

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа №2

Выполнил:

Борисова Элина

Группа К3341

Проверил:

Добряков Д. И.

Санкт-Петербург

2025 г.

Задача

По выбранному варианту необходимо будет реализовать RESTful API средствами express + typescript (используя ранее написанный boilerplate).

Ход работы

В данной работе было добавлено следующее:

```
export const updateRecipe = async (req, res) => {
  try {
    const recipeRepository = AppDataSource.getRepository(Recipe);
    const recipe = await recipeRepository.findOne({
      where: { id: req.params.id },
      relations: ["ingredients"],
    });

    if (!recipe) return res.status(404).json({ error: "Recipe not found" });

    const { title, description, difficulty, ingredients } = req.body;
    if (title) recipe.title = title;
    if (description) recipe.description = description;
    if (difficulty) recipe.difficulty = difficulty;

    if (ingredients && Array.isArray(ingredients)) {
      const ingredientRepository = AppDataSource.getRepository(RecipeIngredient);
      for (const ingredientData of ingredients) {
        const ingredient = await ingredientRepository.findOne({
          where: { id: ingredientData.id }
        });

        if (!ingredient) {
          return res.status(404).json({ error: `Ingredient with id ${ingredientData.id} not found` });
        }

        let recipeIngredient = await ingredientRepository.findOne({
          where: { recipe_id: req.params.id, ingredient_id: ingredient.id }
        });

        if (recipeIngredient) {
          recipeIngredient.quantity = ingredientData.quantity;
          await ingredientRepository.save(recipeIngredient);
        } else {
          recipeIngredient = ingredientRepository.create({
            recipe_id: req.params.id,
            ingredient_id: ingredient.id,
            quantity: ingredientData.quantity,
          });
          await ingredientRepository.save(recipeIngredient);
        }
      }
    }
  }
}
```

```

        const updatedRecipe = await recipeRepository.save(recipe);

        AppDataSource.getRepository(RecipeIngredient)
            .createQueryBuilder("recipeIngredient")
            .innerJoin(recipeIngredient, "ri", "ri.ingredient_id = :ingredientId")
            .where("ri.recipe_id = :recipeId", { recipeId: req.params.id })
            .getMany();

        return res.json(updatedRecipe);
    } catch (error) {
        return res.status(500).json({ error: error.message });
    }
};

```

```

export enum RecipeDifficulty {

    VERY_EASY = 1,

    EASY = 2,

    MEDIUM = 3,

    HARD = 4,

    VERY_HARD = 5

}

```

Обновим рецепт и добавим ингредиенты:

The screenshot shows a REST client interface with a PUT request to `http://localhost:3000/recipes/7556330d-2d5a-4430-8808-fda91941018a`. The request body is a JSON object with the following fields:

```
{
  "title": "Шоколадный торт",
  "description": "Самый простой шоколадный торт с богатым вкусом",
  "difficulty": 2
}
```

The response is a 200 OK status with a JSON body containing the following fields:

```
{
  "id": "7556330d-2d5a-4430-8808-fda91941018a",
  "title": "Шоколадный торт",
  "description": "Самый простой шоколадный торт с богатым вкусом",
  "image_url": null,
  "video_url": null,
  "difficulty": 2,
  "created_at": "2025-04-20T18:13:13.200Z",
  "ingredients": [
    {
      "recipe_id": "7556330d-2d5a-4430-8808-fda91941018a",
      "ingredient_id": "a1ea91b2-5236-42c7-b921-0a6407bbc6c0",
      "quantity": "200 г"
    },
    {
      "recipe_id": "7556330d-2d5a-4430-8808-fda91941018a",
      "ingredient_id": "6405f4b9-307a-43c3-81f6-461255f1e8d0",
      "quantity": "300 г"
    }
  ]
}
```

Вывод

В данной работе был реализован RESTful API с использованием Express и TypeScript на основе заранее подготовленного boilerplate-проекта.