# **Мобильное приложение для ОС Android “Рецепты”**

1. Концептуальная модель:

Наименование: “Рецепты”

Общее описание: приложение для просмотра рецептов.

Особенности приложения:

1. возможность добавлять и сохранять пометки внутри описания способа приготовления блюда;
2. дизайн, рассчитанный на просмотр во время готовки.
3. Функциональная карта

Необходимые функции:

1. Вход/регистрация по логину или почте;
2. Выделение рецептов блюд в разделы;
3. Поиск рецептов блюд по названию;
4. Фильтр рецептов блюд по разделам, времени приготовления, калорийности на 100 г, ингредиентам, оценке, времени добавления;
5. Сохранение рецептов в избранное;
6. Оценка рецепта;
7. Добавление рецепта (с настройкой публичности);
8. Отображение состава блюда и описания шагов по его приготовлению;
9. Сохранение пометок к блюду.
10. Экраны

Точки входа представлены на диаграмме в разделе “Путь пользователя”.

1. **Загрузочный**. Отображается при входе в приложение.

Представляет собой логотип, но однородном фоне.

Во время загрузки использовать запрос /categries?page=cur\_page&size=cur\_size и передать полученные данные на главный экран после загрузки.

Действия при разных ответах сервера:

1. Успех
   1. 2xx
      1. 200: переходим на следующий экран и передаем полученную в ответе информацию.
      2. 204: переходим на следующий экран и передаем полученную в ответе информацию.
2. Неуспех
   1. 4xx
      1. 404: повторяем попытку. По истечении 5 секунд переходим на следующий экран и передаем полученную в ответе информацию.
   2. 5xx: повторяем попытку. По истечении 5 секунд переходим на следующий экран и передаем полученную в ответе информацию.
3. **Главный**. Доступен всем пользователям.

Представляет собой экран с иконками блюд по разделам, при нажатии на который происходит переход на экран “Раздел”.

Элементы:

1. Сверху слева отображается название экрана – “Главная”;
2. Таблицы с иконками *Раздел*;
3. Снизу отображается *меню 1*.

Во время загрузки и при скроллинге использовать запрос /categries?page=cur\_page&size=cur\_size.

Действия при разных ответах сервера:

1. Успех
   1. 2xx
      1. 200: отображаем описанный выше экран.
      2. 204: отображаем экран, заполняя запрашиваемые поля пустым содержимым или содержимым сохраненной информации.
2. Неуспех
   1. 4xx
      1. 404: отображаем экран, заполняя запрашиваемые поля пустым содержимым или содержимым сохраненной информации.
   2. 5xx: отображаем всплывающее сообщение с текстом: “Сервер временно недоступен. Пожалуйста, попробуйте позже.” Отображаем экран, заполняя запрашиваемые поля пустым содержимым или содержимым сохраненной информации.
3. **Поиск.** Доступен всем пользователям.

Элементы:

1. Сверху слева отображается название экрана – “Поиск”;
2. Поисковая строка. Не исчезает при скроллинге;
3. Под кнопкой из пункта 1 располагается выдвижной список с возможностью выбрать способ сортировки. Не исчезает при скроллинге;
4. Таблицы с иконками *Блюдо*;
5. Снизу отображается меню 1.

Во время загрузки и при скроллинге использовать запрос

/recipes?page=cur\_page&size=cur\_size&sort=name,dirtion&fiter\_info=info

с соответствующими параметрами сортировки (sort) и текстом для поиска (filter\_info).

1. Успех
   1. 2xx
      1. 200: отображаем описанный выше экран.
      2. 204: отображаем экран, заполняя запрашиваемые поля пустым содержимым или содержимым сохраненной информации.
2. Неуспех
   1. 4xx
      1. 404: отображаем экран, заполняя запрашиваемые поля пустым содержимым или содержимым сохраненной информации.
   2. 5xx: отображаем всплывающее сообщение с текстом: “Сервер временно недоступен. Пожалуйста, попробуйте позже.” Отображаем экран, заполняя запрашиваемые поля пустым содержимым или содержимым сохраненной информации.
3. **Избранное.** Доступен всем пользователям.

Элементы:

1. Сверху слева отображается название экрана – “Избранное”;
2. Поисковая строка. Не исчезает при скроллинге;
3. Под кнопкой из пункта 1 располагается выдвижной список с возможностью выбрать способ сортировки. Не исчезает при скроллинге;
4. Таблицы с иконками *Блюдо*, добавленными пользователем в избранное;
5. Снизу отображается *меню 1*.

Во время загрузки и при скроллинге использовать запрос /recipes/favourite?page=cur\_page&size=cur\_size&sort=name,direction&filter\_info=info

с соответствующими параметрами сортировки (sort) и текстом для поиска (filter\_info).

1. Успех
   1. 2xx
      1. 200: отображаем описанный выше экран и сохраняем полученную информацию в кеш.
      2. 204: отображаем экран, заполняя запрашиваемые поля пустым содержимым или содержимым сохраненной информации.
2. Неуспех
   1. 4xx
      1. 404: отображаем экран, заполняя запрашиваемые поля пустым содержимым или содержимым сохраненной информации.
   2. 5xx: отображаем всплывающее сообщение с текстом: “Сервер временно недоступен. Пожалуйста, попробуйте позже.” Отображаем экран, заполняя запрашиваемые поля пустым содержимым или содержимым сохраненной информации.
3. **Профиль.** Доступно всем пользователям.
4. Для авторизованных пользователей:
   1. Сверху слева отображается название экрана – “Профиль”;
   2. Под ним имя пользователя;
   3. Под ним почта;
   4. Под ней пароль;
   5. Под ним кнопка добавить рецепт;
   6. Кнопки мои рецепта;
   7. Снизу отображается *меню 1*.
5. Для не авторизованных пользователей:
   1. Сверху слева отображается название экрана – “Профиль”;
   2. Под ним кнопки “Войти” и “Зарегистрироваться”, которые ведут на соответствующие экраны;
   3. Снизу отображается *меню 1*.

При загрузке использовать /account/profile.

При изменении логина использовать /account / change\_login.

При изменении email использовать /account / change\_email.

Для проверки пароля перед изменением использовать метод /account / check\_password.

При изменении пароля использовать /account / change\_password.

1. Успех
   1. 2xx
      1. 200: отображаем описанный выше экран и сохраняем полученную информацию в кеш.
      2. 204: отображаем экран, заполняя запрашиваемые поля пустым содержимым или содержимым сохраненной информации.
2. Неуспех
   1. 4xx
      1. 404: отображаем экран, заполняя запрашиваемые поля пустым содержимым или содержимым сохраненной информации.
   2. 5xx: отображаем всплывающее сообщение с текстом: “Сервер временно недоступен. Пожалуйста, попробуйте позже.” Отображаем экран, заполняя запрашиваемые поля пустым содержимым или содержимым сохраненной информации.
3. **Раздел.** Доступен всем пользователям.

Представляет собой экран с иконками блюд, при нажатии на который происходит переход на экран “Блюдо” соответствующего блюда.

Элементы:

1. Слева наверху имеется кнопка *Кнопка 1 “Назад”*. Исчезает при скроллинге вниз, появляется при скроллинге вверх одновременно со списком из пункта 2;
2. Под кнопкой из пункта 1 располагается выдвижной список с возможностью выбрать способ сортировки. Исчезает при скроллинге вниз, появляется при скроллинге вверх одновременно с кнопкой из пункта 1;
3. Иконки *Блюдо* в виде таблицы;
4. Внизу страницы имеется кнопка возврата к началу страницы.

Во время загрузки и при скроллинге использовать запрос

/recipes?page=cur\_page&size=cur\_size&sort=name,dirtion&fiter\_info=info

с соответствующими параметрами сортировки (sort) и текстом для поиска (filter\_info) с дополнительным параметром section.

1. Успех
   1. 2xx
      1. 200: отображаем описанный выше экран.
      2. 204: отображаем экран, заполняя запрашиваемые поля пустым содержимым или содержимым сохраненной информации.
2. Неуспех
   1. 4xx
      1. 404: отображаем экран, заполняя запрашиваемые поля пустым содержимым или содержимым сохраненной информации.
   2. 5xx: отображаем всплывающее сообщение с текстом: “Сервер временно недоступен. Пожалуйста, попробуйте позже.” Отображаем экран, заполняя запрашиваемые поля пустым содержимым или содержимым сохраненной информации.
3. **Блюдо.** Доступен всем пользователям.

Элементы

1. Слева наверху имеется кнопка *Кнопка 1 “Назад”*, которая не пропадает при скроллинге;
2. Фотография блюда;
3. Рейтинг. Кликабельный. Переводит на соответствующий экран **Оценка**;
4. Краткое описание блюда;
5. Состав;
6. Шаги приготовления. Название шага кликабельно. Ведет на экран **Добавление / редактирование комментария**. Под каждым шагом находятся комментарии пользователя(при наличии). Они так же кликабельны;
7. Внизу страницы имеется кнопка возврата к началу страницы.

Во время загрузки и при скроллинге использовать запрос /recipes?recipe\_id=id.

1. Успех
   1. 2xx
      1. 200: отображаем описанный выше экран
      2. 204: отображаем экран, заполняя запрашиваемые поля пустым содержимым или содержимым сохраненной информации.
2. Неуспех
   1. 4xx
      1. 404: отображаем экран, заполняя запрашиваемые поля пустым содержимым или содержимым сохраненной информации.
   2. 5xx: отображаем всплывающее сообщение с текстом: “Сервер временно недоступен. Пожалуйста, попробуйте позже.” Отображаем экран, заполняя запрашиваемые поля пустым содержимым или содержимым сохраненной информации.
3. **Оценка**. Доступен только авторизованным пользователям. Появляется поверх изначального окна.

Элементы:

1. Слева наверху имеется наименование блюда;
2. Кнопка в виде 5 звезд. Кликабельна. Позволяет выставить оценку;
3. Снизу слева располагается кнопка “Ок”, которая сохраняет текущую оценку и возвращает на предыдущий экран;
4. Снизу справа располагается кнопка “Отмена”, которая возвращает на предыдущий экран.

При загрузке оценки использовать метод /recipe/get\_rating.

При добавлении (изменении) оценки использовать метод /recipe/set\_rating?recipe\_id=id.

1. Успех
   1. 2xx
      1. 200: отображаем описанный выше экран
      2. 204: отображаем экран, заполняя запрашиваемые поля пустым содержимым или содержимым сохраненной информации.
2. Неуспех
   1. 4xx
      1. 404: отображаем экран, заполняя запрашиваемые поля пустым содержимым или содержимым сохраненной информации.
   2. 5xx: отображаем всплывающее сообщение с текстом: “Сервер временно недоступен. Пожалуйста, попробуйте позже.” Отображаем экран, заполняя запрашиваемые поля пустым содержимым или содержимым сохраненной информации.
3. **Добавление / редактирование комментария**. Доступен всем пользователям. Появляется поверх изначального окна.

Элементы:

1. Слева наверху имеется наименование шага;
2. Текстовое поле для ввода комментария;
3. Снизу слева располагается кнопка “Ок”, которая сохраняет текущий комментарий и возвращает на предыдущий экран;
4. Снизу справа располагается кнопка “Отмена”, которая возвращает на предыдущий экран.

Комментарии доступны только с экрана блюдо, поэтому информация о них при загрузке будет иметься.

При добавлении комментария использовать метод /recipes/steps/add\_comment?recipe\_id=id&step\_id=id.

При изменении комментария использовать метод /recipes/steps/update\_comment?recipe\_id=id&step\_id=id&comment\_id=id.

При удалении комментария использовать метод /recipes/steps/delete\_comment?recipe\_id=id&step\_id=id&comment\_id=id.

1. Успех
   1. 2xx
      1. 200: отображаем описанный выше экран
      2. 204: отображаем экран, заполняя запрашиваемые поля пустым содержимым или содержимым сохраненной информации.
2. Неуспех
   1. 4xx
      1. 404: отображаем экран, заполняя запрашиваемые поля пустым содержимым или содержимым сохраненной информации.
   2. 5xx: отображаем всплывающее сообщение с текстом: “Сервер временно недоступен. Пожалуйста, попробуйте позже.” Отображаем экран, заполняя запрашиваемые поля пустым содержимым или содержимым сохраненной информации.
3. **Добавление / редактирование блюда.** Доступен только авторизованным пользователям.

Элементы:

1. Слева наверху имеется кнопка *Кнопка 1 “Назад”,* которая не пропадает при скроллинге;
2. Справа наверху имеется кнопка “Сохранить”, которая не пропадает при скроллинге. По нажатии сохраняет рецепт и возвращает пользователя на предыдыщий экран;
3. Редактируемое поле названия;
4. Редактируемое поле фотографии;
5. Таблица 2 \* n, где первая колонка – название ингредиента, вторая – количество с редактируемыми полями, которые можно удалять и добавлять;
6. Редактируемое по разделам описание.

Для добавления использовать метод /recipes/add.

Для обновления использовать метод /recipes/update?recipe\_id=id.

Для удаления использовать метод /recipes/delete?recipe\_id=id.

1. Успех
   1. 2xx
      1. 200: отображаем описанный выше экран
      2. 204: отображаем экран, заполняя запрашиваемые поля пустым содержимым или содержимым сохраненной информации.
2. Неуспех
   1. 4xx
      1. 404: отображаем экран, заполняя запрашиваемые поля пустым содержимым или содержимым сохраненной информации.
   2. 5xx: отображаем всплывающее сообщение с текстом: “Сервер временно недоступен. Пожалуйста, попробуйте позже.” Отображаем экран, заполняя запрашиваемые поля пустым содержимым или содержимым сохраненной информации.
3. **Регистрация**. Доступно всем пользователям.

Элементы:

1. По центру надпись “Регистрация”;
2. Поле для ввода логина;
3. Поле для ввода почты;
4. Поле для ввода пароля;
5. Кнопка “Регистрация”, регистрирующая пользователя и перенаправляющая на предыдущий экран;
6. Кликабельный текст “Авторизация”, перенаправляющий на экран **Авторизация**.

Для регистрации использовать метод /account/register.

1. Успех
   1. 2xx
      1. 200: отображаем описанный выше экран
      2. 204: отображаем экран, заполняя запрашиваемые поля пустым содержимым или содержимым сохраненной информации.
2. Неуспех
   1. 4xx
      1. 404: отображаем экран, заполняя запрашиваемые поля пустым содержимым или содержимым сохраненной информации.
   2. 5xx: отображаем всплывающее сообщение с текстом: “Сервер временно недоступен. Пожалуйста, попробуйте позже.” Отображаем экран, заполняя запрашиваемые поля пустым содержимым или содержимым сохраненной информации.
3. **Авторизация.** Доступно всем пользователям.

Элементы:

1. По центру надпись “Авторизация”;
2. Поле для ввода логина или почты;
3. Поле для ввода пароля;
4. Кнопка “Авторизация”, авторизирующая пользователя и перенаправляющая на предыдущий экран;
5. Кликабельный текст “Регистрация”, перенаправляющий на экран **Регистрация**.

Для авторизации использовать метод /account/authorization.

1. Успех
   1. 2xx
      1. 200: отображаем описанный выше экран
      2. 204: отображаем экран, заполняя запрашиваемые поля пустым содержимым или содержимым сохраненной информации.
2. Неуспех
   1. 4xx
      1. 404: отображаем экран, заполняя запрашиваемые поля пустым содержимым или содержимым сохраненной информации.
   2. 5xx: отображаем всплывающее сообщение с текстом: “Сервер временно недоступен. Пожалуйста, попробуйте позже.” Отображаем экран, заполняя запрашиваемые поля пустым содержимым или содержимым сохраненной информации.

**Описание элементов**

*Описание меню 1:*

1. Находится внизу экрана в виде иконок с подписями: главная, поиск, избранное, профиль;
2. Иконка текущего экрана изображается темнее;
3. При нажатии на соответствующую иконку происходит переход на соответствующий экран.

*Кнопка 1 “Назад”:*

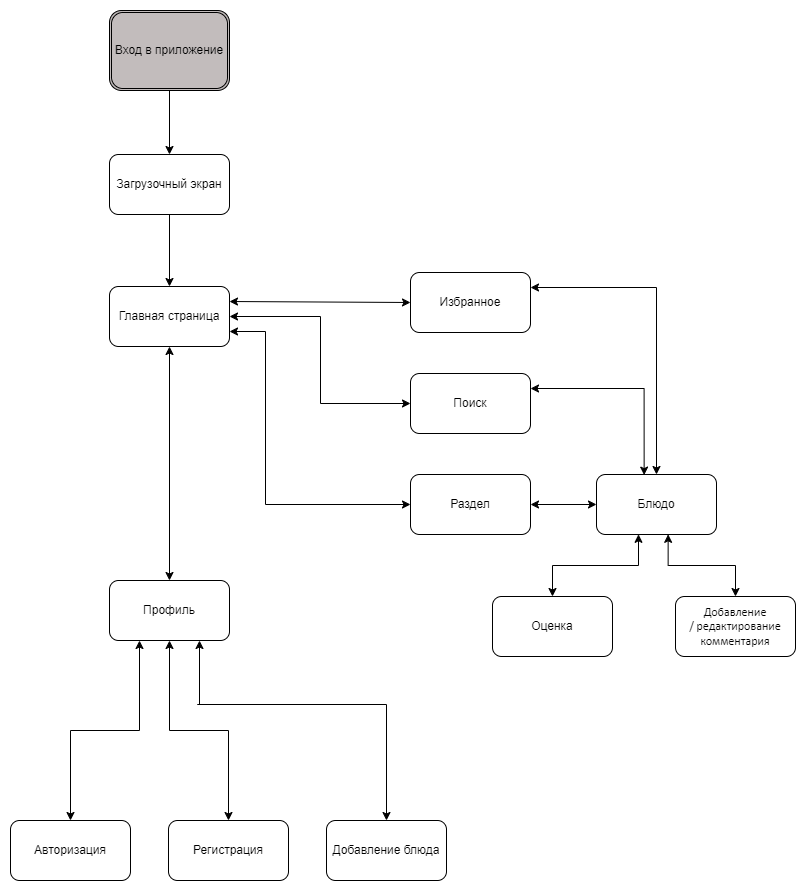
1. Располагается в левом верхнем углу;
2. Видна в виде стрелки;

*Иконка “Раздел”:*

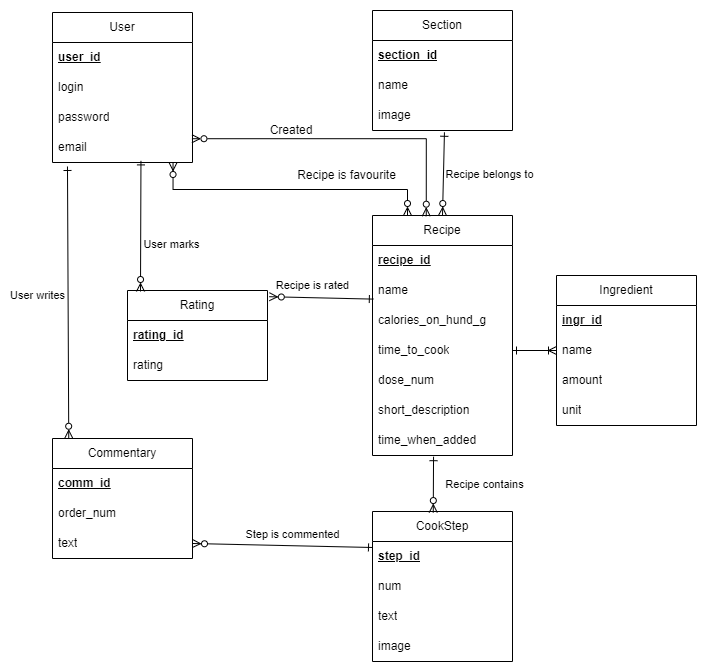
1. Прямоугольной формы;
2. Сверху располагается картинка;
3. Под ней располагается название раздела;
4. Кликабельна. Ведет на соответствующий экран **Раздел**.

*Иконка “Блюдо”:*

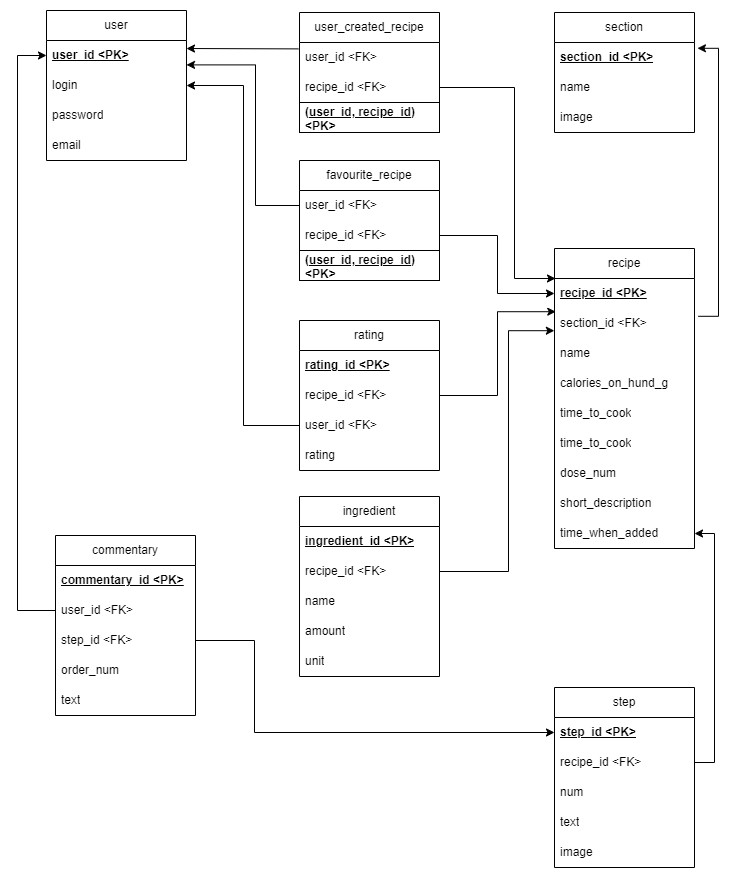
1. Прямоугольной формы;
2. Сверху располагается картинка;
3. Под ней располагается название блюда;
4. Мелким шрифтом ниже отображается полностью или частично краткое описание;
5. В правом нижнем углу располагается кнопка в виде сердечка для добавления в избранное(серая)/удаления из избранного(красная), то есть меняет цвет в зависимости от состояния;
6. В левом нижнем углу располагается рейтинг;
7. Кликабельна. Ведет на соответствующий экран **Блюдо**.
8. Путь пользователя



1. ER-диаграмма



1. Схема базы данных



1. Используемые методы:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Функция | Описание | API  метода | Тип  запроса | Обязательность | Клиент ->  Сервер | Сервер -> Клиент |
| Авторизация | Авторизация | /account/  authorization | GET | + | Имя пользователя (или логин), пароль | Токен |
| Регистрация | Регистрация | /account /  register | POST | + | Имя пользователя, логин, пароль | Токен |
| Изменение логина пользователя | Изменяет логин пользователя | /account / change\_login | PUT | + | Токен, логин | - |
| Изменение почты пользователя | Изменяет email пользователя | /account / change\_email | PUT | + | Токен, email | - |
| Проверка пароля | Сравнивает сохраненный на сервере пароль с отправленным | /account / check\_password | GET | + | Токен, пароль | Токен смены пароля |
| Изменение пароля пользователя | Изменяет парольпользователя | /account / change\_password | PUT | + | Токен смены пароля, токен, пароль | - |
| Проверка авторизации | Проверяет авторизован ли пользователь | /account/profile | GET | + | Токен | Имя пользователя, email |
| Получить список категорий | Возвращает список категорий | /catego­ries?page=cur\_page&size=cur\_size | GET | + | - | Спи­сок кате­горий |
| Получить список рецептов | Возвращает список рецептов | /recipes?page=cur\_page&size=cur\_size&sort=name,direction&filter\_info=info | GET | + | - | Список рецептов |
| Получить список избранных рецептов | Возвращает список рецептов | /recipes/favourite?page=cur\_page&size=cur\_size&sort=name,direction&filter\_info=info | GET | + | Токен | Список избранных рецептов |
| Получить рецепт | Возвращает информацию о рецепте | /recipes?recipe\_id=id | GET | + | ID рецепта, токен | Информация о рецепте + комментарии, если передан токен |
| Добавление комментария | Добавляет комментарий к разделу | /recipes/steps/add\_comment?recipe\_id=id&step\_id=id | POST | + | Токен, комментарий | Добавляет комментарий к разделу в рецепте |
| Обновление комментария | Обновляет комментарий к разделу | /recipes/steps/update\_comment?recipe\_id=id&step\_id=id&comment\_id=id | PUT | + | Токен | Обновляет комментарий к разделу в рецепте |
| Удаление комментария | Удаляет комментарий к разделу | /recipes/steps/delete\_comment?recipe\_id=id&step\_id=id&comment\_id=id | DELETE | + | Токен | Удаляет комментарий к разделу в рецепте |
| Добавление рецепта | Добавляет рецепт от пользователя | /recipes/add | POST | + | Токен, json представление класса Recipe | Добавляет рецепт |
| Удалениерецепта | Удаляет от пользователя | /recipes/delete?recipe\_id=id | DELETE | + | Токен | Удаляет рецепт |
| Обновление рецепта | Обновляет рецепт от пользователя | /recipes/update?recipe\_id=id | PUT | + | Токен, json представление с измененными полями класса Recipe | Обновляет рецепт |
| Добавление ингредиента | Добавляет ингредиент в рецепт | /recipees/add\_ingredient? recipe\_id=id | POST | + | Токен, json представление класса Ingredient | Добавляет ингредиент в рецепт |
| Удаление ингредиента | Удаляет ингредиент из рецепта | /recipees/delete\_ingredient?ingredient\_id=id | DELETE | + | Токен | Добавляет ингредиент в рецепт |
| Обновление ингредиента | Обновляет ингредиент в рецепт | /recipees/update\_ingredient? ingredient=id | PUT | + | Токен, json представление с измененными полями класса Ingredient | Обновляет ингредиент в рецепт |
| Изменение числа порций | Изменяет текущее количество порций и пересчитывает количество ингредиентов | /recipees/dose\_calculation?recipe\_id=id&doses=num | PUT | + | - | Список ингредиентов с перечисленным количеством |
| Получить оценку | Проверяетоценку блюда | /recipe/get\_rating?recipe\_id=id | GET | + | - | Оценка |
| Установить оценку | Добавляет (изменяет) оценку блюду | /recipe/set\_rating?recipe\_id=id | POST | + | Токен, оценка | - |

1. Параметры запросов и ответов
2. /account/authorization

**Параметры запроса:**

GET host/account/authorization HTTP/1.1

Host: host

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| name | - | имя пользователя | string | 123456 |
| email | - | email | string | 123456 |
| password | + | пароль | string | 123456 |

**Параметры ответа:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| foundName | - | сообщает, найдено ли имя пользователя | boolean | true |
| foundEmail | - | сообщает, найдена ли почта пользователя | boolean | true |
| token | - | токен аутентификации | string | dvdhvfg |

Отсутствие токена обозначает, что пароль не верен.

1. /account/ register

**Параметры запроса:**

POST host/account/register HTTP/1.1

Host: host

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| name | + | имя пользователя | string | 123456 |
| email | + | email | string | 123456 |
| password | + | пароль | string | 123456 |

**Параметры ответа:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| alreadyExists | + | сообщает, найдена ли почта пользователя | boolean | true |
| token | - | токен аутентификации | string | dvdhvfg |

1. /account/change\_login

**Параметры запроса:**

PUT host/ account/change\_login HTTP/1.1

Host: host

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| login | + | новый login | string | name |
| token | + | токен аутентификации | string | dvdhvfg |

**Параметры ответа:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| success | + | сообщает, проведено ли обновление успешно | boolean | true |

1. /account/change\_email

**Параметры запроса:**

PUT host/ account/change\_email HTTP/1.1

Host: host

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| email | + | новый email | string | sth@mail.com |
| token | + | токен аутентификации | string | dvdhvfg |

**Параметры ответа:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| success | + | сообщает, проведено ли обновление успешно | boolean | true |

1. /account/check\_password

**Параметры запроса:**

GET host/ account/ check\_password HTTP/1.1

Host: host

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| password | + | пароль | string | sth@mail.com |
| token | + | токен аутентификации | string | dvdhvfg |

**Параметры ответа:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| success | + | сообщает, проведено ли обновление успешно | boolean | true |
| passToken | - | токен смены пароля | string | kpdfjs |

1. /account/change\_password

**Параметры запроса:**

PUT host/ account/change\_password HTTP/1.1

Host: host

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| password | + | новый пароль | string | sth@mail.com |
| passToken | + | токен смены пароля | string | dvdhvfg |

**Параметры ответа:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| success | + | сообщает, проведено ли обновление успешно | boolean | true |
| token | + | токен аутентификации | string | kpdfjs |

1. /account/profile

**Параметры запроса:**

GET host/account/profile HTTP/1.1

Host: host

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| token | + | токен аутентификации | string | sdgsgd |

**Параметры ответа:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| name | + | имя пользователя | string | Name1 |
| email | + | email | string | sth@mail.com |

1. /catego­ries?page=cur\_page&size=cur\_size

**Параметры запроса:**

Запрос будет оформлен как query запрос с атрибутами:

* page. Текущая страница для пагинации;
* size. Количество элементов на странице для пагинации.

GET host/categories?page=cur\_page&size=cur\_size HTTP/1.1

Host: host

**Параметры ответа:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| categories | + | список категорий | Array, Section[] | - |

1. /recipes?page=cur\_page&size=cur\_size&sort=name,direction&

filter\_info=info

**Параметры запроса:**

Запрос будет оформлен как query запрос с атрибутами:

* page. Текущая страница для пагинации;
* size. Количество элементов на странице для пагинации;
* sort. Название и направление сортировки;
* filter\_info. Текст для поиска.

GET host/recipes?page=cur\_page&size=cur\_size&sort=name,direction&

filter\_info=infoHTTP/1.1

Host: host

**Параметры ответа:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| categories | + | список категорий | Array, Section[] | - |

1. /recipes/favourite?page=cur\_page&size=cur\_size&sort=name,direction&filter\_info=info

**Параметры запроса:**

Запрос будет оформлен как query запрос с атрибутами:

* page. Текущая страница для пагинации;
* size. Количество элементов на странице для пагинации;
* sort. Название и направление сортировки;
* filter\_info. Текст для поиска.

GET host/recipes/favourite?page=cur\_page&size=cur\_size

&sort=name,direcion&filter\_info=info HTTP/1.1

Host: host

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| token | + | токен аутентификации | string | fh74m |

**Параметры ответа:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| categories | + | список категорий | Array, Section[] | - |

1. /recipes?recipe\_id=id

**Параметры запроса:**

Запрос будет оформлен как query запрос с атрибутами:

* recipe\_id. ID рецепта.

GET host/ recipes?recipe\_id=id HTTP/1.1

Host: host

**Параметры ответа:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| recipe | + | информация о рецепте | Recipe | - |

1. /recipes/steps/add\_comment?recipe\_id=id&step\_id=id

**Параметры запроса:**

Запрос будет оформлен как query запрос с атрибутами:

* recipe\_id. ID рецепта;
* step\_id. Id шага.

POST host/recipes/steps/add\_comment?recipe\_id=id&step\_id=id HTTP/1.1

Host: host

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| token | + | токен аутентификации | string | fh74m |
| comment | + | текст комментария | string | commenting |

**Параметры ответа:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| success | + | сообщает, проведено ли обновление успешно | boolean | true |

1. /recipes/steps/update\_comment?recipe\_id=id&step\_id=id&

comment\_id=id

**Параметры запроса:**

Запрос будет оформлен как query-запрос с атрибутами:

* recipe\_id. ID рецепта;
* step\_id. ID шага;
* comment\_id. ID изменяемого комментария.

PUT host/recipes/steps/update\_comment?recipe\_id=id&step\_id=id&

comment\_id=id HTTP/1.1

Host: host

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| token | + | токен аутентификации | string | fh74m |
| comment | + | текст комментария | string | commenting |

**Параметры ответа:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| success | + | сообщает, проведено ли обновление успешно | boolean | true |

1. /recipes/steps/delete\_comment?recipe\_id=id&step\_id=id

&comment\_id=id

**Параметры запроса:**

Запрос будет оформлен как query-запрос с атрибутами:

* recipe\_id. ID рецепта;
* step\_id. ID шага;
* comment\_id. ID удаляемого комментария.

DELETE host/recipes/steps/delete\_comment?reci-pe\_id=id&step\_id=id

&comment\_id=id HTTP 1.1

Host: host

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| token | + | токен аутентификации | string | fh74m |

**Параметры ответа:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| success | + | сообщает, проведено ли обновление успешно | boolean | true |

1. /recipes/add

**Параметры запроса:**

POST host/recipes/steps/delete\_comment?reci-pe\_id=id&step\_id=id

&comment\_id=id HTTP 1.1

Host: host

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| token | + | токен аутентификации | string | fh74m |
| recipe | + | информация о рецепте | Recipe | - |

**Параметры ответа:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| success | + | сообщает, проведено ли обновление успешно | boolean | true |

1. / recipes/delete?recipe\_id=id

**Параметры запроса:**

Запрос будет оформлен как query-запрос с атрибутами:

* recipe\_id. ID рецепта.

DELETE host/recipes/delete?recipe\_id=id HTTP 1.1

Host: host

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| token | + | токен аутентификации | string | fh74m |
| recipe | + | информация о рецепте | Recipe | - |

**Параметры ответа:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| success | + | сообщает, проведено ли обновление успешно | boolean | true |

1. / recipes/update?recipe\_id=id

**Параметры запроса:**

Запрос будет оформлен как query-запрос с атрибутами:

* recipe\_id. ID рецепта.

PUT host/recipes/update?recipe\_id=id HTTP 1.1

Host: host

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| token | + | токен аутентификации | string | fh74m |
| recipe | + | информация о рецепте | Recipe | - |

**Параметры ответа:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| success | + | сообщает, проведено ли обновление успешно | boolean | true |

1. /recipees/add\_ingredient? recipe\_id=id

**Параметры запроса:**

Запрос будет оформлен как query-запрос с атрибутами:

* recipe\_id. ID рецепта.

POST host/recipes/update?recipe\_id=id HTTP 1.1

Host: host

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| token | + | токен аутентификации | string | fh74m |
| ingredient | + | информация об ингредиенте | Ingredient | - |

**Параметры ответа:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| success | + | сообщает, проведено ли обновление успешно | boolean | true |

1. /recies/delete\_ingredient?ingredient\_id=id

**Параметры запроса:**

Запрос будет оформлен как query-запрос с атрибутами:

* ingredient\_id. ID ингредиента.

DELETE host/recipes/delete\_ingredient?ingredient\_id=id HTTP 1.1

Host: host

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| token | + | токен аутентификации | string | fh74m |

**Параметры ответа:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| success | + | сообщает, проведено ли обновление успешно | boolean | true |

1. /recipes/update\_ingredient? ingredient=id

**Параметры запроса:**

Запрос будет оформлен как query-запрос с атрибутами:

* ingredient\_id. ID ингредиента.

PUT host/recipes/update\_ingredient? ingredient=id HTTP 1.1

Host: host

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| token | + | токен аутентификации | string | fh74m |
| ingredient | + | информация об ингредиенте | Ingredient | - |

**Параметры ответа:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| success | + | сообщает, проведено ли обновление успешно | boolean | true |

1. /recipes/dose\_calculation?recipe\_id=id&doses=num

**Параметры запроса:**

Запрос будет оформлен как query-запрос с атрибутами:

* recipe\_id. ID рецепта;
* doses. Новое количество порций.

PUT host/recipes/dose\_calculation?recipe\_id=id&doses=num HTTP 1.1

Host: host

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| token | + | токен аутентификации | string | fh74m |

**Параметры ответа:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| success | + | сообщает, проведено ли обновление успешно | boolean | true |

1. /recipe/get\_rating?recipe\_id=id

**Параметры запроса:**

Запрос будет оформлен как query-запрос с атрибутами:

* recipe\_id. ID рецепта.

GET host/recipe/get\_ratingHTTP 1.1

Host: host

**Параметры ответа:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| rating | + | оценка | float | 4.5 |
| numOfVoted | + | количество голосовавших | integer | 5 |

1. /recipe/set\_rating?recipe\_id=id&rating=rating

**Параметры запроса:**

Запрос будет оформлен как query-запрос с атрибутами:

* recipe\_id. ID рецепта;
* rating. Оценка.

POST host/recipe/set\_rating?recipe\_id=id&rating=rating HTTP 1.1

Host: host

**Параметры ответа:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| success | + | сообщает, проведено ли обновление успешно | boolean | true |

1. Используемые классы
2. Ingredient

Параметры:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| ingredientID | + | ID ингредиента | integer | 1 |
| recipeID | + | ID рецепта, к которому он относится | integer | 1 |
| name | + | наименование | string | ингредиент1 |
| amount | + | количество | float | 1.25 |
| unit | + | единица измерения | string | грамм |

**JSON:**

{

“ingredientID”: 1,

“recipeID”:1,

“name”: “ингредиент1”,

“amount”:1.25,

“unit”: “грамм”

}

1. Step

Параметры:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| stepID | + | ID шага | integer | 1 |
| recipeID | + | ID рецепта, к которому он относится | integer | 1 |
| num | + | порядковый номер | integer | 1 |
| text | + | текст | string | текст |
| image | + | картинка | String | data:image  /jpeg;base64,/9j/4A |
| comments | + | комментарии к шагу | Array,  Comment[] | - |

**JSON**

{

“stepId”: 1,

“recipeID”: 1,

“num”: 1,

“text”: “text”,

“image”: “data:image/jpeg;base64,/9j/4A”,

“comments”: [

{

“commentaryID”: 1,

“order\_num”:1,

“text”: “text”

},

{

“commentaryID”: 2,

“order\_num”:2,

“text”: “text2”

}

]

}

1. Recipe

Параметры:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| recipe\_ID | + | ID рецепта | integer | 1 |
| sectionID | + | ID раздела, к которому он относится | integer | 1 |
| name | + | наименование | string | ингредиент1 |
| caloriesOnHundG | + | калории на 100 г | float | 1.25 |
| timeToCook | + | время приготовления в минутах | integer | 1 |
| doseNum | + | количество порций | integer | 1 |
| shortDescription | - | краткое описание | string | описание |
| timeWhenAdded | + | время добавления | dateTime | 1900-01-01 00:00:00 |
| steps | + | шаги приготовления | Array, Step[] | - |
| ingredients | + | ингредиенты | Array, Ingredient[] | - |

**JSON:**

{

“recipeID”: 1,

“sectionID”: 1,

“name”: “ингредиент1”,

“caloriesOnHundG”: 1.25,

“timeToCook”: 1,

“doseNum”: 1,

“shortDescription”: “описание”,

“timeWhenAdded”: “1900-01-01 00:00:00”,

“steps”: [

{

}

],

“ingredients”: [

{

}

]

}

1. Section

Параметры:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| sectionID | + | ID шага | integer | 1 |
| name | + | текст | string | текст |
| image | + | картинка | String | data:image  /jpeg;base64,/9j/4A |

**JSON:**

{

“ingredientID”: 1,

“recipeID”:1,

“name”: “ингредиент1”,

“amount”:1.25,

“unit”: “грамм”

}

1. Commentary

Параметры:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Обязательность | Назначение | Тип | Пример |
| commentaryID | + | ID комментария | integer | 1 |
| order\_num | + | порядковый номер комментария | integer | 1 |
| text | + | текст | string | текст |

**JSON:**

{

“commentaryID”: 1,

“order\_num”:1,

“text”: “text”

}

1. SortStyle (enum):

enum SortStyle {

ALPHABET(“От А до Я”),

TIME(“По времени готовки”),

DATE(“От нового к старому”)

}