## Тестирование REST API

* GET /product/id – нужно добавить тело ответа для случая, если по данному id не найден продукт
* GET /recipe/id – нужно добавить тело ответа для случая, если по данному id не найден рецепт
* GET /cpfc - нужно добавить тело ответа для случая, если по данному id не будет найден ингредиент
* POST /recipe, POST /product, DELETE /product/id, DELETE /recipe/id – в теле запроса нет id, по которому можно будет определить, от кого исходит запрос. Любой человек в сети сможет менять данные в ЬД.
* POST /product – в теле запроса должна быть информация о продукте.
* DELETE /reciepe/id, DELETE /product/id – нужно продумать, какими правами нужно обладать, чтобы удалить продукт или рецепт в общую базу данных
* JSON “Информация о блюде” не нужен, уже есть “Информация о продукте” и “Информация о рецепте”
* GET /diet - предусмотреть тело ответа для невыполнимого случая, когда пользователь хочет похудеть на 20 кг за 1 день (например)

Ниже представлена исходная версия API, для которой производилось тестирование.

# RESTful API Запросы

1. /product (GET) Поиск продуктов по названию;
   1. Тело запроса:

* Название продукта
  1. Тело ответа:
* Список из продуктов, подходящих по названию

{

[

id: int,

name: string,

]

}

1. /product/id (GET) Получение информации о продукте;
   1. Тело запроса:

* ID продукта
  1. Тело ответа:
* Информация о продукте

1. /recipe (GET) Поиск продуктов (рецептов) по названию;
   1. Тело запроса:

* Название рецепта
  1. Тело ответа:
* Список из продуктов, подходящих по названию

{[

id: Int,

name: String,

]}

1. /recipe/id (GET) Получение информации о рецепте;
   1. Тело запроса:

* ID рецепта
  1. Тело ответа:
* Информация о рецепте

1. /cpfc (GET) Расчёт КБЖУ и микронутриентов блюда по ингредиентам;

Тело запроса:

* Состав блюда
  1. Тело ответа:
* Информация о блюде

1. /product (POST) Добавление нового продукта в общую базу данных;
   1. Тело запроса:

* Пищевая ценность продукта
  1. Тело ответа:
* ID продукта

1. /receipe (POST) Добавление нового рецепта в общую базу данных;
   1. Тело запроса:

* Информация о рецепте
  1. Тело ответа:
* ID рецепта

1. /product/id (DELETE) Удаление продукта из общей базы данных;
   1. Тело запроса:

* ID продукта
  1. Тело ответа:
* Результат (String)

1. /receipe/id (DELETE) Удаление рецепта из общей базы данных;
   1. Тело запроса:

* ID рецепта
  1. Тело ответа:
* Результат (String)

1. /dci (GET) Расчёт СНК (Суточная норма калорий);
   1. Тело запроса:

* Информация о человеке
  1. Тело ответа:
* ccal: Int

1. /rda (GET) Расчёт РСК (Рекомендуемое суточное количество);
   1. Тело запроса:

* RDA request body
  1. Тело ответа:
* Diet goal

1. /analize (GET) - Анализ дневного/недельного/месячного рациона;
   1. Тело запроса:

{

diet: Diet

goal: diet goal

}

* 1. Тело ответа:

{

ccal\_control: Int

carbo\_control: Float

protein\_control: Float

fat\_control: Float

micro\_nutrients: [Float]

}

1. /diet (GET) - Формирование рациона;
   1. Тело запроса:

* Diet goal
  1. Тело ответа:
* Diet

1. /deliver (POST) - Оформление подписки на доставку;
   1. Тело запроса:

{

diet: Diet

budget: Float

deliver\_time: BigInt

}

* 1. Тело ответа:
* Результат (String)

1. /deliver (GET) - Проверить подписку на доставку;
   1. Тело запроса:

* ID пользователя
  1. Тело ответа:

{

diet: Diet

budget: Float

deliver\_time: BigInt

}

1. /deliver (DELETE) – Отменить подписку на доставку;
   1. Тело запроса:

* ID пользователя
  1. Тело ответа:
* Результат (String)

1. /deliver (PUT) – Отредактировать доставку
   1. Тело запроса:

* ID пользователя
* Diet
  1. Тело ответа:
* Результат (String)

JSONы

* Пищевая ценность еды:

{

ccal: Int,

carbo: Float,

protein: Float,

fat: Float,

micro\_nutrients: [Float]

}

* Состав еды:

{

Ingredients {[

product\_id: Int

portion: Int

]}

}

* Информация о продукте

{

Id: Int

nutritional\_value

}

* Информация о блюде

{

Id: Int

ingredients(состав)

nutritional\_value(пищевая ценность)

}

* Информация о рецепте

{

Id: Int

ingredients(состав)

nutritional\_value(пищевая ценность)

instruction {[

description: String

]}

cook\_time: Int

prepare\_time: Int

}

* Информация о человеке

{

id: Int

weight: Float

height: Int

activity\_level: Float

}

* RDA request body

{

human: информация о человеке

goal: diet goal

}

* Diet goal

{

days: Int

weight\_delta: float

ccal\_delta: Int

}

* Diet

{

food: [

{

time: BigInt

food: Информация о рецепте/

Информация о блюде/

Информация о продукте

portion: Int

}

]

}