# 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. GET /product/id

Получение информации о продукте по ID.

Request Body: { "id": 1 }

Response Body:

Если продукт не найден:

{ "error": "Product not found for the given ID" }

Если продукт найден:

{

"product":

{

"id": 1,

"nutritional\_value": {Nutritional value}

}

}

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. GET /recipe/id

Получение информации о рецепте по ID.

Request Body: { "id": 1 }

Response Body:

Если рецепт не найден: { "error": "Recipe not found for the given ID" }

Если рецепт найден:

{

"recipe":

{

"id": 1,

"ingredients": [Ingredients],

"nutritional\_value": {Nutritional value}

}

}

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

Тело запроса: Состав блюда

Тело ответа:

Если ингредиент не найден:

{

"error": "Some ingredients not found for the given ID"

}

Если найдены все ингредиенты:  
{

ccal\_control: Int

carbo\_control: Float

protein\_control: Float

fat\_control: Float

micro\_nutrients: [Float]

}

1. /product (POST) Добавление нового продукта в общую базу данных;
   1. Authentication header: Access token
   2. Тело запроса:

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

1. /recipe (POST) Добавление нового рецепта в общую базу данных;
   1. Authentication header: Access token
   2. Тело запроса:

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

1. /product/id (DELETE) Удаление продукта из базы данных;
   1. Authentication header: Access token
   2. Тело запроса:

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

1. /recipe/id (DELETE) Удаление рецепта из базы данных;
   1. Authentication header: Access token
   2. 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;

Request Body: { "goal": {Diet goal} } (цели)

Response Body:

В случае выполнимого результата: { "diet": {Diet} } (диета)

В случае невыполнимого: "error": "Impossible goal"

1. /deliver (POST) - Оформление подписки на доставку;
   1. Authentication header: Access token
   2. 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

delivery\_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)

1. /auth (POST) – Аутентификация пользователя

Request Body:

{

"email": [user@example.com](mailto:user@example.com),

"access string": "djsdjhgskdhfsdfnjdfkj" (зашифрованные почта и пароль)

}

Response Body:

Если верификация для пользователя включена:

{

"Access Token" (в cookie): "ldasjdjahsdkjahdkag",

"Refresh Token" (в cookie): "sldfkjhsdkjfhskdjfhskjdhfkjsh"

}

Если выключена:

{

"error": "Verification for this user was disabled"

}

1. /registration (POST) – регистрация пользователя

Request Body:

{

"email": [user@example.com](mailto:user@example.com),

"access string": "djsdjhgskdhfsdfnjdfkj" (зашифрованные почта и пароль)

}

Response Body:

Если такого пользователя нет:

{ "code": 200}

Если уже есть:

{

"error": "User with this email already exists",

"code": 409 (Conflict)/200

}

JSONы

* Пищевая ценность еды:

{

ccal: Int,

carbo: Float,

protein: Float,

fat: Float,

nutritional\_value: Float[]

}

* Состав еды:

{

Ingredients {[

product\_id: Int,

portion: Float,

price: Float

]}

}

* Информация о продукте

{

Id: Int

nutritional\_value

}

* 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

}

]

}