

Web API for Second-Hand Football Gear Site

Introduction

The API provides information on how certain requests are handled for a Second-Hand Football Gear shopping website. The website is single-page, so each request returns some data that will alter the look of the single HTML page.

Overview

The API contains several GET requests, most of which return JSON arrays containing either item objects or seller objects, which are to be displayed on the single HTML page. Data that was on the HTML page just beforehand will be removed and replaced, in order to match the meaning of each route (i.e. if the current route is /sellers, the display of the page will be a list of sellers. If a button is clicked to go to /items, then the list of sellers will be removed and replaced with a list of items).

The API also contains some POST requests, which correspond to certain forms. The idea of these is to get the data from the form, and add it to a relevant JSON array on the server, provided that all of the entries in the form are of a valid format.

Error Codes

The only status codes to expect are 200 (indicating the response of a certain request is OK) or 404 (indicating a wrong route).

Rate limit

There is no limit to the number of requests that can be sent.

GET 127.0.0.1:8090/home

```
127.0.0.1:8090/home
```

Returns JSON array of the 6 most recently listed items, in order of newest first.

GET 127.0.0.1:8090/items

127.0.0.1:8090/items

Returns JSON array of all listed items. Also contains a query variable in the route called 'sortOption', whose value determines the order in which the items are displayed. Possible options are sort by price (ascending or descending), size (ascending or descending), or no sort at all.

GET 127.0.0.1:8090/items/:itemType

127.0.0.1:8090/items/:itemType

Returns JSON array of all listed items of the specified type. Valid types are "gloves", "pads", or "helmets". Also contains a query variable in the route called 'sortOption', whose value determines the order in which the items are displayed. Possible options are sort by price (ascending or descending), size (ascending or descending), or no sort at all.

GET 127.0.0.1:8090/items/:itemType/:id

127.0.0.1:8090/items/:itemType/:id

Returns JSON array, with the first element being the item of the specified type whose unique ID matches the one given in the route. Valid item types are the same. The rest of the array contains items related to the main item, including items of the same type, or items listed by the same seller.

GET 127.0.0.1:8090/sellers

127.0.0.1:8090/sellers

Returns JSON array of all sellers. Also contains a query variable in the route called 'sortOption', whose value determines the order in which the sellers are displayed. Possible options are sort by number of listed items (ascending or descending), or no sort at all.

GET 127.0.0.1:8090/sellers/:id

127.0.0.1:8090/sellers/:id

Returns JSON array, with the first element being the seller whose unique ID matches the one given in the route. The second element of the array contains an array of items listed by that seller. Also contains a query variable in the route called 'sortOption', whose value determines the order in which the sellers items are displayed. Possible options are sort by

price (ascending or descending), size (ascending or descending), or no sort at all.

POST 127.0.0.1:8090/items/add

```
127.0.0.1:8090/items/add
```

Accepts a form for the listing of a new item. The submitted form contains information about the seller (name and email), and the item as a whole (type, description, size, price, and an image). The item data is then pushed onto the relevant JSON array so as to list a new item. If the seller is identified to be a new seller, then the seller data is pushed onto the relevant JSON array so as to add a new seller.

POST 127.0.0.1:8090/sellers/add

```
127.0.0.1:8090/sellers/add
```

Accepts a form for the addition of a new seller. The submitted form contains information about the seller (name and email). If the seller is identified to be already on the server, no data is pushed onto the relevant JSON array. Otherwise, the new seller data is pushed onto the relevant JSON array so as to add a new seller.