

# SmartCards

## HTTP Requests/ Routes:

GET /stackId

- This gets the cards in the given stack. It replaces the getCardsInStack function.

GET /userId

- This gets the stacks given the user. It replaces the getStacksFromUser function.

GET /settings/userId

- This gets the user data. It is used to display the user's info in the settings page. It replaces the getSettingsData function. The currently user can only get their own settings data.

PUT /stackId [carddata] where carddata is the new card being pushed to the stack

- This adds a new card to a stack. It replaces the saveCard function.

POST /userId [stackdata] where stackdata is the new stack being pushed to the users set of stacks

- This creates a new stack. It replaces the saveStack function

PUT /userId [userData] where userData is the the name, description, and email of the user

- This saves new user settings. It replaces the saveSettings function.

PUT /settings/userId where user settings such as full name, description, and email are updated and saved.

## Contributions:

Yifeng Shi imported the required modules, added all the fundamental functions, and implemented the GET request of home -> grid by clicking stack name.

Conor and Craig pair programed the newcard and savecard function to utilize a new url structure. The page and routing had to be redone and the state initiation also had to be now take in a user and stack parameter. They also had to redesign grid to include a new button within the page instead of on the nav bar. Conor finished the reset database and finished the routing that was missing

Liam worked on getStacksFromUser so the stack information is accurately reflected on the homepage. Also added a “Back to Stack” button to return to grid from the edit card page and a text box to create a new stack from the home page.

Liam, Conor, and Craig pair programmed the make stacks functionality. As this was a new feature we didn’t replace any currently used methods and started constructing newer ones to create and add stack to users list of stacks

Craig and Conor worked on implementing the schemas that are used with the validate function

Zung worked on the Settings page and got the editing data such as user’s full name and email to work. She wrote some functions for the client server and the server server. She had help from the TA and the instructor, but she tried hard and figured it out herself at the end. Even when the page is refreshed, the saved settings still stay unless we reset the database, so it is successful.

## Future Feature:

### Functionality:

Adding Stacks: Fix bug so that adding a stack works

Multi media cards: Allow a user to make cards for (music, images, videos, and text)

Filtered stack search: Allow users to search their stacks from the home page.

Saving pictures: Upload and save profile and background pictures

### Usability:

Notify a user that they’ve successfully added a card to the stack.

Slight Bugs: the validate functions are currently not cooperating with the schemas we have listed