

# Milestones for Web Developer P6

## Build a Secure API for a Review App

In this document, you'll find an example of milestones to follow for delivering your project. You'll see:

- Which section of the deliverable corresponds to which milestone.
- Recommendations for delivering each part.
- Issues to be aware of.
- The estimated progress on the whole project.
- Helpful external resources for each milestone.

### General Recommendations

Throughout each step, make sure that the server starts correctly, with a running message in the console stating that it is connected to MongoDB. If you receive a message stating that the server cannot start, check the backend console for errors.

Test your code at each milestone frequently. It can prevent you from forgetting important parts of the code and alert you to errors.

### Milestone #1 : Start the back-end server

15% progress



 **Milestone deliverable:** Initialized project

 **Recommendations:**

- You can follow these steps:
  - Create an empty project to start the Node.js server.
  - Install Express.
  - Install Mongoose.
- Bodyparser is included with Express 4.16; you do not need to install it.
  - Use ( `express.json()` ) to parse the body request.

 **Issues to be aware of:**

- If another process is using the 3000 port, restart your computer (to allow the use of the port).
- *Unable to connect to MongoDB.* Check the connection string, username, and password. Also, check MongoDB Atlas (or a similar service to use) to allow all IP addresses to connect to the cluster.



### Resources:

- [How to use Express.](#)
- [Tips on connecting your MongoDB with Node.js.](#)
- [An official Mozilla guide for working with your first Mongoose model.](#)
- [Use environment variables to make your Node.js work everywhere.](#)

## Milestone #2 : Build the user route

30% progress

---



### Part of Deliverable: API



### Recommendations:

- Create the following parts of the user route:
  - User model
  - User route
  - User controller
- Check to see that the user can do the following:
  - Create an account.
  - Login and have a valid token.
- Check the DevTools network tab for further information.



### Issues to be aware of:

- Make sure that the password is hashed.
- A user cannot sign up more than once with the same email. Make sure that the code checks for a unique email.



### Resources

- [What API request verbs are and how to use them.](#)

## Milestone #3 : Start the middleware

40% progress

---



### Part of Deliverable: API



### Recommendations:

- Add multer for images.
- Add authorize for token validation.
  - You must add authorize before building the sauce route because authentication is required for a user to perform any action.

### ⚠ Issues to be aware of:

- *Imported images are missing.* Ensure that you have configured the images public folder and added the images folder inside the back-end project at the project root level.
- *Multer is not correctly configured.* The folder from Milestone #1 may be missing. Alternatively, the images may have been named in an incorrect framework. Review the course section on multer configuration to ensure that you've implemented it correctly.
- Be sure to add the static path to the application for serving images.



### Resources

- [How to configure multer to allow file uploads.](#)

## Milestone #4 : Build the Sauce route

70% progress



### Part of Deliverable: API



### Recommendations:

- Create the following parts of the Sauce route:
  - Sauce model
  - Sauce route
  - Sauce controller
- Authorize all of the functions using the authorize middleware.
- Check that the user is able to do the following:
  - Add a new sauce.
  - Delete a sauce.
  - See all of the sauces.
- Check the DevTools network tab for further information.

### ⚠ Issues to be aware of:

- *401 Error (user is not authorized).* Check the authorization middleware and ensure that it can read the userId. Also, check that there are no typos or other errors.
- *Images are not shown on the front end.* Again, check that the images folder is correctly configured and added in the back end.
- *Multer does not save images.* Review the course section on multer configuration to make sure that you have set it up correctly (folder, naming format).



### Resources

- [How to add or remove a file from gitignore.](#)
- [Read this article if you are having trouble using JavaScript array methods.](#)

## Milestone #5 : Finish the Sauce route

100% progress

---

 **Part of Deliverable:** Completed API

 **Recommendations:**

- Run through the app as a user to check that all functions have been implemented correctly:
  - The user can make both types of requests:
    - Request with a file present.
    - Request without a file.
  - All three scenarios of the 'like' function (1, 0, -1) run correctly.
    - A user can like or dislike the sauce or remain neutral.
  - Only the Sauce owner can modify or delete an existing sauce

 **Issues to be aware of:**

- *401 Error (user is not authorized).*
- *Multer does not save images.*
- *Images are not shown on the front end.*
- *Data is not modified when the user attempts to modify an existing sauce.* Check the names for the properties in the schema. Be careful not to mark a non-required item as required.
- *The 'modify' function fails when an image is uploaded or modified.* Make sure the function covers both conditions (with and without a file on the request).
- *The 'like' function fails when the user tries to like or dislike a sauce multiple times.* The array should only ever have one value for 'userId' inside it. Check that the function adds or removes 'like' values to/from the correct array.
- *The sauce owner cannot see 'modify' and 'delete' buttons.* The sauce userId must be valid and contain no typos, since only the sauce owner can modify or delete a sauce.

 **Project completed!**