* Identifies all service endpoints required for your MVP ie. what are the URLs that your client will use to access the backend of your application.
* Describes the purpose of each endpoint based on the goals of a user persona as it relates *to their data* (i.e., not their intentions with respect the feedback they want, but how what they want to do will change their data).
* Provides example requests and responses, including erroneous requests (e.g., error message responses for when bad data is input).
* Diagrams how communication will go from your user interface pages to service endpoints.

# Users

5 endpoints are required for management of the users collection. Three of them manage the user login features using POST requests. The other three manage the built-in character sheet associated with the logged in user. These methods include POST, PUT, and GET requests.

## /api/users/register

Method: POST

Purpose: Endpoint is responsible for managing user registration. This takes the registration form data and creates a new user with name, email, username, and password.

Request/Response

Example workflow

## /api/users/login

Method: POST

Purpose: Endpoint is responsible for managing user login. This takes the login form data and sends it out for authentication such that the server can confirm credentials and generate the user context for the rest of the application.

Request/Response

Example workflow

## /api/users/logout

Method: POST

Purpose: Endpoint is responsible for managing user logout. This clears the current user context.

Request/Response

Example workflow

## /api/users/{id}/character/form

Method: PUT

Purpose: This endpoint is responsible for creation and updates to the player character data that is stored

PUT/GET – create/update character and get form data for updates

/api/users/{id}/character/form

GET – get character data

/api/users/{id}/character

Adventure Logs

POST – create log

/api/adventurelog

PUT – update log

/api/adventurelog/{id}

GET – get log ( requires a filter to do by type)

/api/adventurelog?type={type}

DELETE – delete log

/api/adventurelog/{id}

Random Monsters

GET monsters

/api/monsters

Random NPCs

GET npcs

/api/npcs