- /user/login
    - Type - POST
    - Takes in JSON

```
{
    "email" : "users email",
    "password" : "users password"
}
```

    - Used to log a user in
    - Returns json data with users account information

- /user/create
    - Type - POST
    - Takes in JSON

```
{
    "firstName" : "first name",
    "lastName" : "last name",
    "phoneNumber" : "phone number",
    "email" : "email",
    "address" : "users address",
    "userName" : "user name",
    "password" : "users password"
}
```

    - Used to create a new user account
    - Will return a string
        - Account Exists - email used already has an account
        - Account Created - account was created successfully
    - Once account is created it will send an email with a verification code to the user, there will need to be a pop up or separate page after they create their account so they can enter the code

- /user/update
    - Type - PUT
    - Takes in JSON
        - Even if the field has not changed, still send the old info with the request. All data fields need to be full

```
{
    "email" : "update email",
    "firstName" : "update first name",
    "lastName" : "updated last name",
    "phoneNumber" : "updated phone number",
    "address" : "updated address",
    "userName" : "Updated user name"
}
```

    - Used to update character account information
        - First name
        - Last name
        - Phone number
        - Address
        - Username
    - Returns a boolean
        - True it was updated successfully
        - False there was an error
- /user/updatePassword
    - Type - PUT
    - Takes in JSON
        - Make them type in their old password as well as the new password, it will be used to verify they type in the correct current password for the account before it is updated to the new one

```
{
    "email" : "user email",
    "oldPassword" : "users old password",
    "newPassword" : "users new password"
}
```

    - Allows user to update password from their dashboard
    - Returns a boolean
        - True for successful update
        - False if they old password didn't match correctly

- /user/verify
    - Type-Put
    - Takes in JSON
        - Send the users email along with the request for the purpose of finding their account in the database

```
{
    "email" : "users email",
    "verificationCode" : "typed in code"
}
```

    - Used after a new account is made and they are verifying their email
    - Returns a boolean

- /user/resendVerification
    - Type-PUT
    - Takes in plain text
        - Just send the users email as plain text to the endpoint, JSON format will not work
    - Used to resend a new verification code to the users email
    - Doesn't return anything

- /user/delete
    - Type - DELETE
    - Takes in plain text
        - Send the users email as plain text to the endpoint, JSON format will not work
    - Used to let a user delete their account from our database
    - Returns a boolean
        - True deleted successfully
        - False there was an error deleting the account
- /trip/nextDestination
    - Type - POST
    - Takes plain text, JSON format will not work
        - When on the first destination (the very first time the page is accessed) send 0
        - Everytime after that send the ID of the trip that was returned previous
    - Used to get a destination from the database. Back end increments through and loops the destination to show them all
    - Returns the information for a destination
        - Image name is returned that will be used to get an image stored in the webserver's files

- /trip/selectDestination
    - Type - PUT
    - Takes in JSON data
        - The usersID that the trip is being made for
        - The destinationID the user is selecting

```
{
    "usersID" : 1,
    "destinationID" : 7
}
```

    - Used to start building a trip for the user. It will create a trip in the database linked with the users account, and link the destination to the stored destination they chose
    - Returns the ID of the trip that was just created for the user