25.0 hours

Project 7

Build a Twitter Interface

Why can’t I submit this project yet?

You must first complete all prior Techdegree content before you can submit this project. Additionally, you cannot submit more than one project at a time.

Submit for Peer Review

Your current activity is HTML Tables.

- Instructions
- How you'll be graded

Many existing services like Twitter, Facebook, Google Maps, Paypal, and Github allow you to leverage their code in your own applications by talking to their APIs. Using Node to connect to 3rd party APIs provides much greater control than using just AJAX. Server-side communication with APIs offers many more options and greater access to data, but it also require stricter authentication methods.

In this project, you'll use Twitter’s REST API to access your Twitter profile information and render it to a user. The page should automatically authenticate access to your Twitter profile. It should use this access to populate three columns on your page:

- Your 5 most recent tweets.
- Your 5 most recent friends.
- Your 5 most recent private messages.

The ability to look at a layout and see the data behind it is an essential skill for full-stack developers. With the provided HTML and CSS starter files, you'll be able to see what the final result should look like. Replace the example data from the HTML file with your own information, which you'll grab directly from Twitter's API.

Before you start

To prepare for this project you'll need to make sure you complete and understand these steps.

3 steps

- Download the sample HTML and CSS files to see how the page should look after rendering your Twitter profile information.

- Study the Twitter REST API docs to find which methods will provide the information you need to fill out the templates, based on the sample layout. The docs are linked in the project resources.

- Read the tutorial linked in the project resources about how to create a Twitter application.

Project Instructions

To complete this project, follow the instructions below. If you get stuck, ask a question in the community.

7 steps

- **Set up a new Express project in the provided app.js file.**

  You will need to create the following files:

  - A package.json file that includes your project's dependencies.
  - A Jade/Pug template file to display tweets and messages
  - A config.js file that will contain your application's authentication code
  - A .gitignore file to add your config. js and node_modules/ folder to. This will prevent these files from being committed and pushed to github

  The following files are provided:

  - An app.js file where you will set up your Express app and write your API calls
  - An index.html file that provides an example of what the finished project should look like, and some CSS and image files to use with your Jade/Pug template

- **Create a new Twitter application. This will generate the keys and access tokens you need to authenticate your application so it can communicate with the Twitter API. You can find a link to a tutorial on how to do this in the project resources. Please note that while the tutorial says to create a Twitter dev account at dev.twitter.com, the url to create a Twitter dev account is now https://apps.twitter.com/**

- **To use and interact with the Twitter API, you'll need to set up a way to give the Twitter API the set of keys and access tokens that were generated when you create your Twitter app. It's a good idea to use an npm module to help you with this part. For this project, you'll use an npm module called Twit. You can find a link in the project resources. Be sure to look through the documentation and familiarize yourself with how it works.**

  - Create a file called `config.js`. In this file, you'll assign an object literal to the module.exports object, as shown in the Twit documentation. The object literal should have the following properties with their corresponding values from your Twitter application account:
    - consumer_key
    - consumer_secret
    - access_token
    - access_token_secret
  - Import this code into your app.js file to authenticate your application so you can request data from the Twitter API. The config.js file *must* be listed in the .gitignore file so it won't be committed to your github repository. This will prevent your keys and tokens from getting posted publicly to GitHub. It is very important that you do NOT upload any of your personal API keys / secrets / passwords to Github or other publicly accessible place.

  When your project is reviewed, the project reviewer will use their own config file.

- **Make a Pug/Jade template for the main page. The template should have spaces for:**

  - your 5 most recent tweets
  - your 5 most recent friends
  - your 5 most recent private messages

It should also include your personal Twitter name and profile image at the top of the screen.

Styling is not the important part of this project. Craft your template markup to take advantage of the CSS we've provided you. Knowing how to work with someone else's styles is a very important skill as a full-stack developer. Pay attention to class names, inheritance, and so on. Try to avoid element types that are not used in the provided HTML and CSS files.

- **Using Node and Express, request the data you need from Twitter's API, render it in your template, and send it to the client at the "/" route. Please avoid using Express generator to set up this project. It will be good practice to set up a simple Express app yourself!**

- **Each rendered result must include all of the information seen in the sample layout:**

  *tweets -message content -# of retweets -# of likes -date tweeted

  *friends -profile image -real name -screenname

  *messages -message body -date the message was sent -time the message was sent

  Note that you don't have to display direct messages as a back and forth conversation. You only need to display the last 5 messages that were received, or the last 5 messages that were sent.

- **Make sure the application actually renders your correct Twitter information by running it on your local machine and comparing it to your recent Twitter activity.**

Extra Credit

To get an "exceeds" rating, you can expand on the project in the following ways:

3 steps

- **Add a section to the bottom of your page that allows a user to post a new tweet.**

- **Add an error page to your application, so that if anything goes wrong with your routes, the user will see a friendly message rendered, instead of the default error code.**

- **Include your personal background image from Twitter as a background for the page header.**

NOTE:

- To get an "Exceeds Expectations" grade for this project, you'll need to complete **each** of the items in this section. See the rubric in the "**How You'll Be Graded**" tab above for details on how you'll be graded.
- If you're shooting for the "Exceeds Expectations" grade, it is recommended that you mention so in your submission notes.
- Passing grades are final. If you try for the "Exceeds Expectations" grade, but miss an item and receive a "Meets Expectations" grade, you won't get a second chance. Exceptions can be made for items that have been misgraded in review.

Download files

Zip file

Project Resources

External Link

How to Create a Twitter App in 8 Easy Steps

External Link

Manage Twitter Apps

External Link

Twit Module

External Link

Twitter’s REST API Docs

External Link

Express Basics

External Link

npm Basics

Need Help?

Have questions about this project? Start a discussion with the community and Treehouse staff.

Get Help