# Antibuddies API Handoff Document

## Overview

Our goal was to create a simple, convenient REST API to tie the rest of the project together. This documentation is not an overall project scope outline, rather the infrastructure tools needed to support the project scope. This API allows any other portion of the project (such as the student quizzes, the virtual labs, etc.) to use any tools, frameworks, or language and still be able to easily communicate with each other and the database. The API is mostly used to interface with the database, but also handles things like user authentication and session management.

The stack is composed of Node.js (express) for the server, and MongoDB for the database.

## Getting Started

This section of the project consists primarily of a Node.js webserver application, which implements a REST API, as well as a client-side javascript SDK, which exposes API endpoints as functions. In addition to these externally useful components, this project also includes internally used HTML files to help with testing, demonstration, and API development.

**What you’ll need**

* Node.js
* NPM
* git

Our group did all development on glitch.com, a free service allowing realtime collaboration within a project. Glitch provides all the tools necessary to run and edit the project. Feel free to “remix” this project to start developing on your own:

<https://glitch.com/edit/#!/antibuddies-api>

Otherwise, our code is contained under the “/api” directory of the git repository:

<https://github.com/bradleypeterson/antibuddies>

If you wish to install and run the project locally rather than using glitch.com, simply run

npm start

You will need to have 3 environment variables set, “PORT”, “SECRET”, and “DB\_URI”. You can do this by simply running

PORT=5000 SECRET=... DB\_URI=... npm start

on Unix. This is more complicated on windows, and I would suggest instead using a ‘.env’ file with this package: <https://www.npmjs.com/package/dotenv>

**Do not commit these environment variables to the repository, or anywhere public**

Our environment variables are set as such:

PORT: **5000** (this is dependent on the machine/server running the code.)

SECRET: **\*\*\*\*\*** (this is used for token validation. Changing this is fine, it just means all existing sessions will be invalidated)

DB\_URI: **\*\*\*\*\*** (this is attached to a heroku project just for ease of use, the project is irrelevant. Do not expose the database URI, it gives full access to the database.)

(Brad Peterson has the private information.)

## Project Structure

The entrypoint for the code is **/server.js**. This starts up the web server using Express, and sets up some configuration for it.

It then passes the server into **/endpoints.js**. This includes some basic endpoints used for local pages for testing.

It also runs **/api.js**. Most of the code resides in this file. All public-facing API endpoints are written here (but they don’t have to be, feel free to organize them into separate files).

All endpoints in **/api.js** are documented with a description, its parameters, and its return value.

The project uses a MongoDB database, and Mongoose is used to interface with it. We have created several mongoose Schemas for each of the collections we are handling. All schema models are visible inside **/models**. Reference these when creating new documents.

The client-side javascript SDK for the API is at **/public/js/api.js**. This includes a function for every API endpoint. The BASE\_URL will need to be changed if the API is hosted somewhere else. This file is used by the other Antibuddies projects that run pages in a browser, such as login pages, quiz pages, etc. If a project includes a client application that is not run in the browser, it will need to interface with the REST API with HTTP calls directly. This is explained in the **/implementation.md** file.

The other files in **/public** are for external viewing, such as documentation pages.

The files in **/pages** are mostly for internal use, for testing and development purposes.

**/pages/api.html** is used for testing endpoints. This page is accessible at /api (such as [antibuddies-api.glitch.me/api](https://antibuddies-api.glitch.me/api))

**/pages/index.html** is a test page for creating users. **/pages/login.html** is for logging in, and **/pages/logout.html** is for logging out. These pages are not meant to be used by users of the app, they are just for internal testing and demonstration.

**/pages/test/sdk.html** is our testing suite for the javascript SDK. Its purpose is to run all the available functions in the public SDK script (**/public/js/api.js**), which in turn tests the API functionality itself.

## Unfinished Items

* API keys + authentication
  + This would be given to other parts of the project to allow them to call higher-security endpoints, such as any DELETE endpoints. The user token authentication already exists, but an API key would allow code to run these functions without needing a user account.
* Remaining unit test functions
  + Many of the existing SDK functions do not have a corresponding unit test. This mostly includes POST and DELETE endpoints, since we wanted the API key implemented first because these unit tests would have to always create and delete documents as to not fill/destroy the database.
* Deployment
  + The API exists and functions already thanks to “glitch.com”, but will be likely to cause issues if used by the public-facing app, as it is modified when developing. The API server should be deployed somewhere more persistent.
  + Brad Peterson has a CS Project server for deploying the project.