

Bike Part Picker Adding a Web Server Report

Tony Gao
Matthew Briden
Cong Wang
Peter Nguyen
Rajesh Susai

HTTP Requests / Routes:

getUserData:

Route: GET '/users/'+user

- Gets the information for the user from a userId that is passed in as a parameter; replaces the server function getUserData()
- A user that's attached to the same userId can get their own user data

getBuildData:

Route: GET '/builds/avoid/'+buildId

- Gets the build information for the build of that buildId; replaces the server function getBuildData()
- A user that has the build of that buildId in their buildList can view the contents of that build

getParts:

Route: GET /parts_default

- Gets the list of parts for a certain bike type(currently hardcoded server-side, will be passed as a parameter as the follows : /parts_default/:biketypeid)

getBuilds:

Route: GET /builds/:userId

- Gets the list of builds for a given user, the user bearer token is validated against the url parameter for authentication.

getPartName:

-Route: GET /builds/:buildId/partType/:partTypeId/users/:userId

-Method Replaced: getPartName(partId, partList, cb)

- Gets the name of the selected part in a particular build. The partTypeId is used so that the part name will populate in the correct field for that part type.
- Since the user is authorized to update a build further upstream this does not need any authorization.

getPartPrice:

-Route: GET /partType/:partTypeId/builds/:buildId/users/:userId

-Method Replaced: getPartPrice(partId, partList, cb)

- Gets the price of the selected part in a particular build. The partTypeId is used so that the part price will populate in the correct field for that part type.
- Since the user is authorized to update a build further upstream this does not need any authorization.

addPart:

- Route: PUT /builds/:buildId/parts/:partId
- Method Replaced: addPart(buildId, partId)
- Adds a part to a new or existing build and updates the build's part list and total price along the way. Writes the build to the database updating a build with this new info.
- The user is authorized to update/create a build further upstream in the build process meaning we do not need authorization for this request.

resetDatabase :

Route: POST /resetdb

- Resets the server-side database

updateAccount:

- Route: PUT /user/:users
- Method replaced: updateAccount(userId, fName, lName, email, uName, newPassword)
- Updates the account information with the input from the user.

writeBuildName:

- Route: 'PUT' '/builds/' + buildId + '/build_name/' + buildName, {price: buildPrice}
- Method replaced: writeBuildName() function
- Writes the buildName with the input from the user.

Special Server Setup Procedure:

There are no special server setup procedures. 'npm install' should be run in both client and server folders. 'npm run watch' must be run in './client' and 'node src/server.js' must be run in './server'.

Individual Contributions:

Tony:

- Get parts route
- Get builds route
- ResetDB route
- Step 1 (client server folder separation), server.js parsing logic/boilerplate, moved a portion of client server code to server/server.js
- Fixed all issues from previous report besides account->build relation. However this problem is also considered to be fixed. The logic for creating builds factors in userId, but the userId is still lightly hardcoded. (selectBikeType takes userId as a parameter, which generates a buildId that is then used for the rest of the build process, but the userId is currently hardcoded as a prop passed to the Build component in client/app.js), I also made this fix.
- Error banner

Matt:

- Add part route
- Get part name route
- Get part price route

- Updated the logic used in SelectBikeParts to no longer use readDocument directly and moved those functions to the server. Changed where the functions were called to use callback logic.
- Did the same as the above for the one function that used readDocument in SavedBuilds.

Cong:

- Helped Peter with debugging Account Page
- Worked with Raj, Matt to debug entirety of build.js
- selectBikeType routing
- Help with account routing/bug fixes, state logic during migration to server
- getUserData route, getBuildData route
- Errorbanner util addition
-

Peter:

- Added AccountUpdate route
- Fix Account to display user information
- Fixed button to update Account information
- Helped Rajesh with getBuildData route

Rajesh:

- Added getUserData route
- Added getBuildData route
- Helped people get rid of readDocument calls outside of server.js
- Helped integrate writeBuildName route
- Worked with Andrew on selectBikePart route
- Helped Andrew fix Build.js state logic
- Helped Andrew fix SelectBikeParts state logic
- Integrated proper-functioning server functions into Build.js
- Integrated proper-functioning server functions into SelectBikeParts.js
- Helped Matt integrate proper-functioning server functions in SavedBuilds

Lingering Bugs:

- Can't switch out parts when editing/creating a build.
 - A PUT route and edit logic must be added to the server and client respectively.
- Price displays floating point roundoff error
- Review button does not show price if it is clicked on an existing build without adding any part
- Since parts are saved to a build when they are selected, if the review and save buttons are not clicked after creating a build that build will show up in saved builds without a name nor a price.
- Some issues with Account