



JavaScript, уровень 3

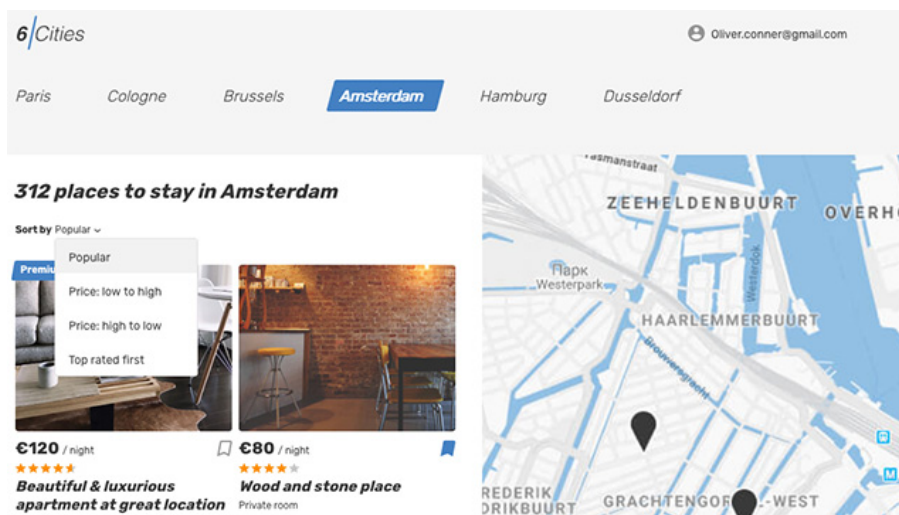
с 3 февраля по 8 апреля 2020

Проекты



[Главная](#) / [Проекты](#) /

Личный проект «Шесть городов»



— [Техническое задание](#)

Репозиторий проекта

[react/1353919-six-cities-3](https://github.com/react-1353919-six-cities-3)

Техническое задание

О проекте

«Шесть городов» — сервис для путешественников, не желающих переплачивать за аренду жилья. Выбирайте один из шести популярных городов для путешествий и получайте актуальный список предложений по аренде. Подробная информация о жилье, показ объекта на карте, а также лаконичный интерфейс сервиса помогут быстро выбрать оптимальное предложение.

Описание функциональности

Страницы приложения

Приложение состоит из нескольких страниц: `Main (/)`, `Sign In (/login)`, `Favorites (/favorites)` (приватная), `Room (/offer/:id)`.

Страница `Favorites` доступна только авторизованным пользователям.

Если пользователь авторизован, то при переходе на страницу `Sign In` выполняется редирект на главную страницу.

Если пользователь не авторизован, то при попытке перехода к приватной странице выполняется редирект на страницу «Sign In» (`/login`).

В шапке каждой страницы отображается ссылка на страницу «Sign In» (если пользователь не авторизован) или email пользователя (если пользователь авторизован).

Клик по email пользователя в шапке выполняет переход на страницу «Favorites» (`/favorites`).

Главная страница

На главной странице отображается список городов, для которых возможно запросить предложения по аренде.

Список городов формируется на основании данных, полученных с сервера. Сервер всегда возвращает информацию только для шести городов.

После загрузки приложения сразу активен первый город и по нему загружены предложения по аренде.

На карте предложения отображаются в виде синих маркеров.

При смене города выполняется обновление списка предложений и карта.

В заголовке списка предложений отображается количество доступных предложений. Пример корректного заголовка: `312 places to stay in Amsterdam`.

Список предложений

Пользователь может менять сортировку списка предложений. Варианты сортировки:

- `Popular`. Вариант по умолчанию. Предложения отображаются в порядке полученном с сервера.
- `Price: low to high`. От дорогих к дешёвым.

- `Price: high to low`. От дешёвых к дорогим.
- `Top rated first`. От высокого рейтинга к низкому.

При смене варианта сортировки список предложений перерисовывается.

Каждая карточка в списке предложений содержит информацию:

- `Изображение`. Фотография жилья.
- `Премиальность`. Метка «Premium».
- `Стоимость за ночь`. Стоимость всегда отображается в евро.
- `Заголовок`. Краткое описание предложения. Например: «Beautiful & luxurious apartment at great location».
- `Тип жилья`. Одно из нескольких значений: `apartment`, `room`, `house`, `hotel`.
- `Рейтинг`. Оценка выводится в виде закрашенных звезд. Максимальное количество звёзд — 5. Оценка пользователя округляется до ближайшего целого. Например, оценка 3.1 округляется до 3-х. Оценка 4.5 округляется до 5.

Клик по заголовку карточки выполняет переход на страницу с подробной информацией о предложении.

Адрес страницы с подробной информации о предложении выглядит следующим образом: `/offer/:id`, где `id` идентификатор предложения. Например, `/offer/1704`.

Если предложения отсутствуют, то в списке отображается надпись «No places to stay available», а вместо карты отображается статичное изображение. См. пример соответствующей страницы макета.

Карта

Все предложения выбранного города отображаются на карте в виде синих маркеров.

При наведении курсора на карточку предложения, маркер, соответствующий объявлению, становится оранжевым.

Страница предложения

На странице предложения (`/offer`) представлена расширенная информация об объекте для аренды:

- `Фотографии`. Выводится до 6-ти изображений.
- `Заголовок`. Краткое описание предложения, например: «Beautiful & luxur studio at great location».

- `Подробное описание`.
- `Премиальность`.
- `Тип жилья`. Одно из предопределённых значений: `apartment` (Apartment), `room` (Private Room), `house` (House), `hotel` (Hotel).
- `Рейтинг`. Оценка предложения отображается в виде закрашенных звезд и среднего бала (например, 4.8). Максимальное количество звёзд — 5.
- `Количество спален`. Например, `3 Bedrooms`.
- `Максимальное количество гостей`. Например, `Max 4 adults`.
- `Стоимость аренды за ночь`. Сумма всегда отображается в евро.
- Список бытовых предметов в квартире (Wifi, Heating, Kitchen, Cable TV и так далее);
- `Информация о хозяине`: аватарка, имя, отметка `super` (звёздочка возле аватарки).

Кнопка «Избранное». Клик по кнопке «Избранное» добавляет карточку в избранное. Если пользователь не авторизован, то выполняется редирект на страницу `Sign In`.

Отзывы пользователей. В заголовке блока отображается общее количество отзывов. Например: `Reviews 1`.

Для авторизованных пользователей отображается форма отправки нового отзыва.

Под списком отзывов отображается карта с предложениями неподалёку. На карте отмечено не больше 3-х предложений неподалёку и маркер текущего предложения. Маркеры предложений выделены синим. Маркер текущего предложения выделен оранжевым. Другая функциональность для карты с предложениями неподалёку не предусмотрена.

Карточки представленных предложений отображаются сразу под картой и содержат тот же набор информации, что и на главной странице.

Отзывы

Каждый отзыв содержит:

- Аватар автора.
- Имя автора.
- Оценку. Оценка выводится в виде закрашенных звезд. Максимальное количество звёзд — 5.
- Дата отзыва в формате: `Месяц Год`. Например: `April 2019`.

— Текст отзыва.

В заголовке блока отображается общее количество отзывов.

На страницу выводится не больше 10 отзывов.

Отзывы должны быть отсортированы от новых к старым (новые сверху).

После успешной отправки отзыва, поля формы очищаются.

Пользователь может оставить любое количество отзывов.

Форма отправки отзыва

Форма отправки отзыва отображается только для авторизованных пользователей.

Пользователь должен выставить рейтинг. Рейтинг выставляется от 1 до 5.

Для выбора рейтинга пользователь отмечает соответствующее количество звёзд.

Текст отзыва должен содержать от 50 до 300 символов.

Пока пользователь не выбрал оценку и не ввёл текст допустимой длины, кнопка отправки отзыва не активна.

При нажатии кнопки «Submit» и отправки данных форма должна блокироваться. Разблокировка формы происходит в случае успешной отправки или при возникновении ошибки.

В случае успешной отправки отзыва форма очищается.

В случае возникновения ошибки следует уведомить пользователя. Способ отображения ошибки остаётся на усмотрение разработчика.

Пользователь может оставить любое количество отзывов.

Страница Sign In

Для входа в сервис пользователь вводит логин (email) и пароль. Поскольку у сервиса отсутствует возможность регистрации, логин и пароль могут быть любыми.

В поле «логин» должен вводиться корректный email.

Страница доступна только неавторизованным пользователям. Авторизованных пользователей редиректит на главную страницу.

Favorites

Страница «Favorites» доступна только авторизованным пользователям. Если пользователь не авторизован, то выполняется редирект на страницу «Sign In».

Переход на страницу «Favorites» осуществляется при клике на email авторизованного пользователя.

На странице «Favorites» отображаются все предложения, которые пользователь добавил в избранное. Предложения группируются по городам.

Если пользователь не добавил ни одного предложения в избранное, то на странице отображается «Nothing yet saved» (смотрите макет).

Взаимодействие с сервером

Все необходимые данные загружаются с сервера.

Сервер доступен по адресу: `https://htmlacademy-react-3.appspot.com/six-cities`.

В случае недоступности сервера отображается информационное сообщение. Дизайн сообщения остаётся на усмотрение разработчика.

Сервер принимает данные в виде JSON объекта.

Запросы должны предоставлять доступ к кукам. В случае если запросы отправляются через `axios`, то должен быть проставлен параметр `withCredentials: true`.

Структуры данных

Hotel

```
{
  "bedrooms": 3,
  "city": {
    "location": {
      "latitude": 52.370216,
      "longitude": 4.895168,
      "zoom": 10
    },
    "name": "Amsterdam"
  },
  "description": "A quiet cozy and picturesque that hides behind a a ri
by the unique lightness of Amsterdam.",
  "goods": ["Heating", "Kitchen", "Cable TV", "Washing machine", "Coffe
machine", "Dishwasher"],
  "host": {
```

```
"avatar_url": "img/1.png",
"id": 3,
"is_pro": true,
"name": "Angelina"
},
{id": 1,
"images": ["img/1.png", "img/2.png"],
"is_favorite": false,
"is_premium": false,
"location": {
  "latitude": 52.35514938496378,
  "longitude": 4.673877537499948,
  "zoom": 8
},
"max_adults": 4,
"preview_image": "img/1.png",
"price": 120,
"rating": 4.8,
"title": "Beautiful & luxurious studio at great location",
"type": "apartment"
}
```

CommentGet

```
{
  "comment": "A quiet cozy and picturesque that hides behind a a river by
the unique lightness of Amsterdam.",
  "date": "2019-05-08T14:13:56.569Z",
  "id": 1,
  "rating": 4,
  "user": {
    "avatar_url": "img/1.png",
    "id": 4,
    "is_pro": false,
    "name": "Max"
  }
}
```

CommentPost

```
{
  "comment": "A quiet cozy and picturesque that hides behind a a river by
the unique lightness of Amsterdam.",
  "rating": 4
}
```

User

```
{  
  "email": "Oliver.conner@gmail.com",  
  "password": "12345678"  
}
```

AuthInfo

```
{  
  "avatar_url": "img/1.png",  
  "email": "Oliver.conner@gmail.com",  
  "id": 1,  
  "is_pro": false,  
  "name": "Oliver.conner"  
}
```

Маршруты

GET /hotels

Получение списка предложений.

Коды ответов:

— 200 OK

Пример:

Request:

— URL: GET /hotels

Response:

— Status: 200 OK

— Body: массив, содержащий структуры типа `Hotel`

GET /hotels/: hotel_id/nearby

Получение списка предложений неподалеку.

Коды ответов:

— 200 OK

Пример:

Request:

— URL: GET /hotels/42/around

Response:

- Status: 200 OK
- Body: массив, содержащий структуры типа `Hotel`

GET /favorite

Получение списка избранных предложений.

Коды ответов:

- 200 OK
- 401 Unauthorized (в случае если не пройдена авторизация)

Пример:

Request:

- URL: GET /favorite

Response:

- Status: 200 OK
- Body: массив, содержащий структуры типа `Hotel`

POST /favorite/: hotel_id/: status

Изменение статуса избранного у предложения. В теле ответа приходят данные по отелю с актуальным состоянием поля `is_favorite`.

Параметры:

- `:hotel_id` — ID отеля, который нужно убрать/добавить в избранное
- `:status` — значения могут быть 1 или 0. 1 добавляет отель в избранное, 0 удаляет

Коды ответов:

- 200 OK
- 401 Unauthorized (в случае если не пройдена авторизация)

Пример:

Request:

- URL: GET /favorite/42/1

Response:

- Status: 200 OK

- Body: структура `Hotel` с актуальным состоянием поля `is_favorite`

GET /comments/: hotel_id

Получить список комментариев для конкретного предложения по его `id`.

Коды ответов:

- 200 OK
- 400 Bad request

Пример:

Request:

- URL: GET /comments/42

Response:

- Status: 200 OK
- Body: массив, содержащий структуры типа `CommentGet`

POST /comments/: hotel_id

Отправить новый комментарий к конкретному предложения по его `id`.

Коды ответов:

- 200 OK
- 400 Bad request
- 401 Unauthorized (в случае если не пройдена авторизация)

Пример:

Request:

- URL: POST /comments/42
- Body: структура `CommentPost`

Response:

- Status: 200 OK
- Body: массив, содержащий структуры типа `CommentGet` для текущего предложения

POST /login

Авторизация пользователя на сервере. В случае успешного запроса в куку записывается токен, по которому в дальнейшем происходит авторизация. E

авторизация на сервере не проходит, то возвращается 401 ошибка при запросах на приватные части сайта.

Коды ответов:

- 200 OK
- 400 Bad request

Пример:

Request:

- URL: POST /login
- Body: структура `User`

Response:

- Status: 200 OK
- Body: структура `AuthInfo`

GET /login

Проверка статуса авторизации пользователя.

Коды ответов:

- 200 OK
- 401 Unauthorized (в случае если не пройдена авторизация)

Пример:

Request:

- URL: GET /login

Response:

- Status: 200 OK
- Body: структура `AuthInfo`

Коды ответов:

- 200 OK
- 401 Unauthorized (в случае если не пройдена авторизация)