



Photo by [Joshua Sortino](#) on [Unsplash](#)

Code Tutorial

Carolyn Mary Stolze

Overview

Reverse engineering of starter code which provides a tutorial to use as a starting point for a new project.

Specifications

A Node.js application using Sequelize and Passport

Node.js is an open-source, cross-platform, back-end, JavaScript runtime environment that executes JavaScript code outside a web browser. Node.js uses asynchronous programming.

NPM is the package manager for the Node JavaScript platform, it puts modules in place so that node can find them. Most commonly, it is used to publish, discover, install, and develop node programs.

Sequelize is a powerful library in Javascript that makes it easy to manage a SQL database and can layer over different protocols. At its core, Sequelize is an Object-Relational Mapper – meaning that it maps an object syntax onto our database schemas.

Passport is a popular authentication middleware for Node. This separation of concerns keeps code clean and maintainable.

Tutorial

I. Directory Setup and Files

- ❏ Develop (root directory)
 - ❏ config
 - ❏ middleware
 - isAuthenticated.js
 - config.json
 - passport.js
 - ❏ models
 - index.js
 - user.js
 - ❏ public
 - ❏ js
 - login.js
 - members.js
 - signup.js
 - ❏ stylesheets
 - style.css
 - login.html
 - members.html
 - signup.html
 - ❏ routes
 - api-routes.js
 - html-routes.js
 - package.json
 - server.js

II. Develop (root directory)

This is the main folder holding all code and necessary assets.



III. package.json

All npm packages contain a file, usually in the project root, called `package.json` which holds various metadata used to give information to npm that allows it to identify the project as well as handle the project's dependencies.

IV. server.js

`server.js` is our backend entry point where we create our server that hears requests from users and returns something, such as an HTML or image file, or completes a process. This file includes:

PORT SETUP:

- Defines a port to listen for incoming requests (used in our listener).
- Includes `"process.env.PORT"` which lets the port be set by Heroku.

EXPRESS APP:

- "require" npm packages:
 - **express**: A Node.js web application framework that provides a robust set of features to develop web and mobile applications.
 - **express-session**: When implemented, every user of your API or website will be assigned a unique session, and this allows you to store the user state
- App instance initialized.

MIDDLEWARE:

- **express.urlencoded** and **express.json** which handles data parsing.
- **express.static** which serves static content from the "public" directory.
- **session** to keep track of our user's login status.
- **passport.initialize** because we are using an Express app.
- **passport.session** because we are using persistent logins.

ROUTES:

- "require" **html-routes.js** and **api-routes.js** files to provide our server a "map" of how to handle incoming requests and send responses.

DATABASE:

- "require" **models** folder which contains files that define our database tables. (Necessary to require here for syncing.)
- **db.sequelize.sync()** syncs our database.

START SERVER

- **app.listen** starts the server to begin listening to client requests.

V. Config Directory

❏ middleware

- **isAuthenticated.js**

- Checks if user is already logged in and defines the route to take depending on the result. (Restricts routes if condition not met.)
- Accessed by html-routes.js

- **config.json**

- This is where we tell Sequelize CLI *how* to connect to the database.
- Accessed by models/index.js

- **passport.js**

- “require” **passport** and **passport-local** which allows us to authenticate the user via email and password.
- “require” our models folder to provide access to the database for authentication searches.
- Dependent on “passport,” “passport-local”, and our models directory.
- Accessed by server.js and api-routes.js

VI. Models Directory

- **index.js**

- This is where we create a Sequelize instance and connect to the database.
- Dependent on “fs,” “path,” “sequelize,” and our config.json file.
- Accessed by passport.js, api-routes.js

- **user.js**

- Defines the database tables.
- Dependent on “bcryptjs” for password hashing.
- Accessed by api-routes.js, passport.js

VII. Public Directory

❏ js

- **login.js**
 - Front-end script to capture the data we need to send to our back-end script and provides next steps thereafter.
 - Dependent on & accessed by login.html, api-routes.js
- **members.js**
 - Front-end script to capture the data we need to send to our back-end script and provides next steps thereafter.
 - Dependent on & accessed by members.html, api-routes.js
- **signup.js**
 - Front-end script to capture the data we need to send to our back-end script and provides next steps thereafter.
 - Dependent on & accessed by signup.html, api-routes.js

❏ stylesheets

- **style.css**
 - Provides color, font and other styling to html pages.
 - Accessed by login.html, members.html, signup.html
- **login.html**
 - Defines the layout of page displayed on client.
 - Includes input fields to gather user information.
 - Dependent on Bootstrap, style.css, jQuery, and login.js
 - Accessed by html-routes.js
- **members.html**
 - Defines the layout of page displayed on client after login success.
 - Dependent on Bootstrap, style.css, jQuery, and members.js
 - Accessed by html-routes.js
- **signup.html**
 - Defines the layout of page displayed on client.
 - Includes input fields to gather user information.
 - Dependent on Bootstrap, style.css, jQuery, and signup.js
 - Accessed by html-routes.js

VIII. Routes Directory

- **api-routes.js**
 - Provides our server with the *back-end* “map” of how to handle incoming requests and their responses.
 - Dependent on passport.js to authenticate the user before proceeding.
 - Dependent on our models directory to interact with the database.
 - Dependent on node module “path” for path locations.
 - Accessed by server.js, login.js, members.js, signup.js
- **html-routes.js**
 - Provides our server with the *front-end* “map” on which html file to send to client and what content to serve the html files.
 - Dependent on our custom middleware isAuthenticated.js for checking if a user is already logged in.
 - Dependent on node module “path” for path locations.
 - Accessed by server.js, login.html, members.html, signup.html

IX. Editing

PAGE SETUP:

- Edit the .html files in the **public folder**.

PAGE STYLING:

- Edit style.css in the public > **stylesheets folder**.

FRONT-END SCRIPTS:

- Edit files in the public > **js folder**.

BACK-END SCRIPTS:

- Edit files in the **routes folder**.

DATABASE:

- Edit files in the **models folder**.

X. Installation and Usage

The **package.json** file lists all the npm packages that are needed (“dependencies”) in order to run the application.

- **Run command** `npm i` in your CLI to install all dependencies.
- **Run command** `npm start` in your CLI.
- **Go to** localhost:8080