

API: (endpoint) /munchbot/api/

Table: <vendor_table> (e.g. safeway_<date>)

GET /vendor/<vendor_id:string>/job/<job_id:string>/status

- Returns the status of a request, given the **job_id**

GET /vendor/<vendor_id:string>/products/sample

- Returns a sample set of products from the vendor API
- Acts as a health check to make sure the vendor API requests are still working

GET /vendor/<vendor_id:string>/products/category/<category_id:int>/page/<page:int>/per_page/<per_page:int>

- Returns a list of products from the vendor given a category
- Supports pagination

GET /vendor/<vendor_id:string>/products/all

- Async request, returns a **job_id** that can be used to check the status
- Saves the entire product catalog of the vendor to a **data.json** file

POST /vendor/products/upload

{ vendor_id: <string> }

- Async request, returns a **job_id** that can be used to check the status
- Reads the latest **data.json** (cleans data as well) and uploads to a new table in mysql
- If no new **data.json** exists or is malformed it will return an error and delete **data.json**
- If successful it will:
 - set the new table as the default table for the grocery app queries
 - delete the **data.json** file

Table: product (custom)

GET /products/vendor/<vendor_id:string>/page/<page:int>/per_page/<per_page:int>/search/<filter:string>

- Returns a list of products given a **vendor_id**
 - **none** will search through all vendors (e.g. safeway and custom)
 - **custom** will search through only the manually entered products
 - if vendor is not found it will return an empty response
- **filter** value will be used for keyword matching on the **product_name**
- Supports pagination

GET,DEL /product/<product_id>/vendor/<vendor_id:string>

- Returns or deletes a specific product
- If **vendor_id** is supplied it will only look in either the vendor table or custom table

POST,PUT /product

```
{  
  product_id: <int> (optional),  
  product_name: <string>,  
  vendor: <string>,  
  price: <float>  
}
```

- Returns a success or error
- If **product_id** is supplied, it will update the row

Table: item

GET /items/page/<page:int>/per_page/<per_page:int>/search/<filter:string>

- Returns a list of items supporting pagination
- **filter** value will be used for keyword matching on the **item_name**

GET,DEL /item/<item_id>

- Returns or deletes a specific item

POST,PUT /item

```
{  
    item_id: <int> (optional),  
    item_name: <string>,  
    item_desc: <string>,  
}
```

- Returns a success or error
- If **item_id** is supplied, it will update the row

Table: **item_map**

GET /item_maps/page/<page:int>/per_page/<per_page:int>/search/<filter:string>

- Returns a list of item mappings, supporting pagination
- **filter** value will be used for keyword matching on the **vendor_id**

GET,DEL /item_map/<item_map_id>

- Returns or deletes a specific item mapping

POST,PUT /item_map

```
{  
    item_map_id: <int> (optional),  
    item_id: <int>,  
    vendor_id: <string>,  
    product_id: <int>  
}
```

- Returns a success or error
- If **item_map_id** is supplied, it will update the row

Table: **recipe**

GET /recipe/page/<page:int>/per_page/<per_page:int>/search/<filter:string>

- Returns a list of recipes, supporting pagination
- **filter** value will be used for keyword matching on the **recipe_name**

GET,DEL /recipe/<recipe_id>

- Returns or deletes a specific recipe

POST,PUT /recipe

```
{  
    recipe_id: <int> (optional),  
    recipe_name: <string>,  
    recipe_desc: <string>,  
    recipe_link: <string>,  
    recipe_photo: <string>,  
    recipe_time: <float>,  
    recipe_difficulty: <int>  
}
```

- Returns a success or error
- If **recipe_id** is supplied, it will update the row

Table: **ingredient**

GET /ingredient/page/<page:int>/per_page/<per_page:int>/search/<filter:string>

- Returns a list of ingredients, supporting pagination
- **filter** value will be used for keyword matching on the **ingredient_unit**

GET,DEL /ingredient/<ingredient_id>

- Returns or deletes a specific ingredient

POST,PUT /ingredient

```
{  
    ingredient_id: <int> (optional),  
    item_id: <string>,  
    ingredient_quantity: <string>,  
    ingredient_unit: <string>,  
    recipe_id: <string>  
}
```

- Returns a success or error
- If **ingredient_id** is supplied, it will update the row

Table: **grocery_list**

GET /grocery_list/page/<page:int>/per_page/<per_page:int>/search/<filter:string>

- Returns a list of grocery lists, supporting pagination
- **filter** value will be used for keyword matching on the **created_date**

GET,DEL /grocery_list/<grocery_list_id>

- Returns or deletes a specific grocery_list

POST,PUT /grocery_list

```
{  
    grocery_list_id: <int> (optional),  
    grocery_list_cost: <float>  
}
```

- Returns a success or error
- If **grocery_list_id** is supplied, it will update the row

Table: **cart**

GET /cart/page/<page:int>/per_page/<per_page:int>/search/<filter:string>

- Returns a list of carts, supporting pagination
- **filter** value will be used for keyword matching on the **created_date**

GET,DEL /cart/<cart_id>

- Returns or deletes a specific cart

POST,PUT /cart

```
{  
    cart_id: <int> (optional),  
    item_map_id: <int>,  
    grocery_list_id: <int>  
}
```

- Returns a success or error
- If **cart_id** is supplied, it will update the row

Table: ****mix** (custom queries joining various table data)**