

API and SPA with Backbone Router Lab

You now know the basics of how to build RESTful JSON APIs with Rails, building a single page app with javascript, html, css, and jquery, and how to use the Backbone router to implement state management for your single page app.

It's time to practice your skills.

- Use:
 - Rails to build an API
 - Backbone.js for your frontend router
 - jQuery and Handlebars
- Start on this project in pairs on Friday
 - On Friday, plan out how you'll accomplish this
 - Figure out how to do the auth flow with your partner
- Complete the projects individually over the weekend

Build a simplified medium clone backed by a RESTful JSON API

~~User signup (/#signup)~~

User attributes:

- first
- last
- email
- password
- After signing up, the user should be assigned an auth token, which should be returned by the POST /users endpoint and stored in the user's session storage under the

key `auth_token`. See [info about session storage](#)

- If a user attempts to sign up with an email that already exists in the system, their token should be regenerated, and the new token should be set in the user's `sessionStorage` under the key `auth_token`.
- The `auth_token` is to be used on all subsequent requests to the API, and passed via the `Authorization` header. See [this stackoverflow post](#). You'll obviously have to change the header name, as per the lesson earlier this week.
- After a successful signup, they should be taken to `/#posts`.

~~User login (`/#login`)~~

- To sign in, the user needs to provide his email and password. Upon successfully signing in, a new `auth_token` should be generated, returned, and stored in the `sessionStorage`

~~User should be able to write a blog post (`/#posts/new`)~~

- The new post form should have a button `Publish` to allow the writer to make the post public.
- The new post form should have a button `Save to Drafts` to allow the writer to save the post to their drafts.

All blog posts should be viewable via (`/#posts`)

- Posts that are public (published), should be viewable without logging in, by anyone.
- Posts that are not public, should only be viewable to the writer, and should have a `Publish` button next to them to make them public.
- After the `Publish` button is clicked, that post's `Publish` button should be removed, and the post should be viewable by anyone else.

~~A single blog post should be viewable via (`/#posts/:id`)~~

- If the post is public, anyone should be able to view it.
- Posts that are not public, should not be viewable by anyone, but the author.

- Non-public posts, should be publishable by the author.
- When an author publishes an individual post, the author should be taken to the /#posts route.

Bonuses

Allow authors to unpublish posts

Allow non-authors to add comments to posts