimeline"
12   s.description = "Twitter plugin for MyTimeline"
13   s.license     = 'MIT'
14   s.files       = `git ls-files`.split("\n")
15
16   s.add_runtime_dependency "my_timeline", '>= 0.1.0'
17   s.add_runtime_dependency "twitter", ['>= 5.0', '< 6.0']
18 end
19
```

/my_timeline-twitter/db/migrate/20131107025006_create_tweets.rb

```
1  class CreateTweets < ActiveRecord::Migration
2    def change
3      create_table :my_timeline_twitter_tweets do |t|
4        t.datetime :happened_on
5
6        t.text :uri
7        t.text :post
8
9        t.references :event
10       t.timestamps
11     end
12   end
13 end
14
```

/my_timeline-twitter/app/views/my_timeline/twitter/_control_panel.html.erb

```
1  <h3> Twitter Settings </h3>
2  <% if @user.settings(:twitter).user_name %>
3    <%= button_to "Scrape tweets from #{@user.settings(:twitter).user_name}!", twitter_scrape_pat
   h, method: :get, class: "btn btn-primary"  %>
4  <% else %>
5    <%= form_tag new_twitter_path, method: :post do %>
6      <%= label_tag :user_name %>
7      <%= text_field_tag :user_name %>
8      <br>
9      <%= submit_tag "Save username", class: "btn btn-primary"   %>
10   <% end %>
11 <% end %>
12
```

/my_timeline-twitter/app/models/my_timeline/twitter/tweet.rb

```
1  module MyTimeline
2    module Twitter
3      class Tweet < ActiveRecord::Base
4        self.table_name = :my_timeline_twitter_tweets
5        belongs_to :event #,dependant: :destroy
6
7        attr_protected unless rails4?
8      end
9    end
10 end
11
```

/my_timeline-twitter/app/scrapers/my_timeline/twitter/tweet_builder.rb

```ruby
module MyTimeline
  module Twitter
    class TweetBuilder

      attr_reader :user, :tweet_hash

      def initialize(user)
        @user = user
      end

      def build_tweet(tweet_hash)
        @tweet_hash = tweet_hash
        return false if already_exists_in_db?

        event.linkable = tweet
        event.user = user if MyTimeline.user_class
        event.save

        tweet.event = event
        tweet.save
      end

    private

      def event
        @event ||= MyTimeline::Event.create(
          happened_on:    tweet_hash.created_at,
          original_id:    tweet_hash.id,
          external_link:  "http://twitter.com/#{user.settings(:twitter).user_name}/status/#{tweet_hash.id}",
          icon_name:      "tweetweet_hash.png",
          importance:     5,
          public:         1,
          description:    linkup_mentions_and_hashtags
        )
      end

      def tweet
        @tweet ||= MyTimeline::Twitter::Tweet.new(
         happened_on: tweet_hash.created_at,
         uri:          tweet_hash.id,
         post:         tweet_hash.text
        )
      end

      def already_exists_in_db?
        MyTimeline::Twitter::Tweet.find_by_uri tweet_hash.id
      end

      def linkup_mentions_and_hashtags
        text = tweet_hash.text.dup
        text.gsub!(/@([\w]+)(\W)?/, '<a href="http://twitter.com/\1">@\1</a>\2')
        text.gsub(/#([\w]+)(\W)?/, '<a href="http://twitter.com/search?q=%23\1">#\1</a>\2')
        %Q{"#{text}"}
      end
    end
  end
end

```

/my_timeline-twitter/app/scrapers/my_timeline/twitter/tweet_scraper.rb

```ruby
require_relative 'tweet_builder'
require 'twitter'

module MyTimeline
  module Twitter
    class TweetScraper

      attr_accessor :user, :tweets

      def initialize(user)
        @user = user
      end

      def scrape
        load_tweets
        @count = 0

        tweets.each do |tweet_hash|
          @count = @count + 1 if builder.build_tweet(tweet_hash)
        end

        @count
      end

      def load_tweets
        @tweets = twitter.user_timeline(@user.settings(:twitter).user_name, count: 200)
      end

      def builder
        @builder ||= MyTimeline::Twitter::TweetBuilder.new(user)
      end

      def twitter
        @twitter ||= ::Twitter::REST::Client.new do |config|
          config.consumer_key        = MyTimeline::Twitter.consumer_key
          config.consumer_secret     = MyTimeline::Twitter.consumer_secret
          config.access_token        = MyTimeline::Twitter.access_token
          config.access_token_secret = MyTimeline::Twitter.access_token_secret
        end
      end
    end
  end
end

```

/my_timeline-twitter/app/controllers/my_timeline/twitter_controller.rb

```
1  module MyTimeline
2    class TwitterController < ApplicationController
3
4      def new
5        @user.settings(:twitter).user_name = params[:user_name]
6        @user.save!
7        redirect_to control_panel_path, notice: "Twitter added!"
8      end
9
10     def scrape
11       scrapey = MyTimeline::Twitter::TweetScraper.new(@user).scrape
12       redirect_to :back, notice: "Added #{scrapey} tweets."
13     end
14   end
15 end
16
```

/my_timeline-twitter/lib/my_timeline-twitter.rb

```
 1  require 'twitter'
 2
 3  require "my_timeline"
 4  require "my_timeline/twitter/engine"
 5
 6  module MyTimeline
 7    module Twitter
 8
 9      mattr_accessor :consumer_key, :consumer_secret, :access_token, :access_token_secret
10
11      def self.setup
12        yield self
13      end
14    end
15  end
16
```

/my_timeline-twitter/lib/my_timeline/twitter/engine.rb

```
1  module MyTimeline
2    module Twitter
3      class Engine < ::Rails::Engine
4        isolate_namespace MyTimeline::Twitter
5
6        config.autoload_paths << File.expand_path("../../../app/classes/**", __FILE__)
7        config.autoload_paths << File.expand_path("../../../app/scrapers/**", __FILE__)
8
9        config.generators do |g|
10         g.test_framework      :rspec,        fixture: false
11         g.fixture_replacement :factory_girl, dir: 'spec/factories'
12         g.assets false
13         g.helper false
14       end
15
16       config.after_initialize do |app|
17         MyTimeline.register_plugin :twitter, defaults: {user_token: nil}
18       end
19
20       config.to_prepare do |app|
21         MyTimeline.register_plugin :twitter, defaults: {user_token: nil}
22       end
23
24       rake_tasks do
25         load File::expand_path "railties/twitter_tasks.rake", File.dirname(__FILE__)
26       end
27     end
28   end
29 end
30
```

/my_timeline-twitter/lib/my_timeline/twitter/version.rb

```
1  module MyTimeline
2    module Twitter
3      VERSION = "0.1.0"
4    end
5  end
6
```

/my_timeline-twitter/lib/my_timeline/twitter/railties/twitter_tasks.rake

```
1  namespace "my_timeline-twitter" do
2    namespace :install do
3      desc "Copy migrations from my_timeline-twitter to application"
4      task :migrations do
5        timestamp = Time.now.strftime("%Y%m%d%H%M%S")
6
7        source    = File.expand_path "../../../../db/migrate/20131107025006_create_tweets.rb", File.dirna
   me(__FILE__)
8        dest      = File.expand_path "db/migrate/#{timestamp}_create_tweets.my_timeline.rb"
9        puts "cp #{source} #{dest}"
10       `cp #{source} #{dest}`
11     end
12   end
13 end
14
```

/my_timeline-twitter/config/routes.rb

```
1  MyTimeline::Engine.routes.draw do
2    get 'twitter/scrape'   => 'twitter#scrape', as: "twitter_scrape"
3    post 'twitter/new'    => 'twitter#new',    as: "new_twitter"
4    resources :twitter
5  end
6
```