John Zaiss

SWDV 691: Software Development Capstone

Joseph Gradecki

19 July 2020

# Re-Design: Service Layer(s)

# My Garden Web App

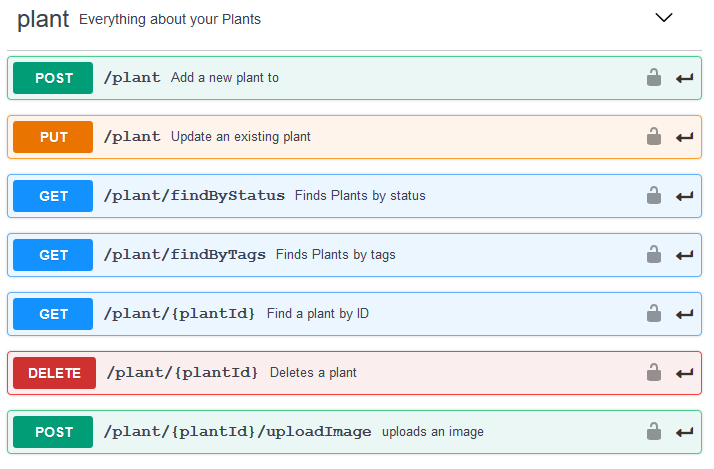
# Overview

The My Garden app is designed for the personal gardener who wants to manage their gardening experience. The purpose is to track progress of items planted in the garden and use as a historical reference of what has been planted in previous seasons.

Creating records, storing photos and retrieving records are the primary functions of the My Garden App. We have two primary service endpoints **/plant** and **/garden**.

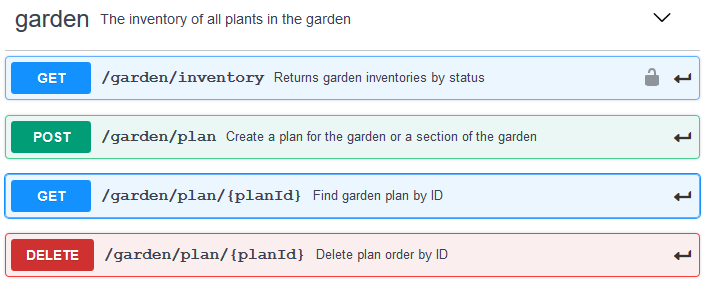
## /plant

The **/plant** endpoint has several operations



## /garden

The **/garden** endpoint has several operations



## UI Page Mappings

UI Page to service endpoint operation mappings

## Page Title: My Garden

This uses the **GET /garden/inventory** operation to return all active garden items by default.

## Page Title: Item Detail

This page is activated when a garden item is selected from either the search results page or by selecting a garden item on the **My Garden** page.

This uses the **GET /plant/{plantId}** to retrieve the plant record, all tags and photos related to the plant

## Page Title: Select Item

This is the entry page to update an item. The page is rendered with a dropdown list of all garden items queried from the service endpoint **GET /garden/inventory** to find all items. Then once an item is selected the **GET /plant/{plantId}** is used to retrieve the specific item and then the item is displayed on the **Update Item Form** page.

## Page Title: Update Item Form

This page is rendered with the **GET /plant/{plantId}** where the gardner can use **Add Photo** button to add a photo using the **POST /plant/{plantId}/uploadImage** operation to put the image in the database.

## Page Title: Dig in Garden

This is an advanced search page. The page is rendered with a dropdown list of all garden items queried from the service endpoint **GET /garden/inventory** to find all items. Note: This may prove to need a different operation once coding begins.

## Page Title: Search Results

This page can be created from any one of the following operations based on the type of search performed.

**GET /plant/findByStatus**

**GET /plant/findByTags**

**GET /plant/{plantId}**

## Page Title: Add Item

This page uses the **POST /plant** operation when a user hits the submit button.

# Example Responses

These are generic examples. More specific examples are planned once the database design is complete and aligned with the service layer.

[

{

"id": **0**,

"category":{

"id": **0**,

"name":"string"

},

"name":"sunflower",

"photoUrls":[

"string"

],

"tags":[

{

"id": **0**,

"name":"string"

}

],

"status":"available",

"watering":"Frequently",

"hardiness":"hardiness description",

"fertilization":"how to fertilize",

"size":"how tall and wide",

"spacing":"how far apart",

"details":"open text field"

}

]