

# Work on project. Stage 2/5: Multiple recipes

Project: [Recipes](#)

## Multiple recipes

 Hard  58 minutes  670 users solved this stage. Latest completion was **less than a minute ago**.











### Description

Our service can store only one recipe at a time which is not very convenient. In this stage, improve the service to store a lot of recipes and access recipes by a unique `id`. Some changes in the recipe structure are also required.

The new structure of a recipe includes the same 4 fields, but the type of two of them is different. `ingredients` and `directions` should now be arrays. Here's an example of the new structure:

```
{
  "name": "Warming Ginger Tea",
  "description": "Ginger tea is a warming drink for cool weather, ...",
  "ingredients": ["1 inch ginger root, minced", "1/2 lemon, juiced", "1/2 tea bag"],
  "directions": ["Place all ingredients in a mug and fill with warm water (or 10 minutes", "Drink and enjoy"]
}
```

### 6 / 6 Prerequisites

-  [IoC Container](#) 
-  [Spring beans](#) 
-  [Interfaces](#) 
-  [Spring components](#) 
-  [Exception handling](#) 

Show all

### [Join a study group for the project Recipes](#)

Discuss your current project with fellow learners and help each other.

### Objectives

Rearrange the existing endpoints; the service should support the following:

- `POST /api/recipe/new` receives a recipe as a JSON object and returns a JSON object with one `id` field. This is a uniquely generated number by which we can identify and retrieve a recipe later. The status code should be `200 (Ok)`.
- `GET /api/recipe/{id}` returns a recipe with a specified `id` as a JSON object (where `{id}` is the `id` of a recipe). The server should respond with the `200 (Ok)` status code. If a recipe with a specified `id` does not exist, the server should respond with `404 (Not found)`.

### Examples

**Example 1:** `POST /api/recipe/new` request with the following body:

```
{
  "name": "Fresh Mint Tea",
  "description": "Light, aromatic and refreshing beverage, ...",
  "ingredients": ["boiled water", "honey", "fresh mint leaves"],
  "directions": ["Boil water", "Pour boiling hot water into a mug", "Add fresh mint leaves", "Mix and let the mint 5 minutes", "Add honey and mix again"]
}
```

Response:

```
{
  "id": 1
}
```

**Example 2:** `GET /api/recipe/1` request


Response:

```
{
  "name": "Fresh Mint Tea",
  "description": "Light, aromatic and refreshing beverage, ...",
  "ingredients": ["boiled water", "honey", "fresh mint leaves"],
  "directions": ["Boil water", "Pour boiling hot water into a mug", "Add fresh mint leaves", "Mix and let the mint 5 minutes", "Add honey and mix again"]
}
```

**Example 3:** `GET /api/recipe/999` request

Status code: `404 (Not found)`

 [Report a typo](#)

 See hint

↙ Write a program

**32** users liked this problem. **0** didn't like it. **What about you?**



Continue

[Solutions \(117\)](#)

[Comments \(26\)](#)

[Hints \(7\)](#)

[Useful links \(0\)](#)

[Solutions \(117\)](#)

[Show discussion](#)