

# Тестовое задание (фронтенд-разработчик)

## Задача: Разработать приложение-клиент(фронтенд) для REST сервиса.

В рамках тестового задания необходимо спроектировать и реализовать веб приложение (только фронтенд) реализующее функционал простейшей рекомендательной сети (каталог продуктов).

Обязательные функции:

- просмотр списка продуктов.
- оставление комментариев.
- голосование за продукт.

В приложении должна присутствовать возможность зарегистрироваться и войти в каталог для пользователя. Предусмотреть возможность просмотра продуктов для незарегистрированного пользователя.

После входа должен отобразиться список продуктов.

При выборе продукта отобразить следующую информацию: наименование, изображение, развернутое описание и список отзывов. Список отзывов должен отображать текст отзыва и оценку по пятибалльной шкале. Также должна присутствовать возможность оставить отзыв с оценкой для товара. Возможность оставлять отзыв должна присутствовать только для зарегистрированного пользователя.

При разработке пользовательского интерфейса нет никаких определенных требований к внешнему виду приложения, однако функциональные и удобные в использовании варианты интерфейса будут оценены существенно выше. Фактически будет проверена способность разработчика разрабатывать UI (user interface), опираясь на общепринятые представления.

Результат тестового задания должен продемонстрировать знание JavaScript, HTML5, CSS3, HTTP, Json,.

Использование Angular2+(AngularJS), React.js либо другого JS фреймворка поддерживающего парадигму MVC (например Vue.js) является необходимым.

Результат тестового задания может продемонстрировать знание разработчиком паттернов проектирования, unit tests, стандартов кодирования; понимание принципов разработки веб-приложений. Это будет дополнительным плюсом при оценке задания.

В качестве бекенда для разработки можно использовать REST API описанный ниже. Ровно как и создать свой с помощью инструментов быстрого прототипирования или любых других. (например [www.mockable.io](http://www.mockable.io), [apiary.io](http://apiary.io), [jsonplaceholder.typicode.com](http://jsonplaceholder.typicode.com) и тд.)

## Описание REST API, которое можно использовать в разработке.

URL сервиса: <http://smktesting.herokuapp.com/>

На сервере хранится список продуктов и список отзывов к ним с оценками.

Модель продукта имеет следующий вид:

- Id — идентификатор продукта
- Title — наименование продукта
- Image — изображение продукта
- Text — описание продукта

Каждый товар может иметь отзывы. Модель отзыва определена следующим образом:

- Id — идентификатор отзыва
- Rate — оценка
- Text — комментарий
- Id\_user — идентификатор пользователя
- Id\_entry — идентификатор продукта

Модель пользователя:

- Id — идентификатор пользователя
- Username — имя пользователя.
- Password — пароль. Описание API запросов

Для выполнения запросов к api нужно передавать полученный токен доступа в заголовке http-запроса:

Authorization: Token <token>

POST /api/register/ — зарегистрировать пользователя.

Пример запроса:

```
{
  "username": <username>,
  "password": <password>
}
```

Пример ответа:

```
{
  "success": true,
  "token": <token>
}
```

POST /api/login/ — получение токена для авторизации.

Пример запроса:

```
{
  "username": <username>,
  "password": <password>
}
```

Пример ответа:

```
{
  "success": true,
  "token": <token>
}
```

GET /api/products/ — возвращает список продуктов.

Пример ответа:

```
[
  {
    "id": 1,
    "img": "img1.png",
    "text": "Lorem ipsum dolor sit amet.",
    "title": "product 1"
  },
  ...
]
```

POST /api/reviews/<product\_id>/ - постить отзыв о продукте с идентификатором .

Пример запроса:

```
{
  "rate": 5,
  "text": "Lorem"
}
```

Пример ответа:

```
{
  "review_id": 5
}
```

GET /api/reviews/<product\_id>/ - возвращает список отзывов для продукта с идентификатором .

Пример ответа:

```
[
  {
    "id": 10,
    "product": 1,
    "created_by": {
      "id": 1,
      "username": "user"
    },
    "rate": 0,
    "text": "ipsum"
  },
  ...
]
```

## Требования к оформлению и критерии оценки

Программа должна быть рабочей. Если присланный код не работает, или его не удаётся запустить из-за отсутствия необходимых инструкций - тестовое задание не рассматривается.

**Обязательным условием является использование систем контроля версии. Проверяться будут только те тестовые задания исходный код которых в репозитории (например github или bitbucket).**

Код должен соответствовать общепринятым стилевым и организационным стандартам действующим для выбранных вами языков и технологий.

По возможности код должен сопровождаться разумными комментариями, юнит-тестами, прочими инструкциями.

Оцениваться тестовое задание будет по следующим критериям:

- За полноту выполнения и работоспособность - максимум 6 баллов;
- За качество кода разметки (html, css), соответствие общепринятым стандартам и т.п. - максимум 2 бала;
- За качество js кода и проектирование - максимум 2 бала;
- За внешний вид и креативность - 2 бала.

Для прохождения отбора соискатель должен набрать 7 баллов.