

# Cooking Calendar: Service Layer

## Overview

I will use an ASP.NET core web API hosted by Azure. For now, user information will be passed in the token from the client otherwise an `userId` would need to be passed on almost every route. I have broken up the rest of the document by controllers which will have a base route for all endpoints. In theory, the error responses will only ever be seen while developing and will never occur in production as long as the server is running correctly and the client formats its input correctly. I will use my localhost server to demonstrate responses, but in the future, the base server URI "<https://localhost:7133>" will change.

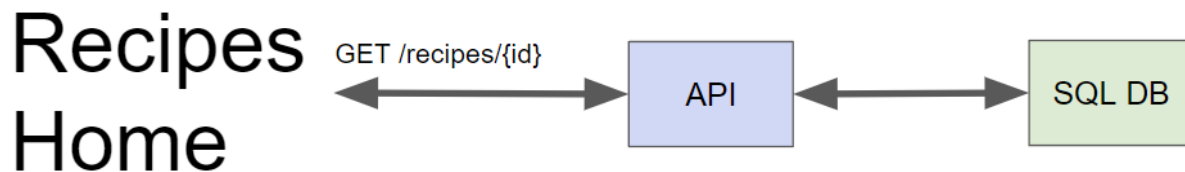
## Recipe Controller “/recipes”

### GET Requests

#### Retrieve Recipe

Route: “/recipes/{recipeId}”

Purpose: Used when the user navigates to the recipe page to view the recipe they have input. Also called on the home page to get details for the next upcoming meal.



Successful Request GET /recipes/1

Response: Status Code 200

```
{
  "id": 1,
  "name": "Tacos",
  "description": "Yummy Mexican Food",
  "servingSize": 12,
  "ingredients": [
    {
      "id": 1,
```

```
"ingredientId": 1,
"ingredient": "Ground Beef",
"measurement": 9,
"sortOrder": 1,
"amount": 1,
"numerator": 1,
"denominator": 2
},
{
  "id": 2,
  "ingredientId": 2,
  "ingredient": "Lettuce",
  "measurement": 5,
  "sortOrder": 1,
  "amount": 0,
  "numerator": 1,
  "denominator": 2
},
{
  "id": 3,
  "ingredientId": 3,
  "ingredient": "Cheese",
  "measurement": 5,
  "sortOrder": 1,
  "amount": 1,
  "numerator": 0,
  "denominator": 0
},
{
  "id": 4,
  "ingredientId": 4,
  "ingredient": "Taco Shells",
  "measurement": 0,
  "sortOrder": 1,
  "amount": 12,
  "numerator": 0,
  "denominator": 0
}
],
"tags": [
  {
    "id": 1,
    "tagId": 1,
    "name": "Mexican"
```

```
},
{
  "id": 2,
  "tagId": 2,
  "name": "Easy"
}
],
"steps": [
  {
    "id": 1,
    "step": "Brown @1",
    "sortOrder": 1,
    "ingredients": [
      {
        "id": 1,
        "recipeIngredientId": 1,
        "ingredientId": null,
        "numerator": 1,
        "denominator": 1
      }
    ]
  },
  {
    "id": 2,
    "step": "Get @1 and put @2 into each shell, top with @3 and @4 to taste",
    "sortOrder": 1,
    "ingredients": [
      {
        "id": 2,
        "recipeIngredientId": 4,
        "ingredientId": null,
        "numerator": 1,
        "denominator": 1
      },
      {
        "id": 3,
        "recipeIngredientId": 1,
        "ingredientId": null,
        "numerator": 1,
        "denominator": 12
      },
      {
        "id": 4,
        "recipeIngredientId": 2,
```

```

    "ingredientId": null,
    "numerator": 1,
    "denominator": 1
  },
  {
    "id": 5,
    "recipeIngredientId": 3,
    "ingredientId": null,
    "numerator": 1,
    "denominator": 1
  }
]
},
{
  "id": 3,
  "step": "Enjoy",
  "sortOrder": 1,
  "ingredients": []
}
]
}

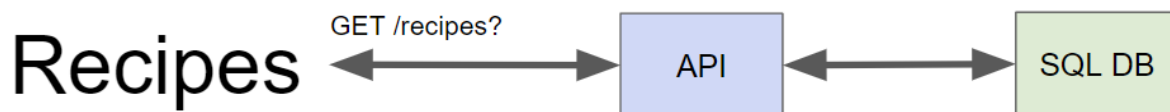
```

Error Request: <https://localhost:7133/recipe/-1>  
 Response: **400** Recipe with given Id could not be found

## Retrieve Filtered Recipes

Route: "/recipes"

Purpose: On the recipe search page the user can use this endpoint to search through all of their recipes using the filters. Results will be paginated so filtering needs to be done on the server side.



Successful Request 1: GET

<https://localhost:7133/recipes?Limit=10&Name=Ta&TagIds=1>

Response 1: **200**

```

{
  "id": 1,
  "name": "Tacos",
  "tags": [

```

```

    "Mexican"
  ],
  "ingredients": [
    "Ground Beef",
    "Lettuce",
    "Cheese",
    "Taco Shells"
  ]
}

```

Successful Request 2: GET

<https://localhost:7133/recipes?Limit=10&Offset=0&Name=Ta&TagIds=1&TagIds=2&MustBeVegetarian=true&MustBeDairyFree=false&MustBeGlutenFree=true&IngredientsToInclude=1&IngredientsToInclude=2&IngredientsToExclude=3&IngredientsToExclude=4&NotMadeSinceDate=2023-03-23>

Response 2: **204**

Error Request: <https://localhost:7133/recipes?Name=Ta&TagIds=1>

Error Response: **400**

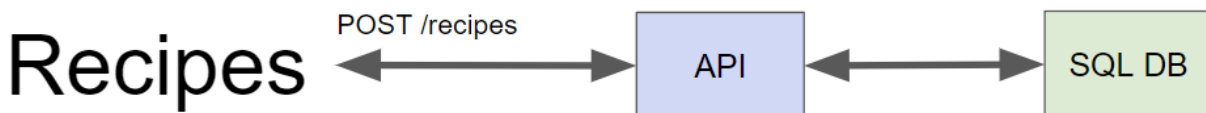
Limit is required

## POST Requests

### Create Recipe

Route: "/recipes"

Purpose: Called when the save button is clicked on a new recipe so the user can create a new recipe to cook.



Successful Request: Post <https://localhost:7133/recipe>

Body: {

```

  "id": null,
  "name": "Tacos",
  "description": "Yummy Mexican Food",
  "servingSize": 12,
  "ingredients": [
    {
      "id": null,
      "ingredientId": 1,

```

```
"ingredient": "Ground Beef",
"measurement": 9,
"sortOrder": 1,
"amount": 1,
"numerator": 1,
"denominator": 2
},
{
  "id": null,
  "ingredientId": 2,
  "ingredient": "Lettuce",
  "measurement": 5,
  "sortOrder": 1,
  "amount": 0,
  "numerator": 1,
  "denominator": 2
},
{
  "id": null,
  "ingredientId": 3,
  "ingredient": "Cheese",
  "measurement": 5,
  "sortOrder": 1,
  "amount": 1,
  "numerator": 0,
  "denominator": 0
},
{
  "id": null,
  "ingredientId": 4,
  "ingredient": "Taco Shells",
  "measurement": 0,
  "sortOrder": 1,
  "amount": 12,
  "numerator": 0,
  "denominator": 0
}
],
"tags": [
  {
    "id": null,
    "tagId": 1,
    "name": "Mexican"
  },
]
```

```
{
  "id": null,
  "tagId": 2,
  "name": "Easy"
}
],
"steps": [
  {
    "id": null,
    "step": "Brown @1",
    "sortOrder": 1,
    "ingredients": [
      {
        "id": null,
        "recipeIngredientId": 1,
        "ingredientId": null,
        "numerator": 1,
        "denominator": 1
      }
    ]
  },
  {
    "id": null,
    "step": "Get @1 and put @2 into each shell, top with @3 and @4 to taste",
    "sortOrder": 1,
    "ingredients": [
      {
        "id": null,
        "recipeIngredientId": null,
        "ingredientId": 4,
        "numerator": 1,
        "denominator": 1
      },
      {
        "id": null,
        "recipeIngredientId": 1,
        "ingredientId": null,
        "numerator": 1,
        "denominator": 12
      },
      {
        "id": null,
        "recipeIngredientId": 2,
        "ingredientId": null,
```

```

        "numerator": 1,
        "denominator": 1
    },
    {
        "id": null,
        "recipeIngredientId": 3,
        "ingredientId": null,
        "numerator": 1,
        "denominator": 1
    }
]
},
{
    "id": null,
    "step": "Enjoy",
    "sortOrder": 1,
    "ingredients": []
}
]
}
Response: 201
{
    "id": 1,
    "name": "Tacos",
    "description": "Yummy Mexican Food",
    "servingSize": 12,
    "ingredients": [
        {
            "id": 1,
            "ingredientId": 1,
            "ingredient": "Ground Beef",
            "measurement": 9,
            "sortOrder": 1,
            "amount": 1,
            "numerator": 1,
            "denominator": 2
        },
        {
            "id": 2,
            "ingredientId": 2,
            "ingredient": "Lettuce",
            "measurement": 5,
            "sortOrder": 1,
            "amount": 0,

```



```
    "numerator": 1,
    "denominator": 2
  },
  {
    "id": 3,
    "ingredientId": 3,
    "ingredient": "Cheese",
    "measurement": 5,
    "sortOrder": 1,
    "amount": 1,
    "numerator": 0,
    "denominator": 0
  },
  {
    "id": 4,
    "ingredientId": 4,
    "ingredient": "Taco Shells",
    "measurement": 0,
    "sortOrder": 1,
    "amount": 12,
    "numerator": 0,
    "denominator": 0
  }
],
"tags": [
  {
    "id": 1,
    "tagId": 1,
    "name": "Mexican"
  },
  {
    "id": 2,
    "tagId": 2,
    "name": "Easy"
  }
],
"steps": [
  {
    "id": 1,
    "step": "Brown @1",
    "sortOrder": 1,
    "ingredients": [
      {
        "id": 1,
```

```

        "recipeIngredientId": 1,
        "ingredientId": null,
        "numerator": 1,
        "denominator": 1
    }
]
},
{
    "id": 2,
    "step": "Get @1 and put @2 into each shell, top with @3 and @4 to taste",
    "sortOrder": 1,
    "ingredients": [
        {
            "id": 2,
            "recipeIngredientId": 4,
            "ingredientId": null,
            "numerator": 1,
            "denominator": 1
        },
        {
            "id": 3,
            "recipeIngredientId": 1,
            "ingredientId": null,
            "numerator": 1,
            "denominator": 12
        },
        {
            "id": 4,
            "recipeIngredientId": 2,
            "ingredientId": null,
            "numerator": 1,
            "denominator": 1
        },
        {
            "id": 5,
            "recipeIngredientId": 3,
            "ingredientId": null,
            "numerator": 1,
            "denominator": 1
        }
    ]
},
{
    "id": 3,

```

```
    "step": "Enjoy",
    "sortOrder": 1,
    "ingredients": []
  }
]
```

Error Request: POST <https://localhost:7133/recipes>

Body: {}

Response: **400**

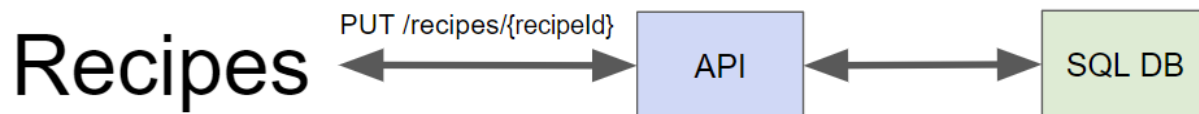
```
{
  "type": "https://tools.ietf.org/html/rfc7231#section-6.5.1",
  "title": "One or more validation errors occurred.",
  "status": 400,
  "traceId": "00-0d9f04cb86376360dffb0cb729620d03-b9f635586fca5feb-00",
  "errors": {
    "": [
      "A non-empty request body is required."
    ],
    "recipe": [
      "The recipe field is required."
    ]
  }
}
```

## PUT Requests

### Update Recipe

Route: "/recipes/{recipeId}"

Purpose: Called when the save button is clicked on an existing recipe so the user can update a recipe after it has been created.



Successful Request: PUT <https://localhost:7133/recipe/1>

```
Body: {
  "id": 1,
  "name": "Tacos",
  "description": "Yummy Mexican Food",
  "servingSize": 12,
```

```
"ingredients": [  
  {  
    "id": 1,  
    "ingredientId": 1,  
    "ingredient": "Ground Beef",  
    "measurement": 9,  
    "sortOrder": 1,  
    "amount": 1,  
    "numerator": 1,  
    "denominator": 2  
  },  
  {  
    "id": 2,  
    "ingredientId": 2,  
    "ingredient": "Lettuce",  
    "measurement": 5,  
    "sortOrder": 1,  
    "amount": 0,  
    "numerator": 1,  
    "denominator": 2  
  },  
  {  
    "id": 3,  
    "ingredientId": 3,  
    "ingredient": "Cheese",  
    "measurement": 5,  
    "sortOrder": 1,  
    "amount": 1,  
    "numerator": 0,  
    "denominator": 0  
  },  
  {  
    "id": 4,  
    "ingredientId": 4,  
    "ingredient": "Taco Shells",  
    "measurement": 0,  
    "sortOrder": 1,  
    "amount": 12,  
    "numerator": 0,  
    "denominator": 0  
  }  
],  
"tags": [  
  {
```

```
"id": 1,
"tagId": 1,
"name": "Mexican"
},
{
  "id": 2,
  "tagId": 2,
  "name": "Easy"
}
],
"steps": [
  {
    "id": 1,
    "step": "Brown @1",
    "sortOrder": 1,
    "ingredients": [
      {
        "id": 1,
        "recipeIngredientId": 1,
        "ingredientId": null,
        "numerator": 1,
        "denominator": 1
      }
    ]
  },
  {
    "id": 2,
    "step": "Get @1 and put @2 into each shell, top with @3 and @4 to taste",
    "sortOrder": 1,
    "ingredients": [
      {
        "id": 2,
        "recipeIngredientId": 4,
        "ingredientId": null,
        "numerator": 1,
        "denominator": 1
      },
      {
        "id": 3,
        "recipeIngredientId": 1,
        "ingredientId": null,
        "numerator": 1,
        "denominator": 12
      }
    ]
  },
  {
    "id": 4,
    "step": "Put @1 in a bowl, add @2 and @3, mix well",
    "sortOrder": 2,
    "ingredients": [
      {
        "id": 4,
        "recipeIngredientId": 1,
        "ingredientId": null,
        "numerator": 1,
        "denominator": 1
      },
      {
        "id": 5,
        "recipeIngredientId": 1,
        "ingredientId": null,
        "numerator": 1,
        "denominator": 1
      },
      {
        "id": 6,
        "recipeIngredientId": 1,
        "ingredientId": null,
        "numerator": 1,
        "denominator": 1
      }
    ]
  }
]
```

```

    {
      "id": 4,
      "recipeIngredientId": 2,
      "ingredientId": null,
      "numerator": 1,
      "denominator": 1
    },
    {
      "id": 5,
      "recipeIngredientId": 3,
      "ingredientId": null,
      "numerator": 1,
      "denominator": 1
    }
  ]
},
{
  "id": 3,
  "step": "Enjoy",
  "sortOrder": 1,
  "ingredients": []
}
]
}

```

Response: **204**

Error Request: PUT /recipes/1

```

Body: {
  "id": 1,
  "description": "string",
  "servingSize": 0,
  "ingredients": [
  ],
  "tags": [
  ],
  "steps": [

```

```

  ]
}

```

Response: Status Code 400

```

{
  "type": "https://tools.ietf.org/html/rfc7231#section-6.5.1",
  "title": "One or more validation errors occurred.",
  "status": 400,

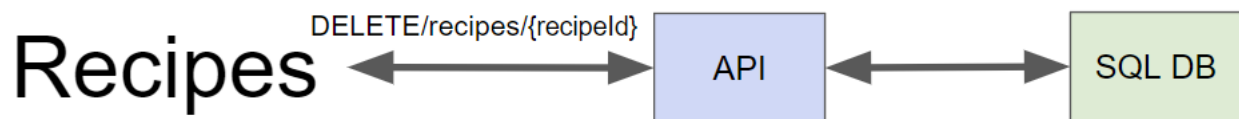
```

```
"traceId": "00-36692cb3b1829197a65595e1420d708a-bb9cb45e64cf8c72-00",
"errors": {
  "Name": [
    "The Name field is required."
  ]
}
```

## DELETE Requests

Route: `"/recipes/{recipeId}"`

Purpose: Called when the user clicks a delete button and expects the unwanted recipe to be deleted



Successful Request DELETE <https://localhost:7133/recipes/1>

Response: **204**

Error Request DELETE <https://localhost:7133/recipes/-1>

Response: Recipe can not be deleted

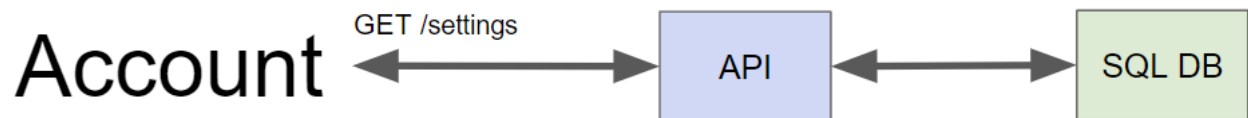
# Settings Controller “/settings”

## GET Requests

### Retrieve Settings

Route: “/settings”

Purpose: On the account page this endpoint will load the current user's settings so the user can make changes to them if needed.



Successful Request GET <https://localhost:7133/settings>

Response:{

"isDefaultMeasurementStandard": true,

"isDarkMode": true,

"isMonthDefaultView": false,

"categories": [

{

"id": 1,

"name": "Mystery",

"dayOfWeek": 1,

"categoryType": 4,

"tagId": null,

"ingredientId": null

},

{

"id": 1,

"name": "Taco Tuesday",

"dayOfWeek": 2,

"categoryType": 1,

"tagId": 1,

"ingredientId": null

},

{

"id": 1,

"name": "Chicken Day",

"dayOfWeek": 4,

"categoryType": 2,

"tagId": null,

"ingredientId": 1

},



```
{
  "id": 1,
  "name": "Take Out",
  "dayOfWeek": 0,
  "categoryType": 3,
  "tagId": null,
  "ingredientId": null
}
```

Error Request: GET <https://localhost:7133/settings>

Response: **404**

User does not exist

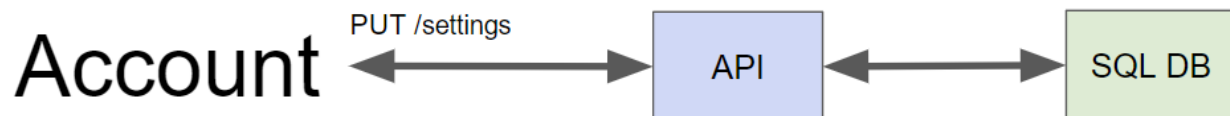
## PUT Requests

### Update Settings

Route: "/settings"

Purpose: When the user makes changes on the account page they will want to save the changes to make the website take into account their preferences.

Successful Request PUT <https://localhost:7133/settings>



Body: {

```

  "isDefaultMeasurementStandard": false,
  "isDarkMode": false,
  "isMonthDefaultView": true,
  "categories": [
    {
      "id": null,
      "name": "Mystery",
      "dayOfWeek": 1,
      "categoryType": 4,
      "tagId": null,
      "ingredientId": null
    },
    {
      "id": null,

```

```

    "name": "Taco Tuesday",
    "dayOfWeek": 2,
    "categoryType": 1,
    "tagId": 1,
    "ingredientId": null
  },
  {
    "id": null,
    "name": "Chicken Day",
    "dayOfWeek": 4,
    "categoryType": 2,
    "tagId": null,
    "ingredientId": 1
  },
  {
    "id": null,
    "name": "Take Out",
    "dayOfWeek": 0,
    "categoryType": 3,
    "tagId": null,
    "ingredientId": null
  }
]
}

```

Response: **204**

Error Request PUT <https://localhost:7133/settings>

Body:

```

{
  "isDefaultMeasurementStandard": true,
  "isDarkMode": true,
  "isMonthDefaultView": true
}

```

Response: **400**

```

{
  "type": "https://tools.ietf.org/html/rfc7231#section-6.5.1",
  "title": "One or more validation errors occurred.",
  "status": 400,
  "traceId": "00-b327928a1265986a74b500b655bf8942-ce19bd5bad86f74f-00",
  "errors": {
    "Categories": [
      "The Categories field is required."
    ]
  }
}

```

# Calendar Controller “/calendars”

## GET Requests

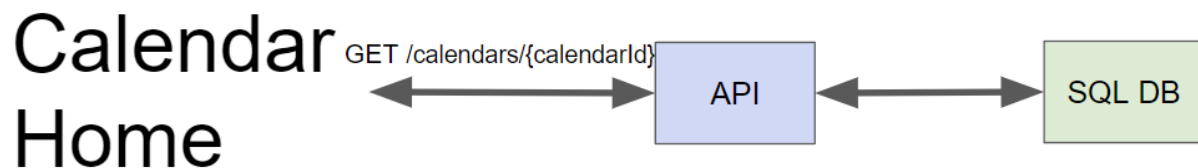
### Retrieve Calendar

Route: “/calendars/{calendarId}”

Purpose: The user will use this endpoint on the calendar page to see upcoming meals in a specified time frame. The user will also get upcoming meals on the home page from this endpoint since similar information will be displayed in both places.

Successful Request GET

<https://localhost:7133/calendars/1?StartDate=2023-03-23&EndDate=2023-03-31>



Successful Response: **200**

```
{
  "id": 1,
  "lastGenerated": "2023-03-23T20:42:56.700962-04:00",
  "meals": [
    {
      "id": 1,
      "mealDate": "2023-03-29",
      "isUserAssigned": false,
      "recipe": {
        "id": 1,
        "name": "Tacos",
        "tags": [
          "Mexican"
        ],
        "ingredients": [
          "Ground Beef",
          "Lettuce",
          "Cheddar Cheese",
          "Taco Shells"
        ]
      }
    }
  ],
},
{
```

```

    "id": 2,
    "mealDate": "2023-03-31",
    "isUserAssigned": false,
    "recipe": {
      "id": 1,
      "name": "Grilled Cheese",
      "tags": [
        "Easy"
      ],
      "ingredients": [
        "White Bread",
        "American Cheese",
        "Butter"
      ]
    }
  }
]
}

```

Error Request: GET

<https://localhost:7133/calendars/1?StartDate=2023-03-23&EndDate=2023-03-22>

Error Response: **400**

Start date must come before end date

## Post Requests

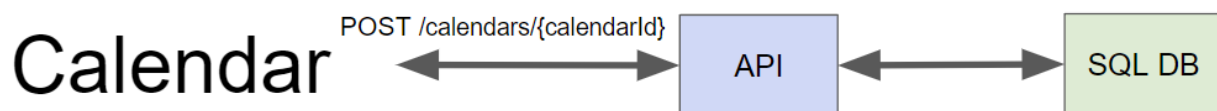
### Generate Plan

Route: `"/calendars/{calendarId}"`

Purpose: The user will use this endpoint on the calendar page to make the system generate meals based on the categories the user has set.

Successful Request: Post

<https://localhost:7133/calendars/1?StartDate=2023-03-23&EndDate=2023-03-31>



Body: `{}`

Response: **200**

```

{
  "id": 1,
  "lastGenerated": "2023-03-23T20:54:55.3134399-04:00",
  "meals": [
    {

```

```
"id": 1,
"mealDate": "2023-03-29",
"isUserAssigned": false,
"recipe": {
  "id": 1,
  "name": "Tacos",
  "tags": [
    "Mexican"
  ],
  "ingredients": [
    "Ground Beef",
    "Lettuce",
    "Cheddar Cheese",
    "Taco Shells"
  ]
},
{
  "id": 2,
  "mealDate": "2023-03-31",
  "isUserAssigned": true,
  "recipe": {
    "id": 1,
    "name": "Grilled Cheese",
    "tags": [
      "Easy"
    ],
    "ingredients": [
      "White Bread",
      "American Cheese",
      "Butter"
    ]
  }
}
]
```

Error Request: Post

<https://localhost:7133/calendars/1?StartDate=2023-03-23&EndDate=2023-03-22>

Error Response: **400**

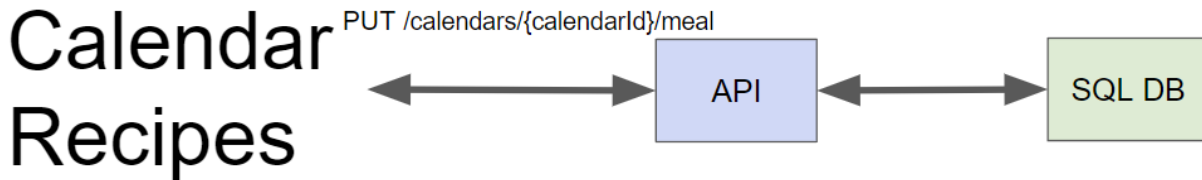
Start date must come before end date

## PUT Requests

### Assign Meal

Route: `"/calendars/{calendarId}/meal"`

Purpose: This endpoint will allow the user to specifically assign a recipe to a given day on either the calendar page or the view recipe page.



Successful Request 1: PUT <https://localhost:7133/calendars/1/meal>

```
Body: {
  "mealId": 1,
  "mealDate": "2023-03-23",
  "recipeId": 1
}
```

Response 1: **204**

Successful Request 2: PUT <https://localhost:7133/calendars/1/meal>

```
Body: {
  "mealDate": "2023-03-23",
  "recipeId": 1
}
```

Response 2: **201** (Still no content)

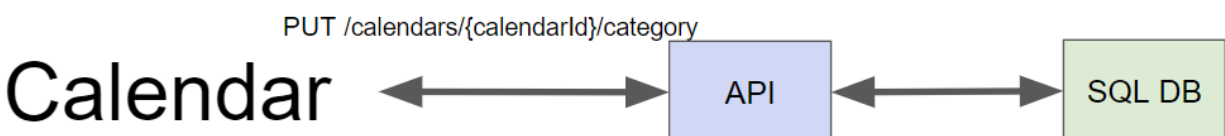
Error Request: <https://localhost:7133/calendars/1/meal>

```
Body: {}
ErrorResponse: 400
RecipeId is required
```

### Update Category

Route: `"/calendars/{calendarId}/category"`

Purpose: This endpoint will allow the user to conveniently change one category when they are on the calendar tab and realize they want a change



Successful Request 1: PUT <https://localhost:7133/calendars/1/category>

```
Body: {  
  "id": 1,  
  "name": "Mystery",  
  "dayOfWeek": 1,  
  "categoryType": 4,  
  "tagId": null,  
  "ingredientId": null  
}
```

Response 1: **204**

Successful Request 2: PUT <https://localhost:7133/calendars/1/category>

```
Body:{  
  "id": null,  
  "name": "Mystery",  
  "dayOfWeek": 1,  
  "categoryType": 4,  
  "tagId": null,  
  "ingredientId": null  
}
```

Response 2: **201** (Still no content)

Error Request: <https://localhost:7133/calendars/1/category>

```
Body: {  
  "id": null,  
  "name": "Mystery",  
  "dayOfWeek": 1,  
  "tagId": null,  
  "ingredientId": null  
}
```

ErrorResponse: **400**

CategoryType is required

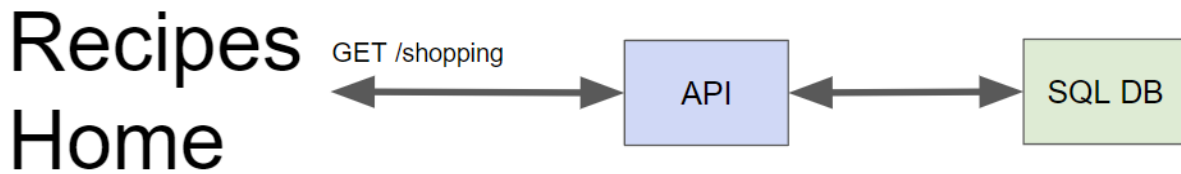
# Shopping List Controller “/shopping”

## GET Requests

### Retrieve Shopping List

Route: “/shopping”

Purpose: This endpoint will be called when the user goes to the shopping list page and will return a list if one already exists for the user. This endpoint will also be used on the home page to fill in the shopping list section



Successful Request GET <https://localhost:7133/shopping>

Successful Response: **200**

```
{
  "id": 1,
  "startDate": "2023-03-23",
  "endDate": "2023-03-31",
  "createdOn": "2023-03-23T23:51:23.4071655-04:00",
  "generatedItems": [
    {
      "id": 1,
      "isChecked": false,
      "ingredientId": 1,
      "name": "Ground Beef",
      "amount": 3,
      "measurementType": 9
    },
    {
      "id": 2,
      "isChecked": true,
      "ingredientId": 1,
      "name": "Cheddar Cheese",
      "amount": 12,
      "measurementType": 10
    }
  ],
  "enteredItems": [
    {
```



```

    "id": 1,
    "isChecked": false,
    "name": "Mountain Dew",
    "category": 2
  },
  {
    "id": 2,
    "isChecked": true,
    "name": "Tortilla Chips",
    "category": 3
  }
]
}

```

No error response (unless there would be an internal server error)

## POST Requests

### Generate Shopping List

Route: “/shopping”

Purpose: This endpoint will generate a shopping list for the user when requested on the shopping page so that the user can take advantage of the database knowing what ingredients they will need.



Successful Request POST <https://localhost:7133/shopping>

Body: {

```

  "startDate": "2023-03-23",
  "endDate": "2023-03-31"
}

```

Successful Response: **201**

```

{
  "id": 1,
  "startDate": "2023-03-23",
  "endDate": "2023-03-31",
  "createdOn": "2023-03-23T23:57:43.7089553-04:00",
  "generatedItems": [
    {
      "id": 1,
      "isChecked": false,
      "ingredientId": 1,

```

```

    "name": "Ground Beef",
    "amount": 3,
    "measurementType": 9
  },
  {
    "id": 2,
    "isChecked": true,
    "ingredientId": 1,
    "name": "Cheddar Cheese",
    "amount": 12,
    "measurementType": 10
  }
],
"enteredItems": []
}

```

Error! Request POST <https://localhost:7133/shopping>

Body: {  
 "startDate": "2023-03-23",  
 "endDate": "2023-03-22"  
}

Error Response: **400**

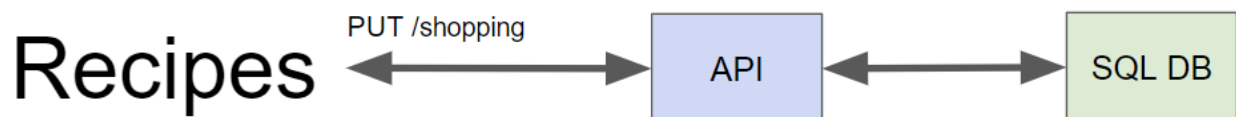
Start date must come before end date

## PUT Requests

### Update Shopping List

Route: "/shopping"

Purpose: This endpoint will allow the user to add, edit, and delete items from the shopping list as they see fit.



Successful Request PUT <https://localhost:7133/shopping>

Body: {  
 "id": 1,  
 "generatedItemIdsToCheck": [  
 1,  
 2  
 ],  
 "generatedItemIdsToUnCheck": [  
 3  
 ]  
}

```
],
"generatedItemIdsToDelete": [
  4
],
"enteredItemsToUpdate": [
  {
    "id": 1,
    "isChecked": true,
    "name": "Mountain Dew",
    "category": 2
  }
],
"enteredItemIdsToDelete": [
  1
]
}
Successful Response: 204
Error Request: PUT https://localhost:7133/shopping
Body: {
  "id": 1,
  "enteredItemsToUpdate": [
    {
      "id": -1,
      "isChecked": true,
      "name": "Mountain Dew",
      "category": 2
    }
  ]
}
```

Error Response: **400** Entered Item Id did not exist