

Тестовое задание (Mobile Developer)

Тестовое задание для Mobile Developer имеет основное задание и 2 дополнительных задания. Это необходимо для более гибкого понимания уровня знаний и опыта кандидата. Дополнительные задания не являются обязательными к выполнению. Задание должно выполняться так: обязательная основная часть и по мере достижения ожидаемого результата 1й дополнительный сценарий и после его реализации 2й дополнительный сценарий.

В рамках тестового задания необходимо спроектировать и реализовать мобильное приложение реализующее функционал простейшего каталога продуктов.

Обязательные функции:

- Авторизация.
- 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 отображаться 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.

В качестве бекенда для разработки можно использовать REST API описанный ниже.

Описание REST API, которое можно использовать в разработке.

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

URL для отображения картинок: <http://smktesting.herokuapp.com/static/>

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"
  },
  ...
]
```

Дополнительное задание

Для кандидатов претендующих на более высокий уровень:

- 1) Добавить возможность создания и редактирования профайла пользователя в случае если был пройден процесс регистрации.

Пользователь может добавить свое имя, фамилию, аватарку. Данные должны храниться локально на девайсе. Профайл пользователь должен заполнить после регистрации и иметь в последующем возможность его отредактировать. Когда пользователь выходит из своего аккаунта эти данные должны очищаться.

- 2) Добавить возможность сохранения товара, по желанию пользователя, на девайс. В случае отсутствия интернета уведомить пользователя об этом и предоставить ему доступ к товарам в "офлайн" режиме. Это условие должно работать только в случае если пользователь прошел процесс регистрации.

Требования к оформлению и критерии оценки

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/UX, опираясь на общепринятые представления.

Приложение должно быть рабочим. 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 5 баллов;
- За качество кода разметки, соответствие общепринятым стандартам и т.п. - максимум 3 бала;
- За качество кода и проектирование - максимум 3 бала;
- За внешний вид и креативность - 1 бала.
- За каждое дополнительное задание по 1 баллу

Для прохождения отбора соискатель должен набрать 8 баллов.