



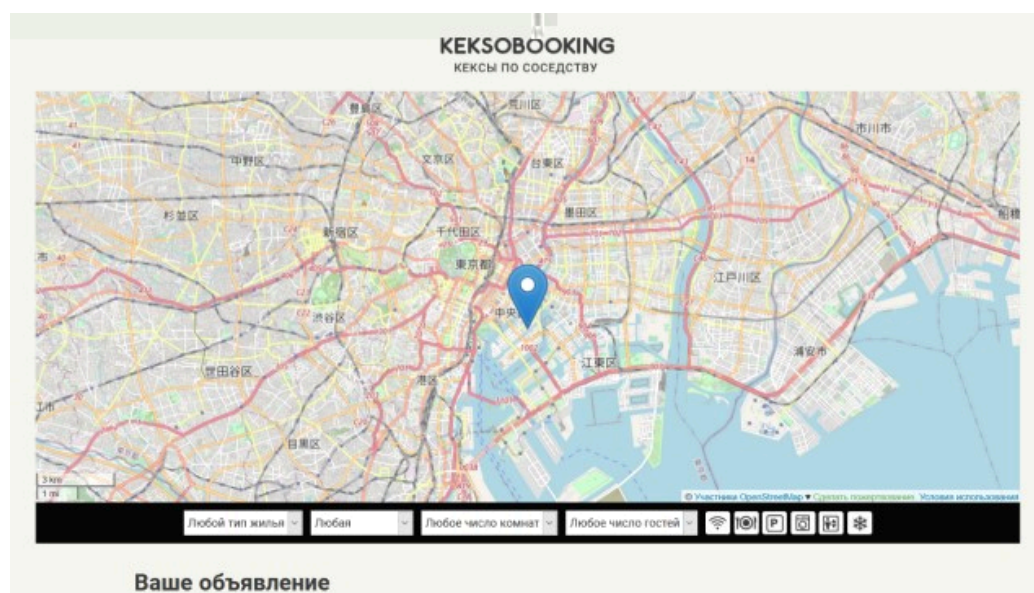
# Профессия «Бэкенд-разработчик #1»

## JavaScript. Профессиональная разработка веб-интерфейсов

Меню курса

[Главная](#) / [Проекты](#) /

## Проект «Кексобукинг»



— [Скачать архив с начальным состоянием](#)

— [Техническое задание](#)

### Техническое задание

#### О проекте

Кексобукинг — сервис размещения объявлений о сдаче в аренду недвижимости в центре Токио. Пользователям предоставляется возможность размещать объявления о своей недвижимости или просматривать уже размещённые объявления.

# Описание функциональности

## 1. Состояния страницы

**1.1. Неактивное состояние.** При открытии страница находится в неактивном состоянии:

- На месте карты отображается серый прямоугольник.
- Форма заполнения информации об объявлении `.ad-form` содержит класс `ad-form--disabled`;
- Все интерактивные элементы формы `.ad-form` должны быть заблокированы с помощью атрибута `disabled`, добавленного на них или на их родительские блоки `fieldset`;
- Форма с фильтрами `.map__filters` заблокирована так же, как и форма `.ad-form` — на форму добавлен специальный класс, а на её интерактивные элементы атрибуты `disabled`.

**1.2. Активное состояние.** Загрузка и успешная инициализация карты (карта реализуется сторонней библиотекой Leaflet) переводит страницу в активное состояние. В активном состоянии страница позволяет:

- Вносить изменения в форму и отправлять её на сервер;
- После загрузки данных с сервера просматривать похожие объявления на карте, фильтровать их и уточнять подробную информацию о них, показывая для каждого из объявлений карточку.

## 2. Заполнение информации

**2.1. Заполнение информации и отправка данных:**

- фотография пользователя;
- заголовок;
- адрес (координаты);
- тип жилья;
- цена за ночь;
- количество комнат;

- количество спальных мест;
- время заезда и выезда из квартиры;
- дополнительные параметры:
  - парковка;
  - WiFi;
  - кондиционер;
  - кухня;
  - стиральная машина;
  - лифт.
- свободное текстовое описание;
- фотография жилья.

2.2. Заполнение всей информации производится на одной странице без промежуточных переходов. Порядок заполнения информации не важен.

2.3. После заполнения всех данных, при нажатии на кнопку «Опубликовать», все данные из формы, включая изображения, с помощью AJAX-запроса отправляются на сервер <https://23.javascript.htmlacademy.pro/keksobooking> методом `POST` с типом `multipart/form-data`.

2.4. Страница реагирует на неправильно введенные значения в форму. Если данные, введенные в форму, не соответствуют ограничениям, указанным в разделе, описывающем поля ввода, форму невозможно отправить на сервер. При попытке отправить форму с неправильными данными, отправки не происходит, а неверно заполненные поля подсвечиваются красной рамкой. Способ добавления рамки и её стиль произвольные.

2.5. При успешной отправке формы или её очистке (нажатие на кнопку `.ad-form__reset`) страница, не перезагружаясь, переходит в состояние, когда:

- все заполненные поля возвращаются в изначальное состояние;
- фильтрация (состояние фильтров и отфильтрованные метки) сбрасывается;

- метка адреса возвращается в исходное положение;
- значение поля адреса корректируется соответственно исходному положению метки.

2.6. Если отправка данных прошла успешно, показывается соответствующее сообщение. Разметку сообщения, которая находится в блоке `#success` внутри шаблона `template`, нужно разместить перед закрывающим тегом `</body>`. Сообщение должно исчезать по нажатию на клавишу `Esc` и по клику на произвольную область экрана.

2.7. Если при отправке данных произошла ошибка запроса, показывается соответствующее сообщение. Разметку сообщения, которая находится в блоке `#error` в шаблоне `template`, нужно разместить перед закрывающим тегом `</body>`. Сообщение должно исчезать после нажатия на кнопку `.error__button`, по нажатию на клавишу `Esc` и по клику на произвольную область экрана. **В таком случае вся введённая пользователем информация сохраняется, чтобы у него была возможность отправить форму повторно.**

### 3. Ограничения, накладываемые на поля ввода

3.1. Заголовок объявления:

- Обязательное текстовое поле;
- Минимальная длина — 30 символов;
- Максимальная длина — 100 символов.

3.2. Цена за ночь:

- Обязательное поле;
- Числовое поле;
- Максимальное значение — 1 000 000.

3.3. Поле «Тип жилья» влияет на минимальное значение поля «Цена за ночь»:

- «Бунгало» — минимальная цена за ночь 0;
- «Квартира» — минимальная цена за ночь 1 000;
- «Отель» — минимальная цена за ночь 3 000;

- «Дом» — минимальная цена 5 000;
- «Дворец» — минимальная цена 10 000.

«**Обратите внимание:** вместе с минимальным значением цены нужно изменять и плейсхолдер.

«**Обратите внимание:** ограничение минимальной цены заключается именно в изменении минимального значения, которое можно ввести в поле с ценой, изменять само значение поля не нужно, это приведёт к плохому UX (опыту взаимодействия). Даже если текущее значение не попадает под новые ограничения, не стоит без ведома пользователя изменять значение поля.

3.4. Адрес: ручное редактирование поля запрещено. Значение автоматически выставляется при перемещении специальной метки по карте. Подробности заполнения поля адреса описаны вместе с поведением метки.

3.5. Поля «Время заезда» и «Время выезда» синхронизированы: при изменении значения одного поля во втором выделяется соответствующее ему значение. Например, если время заезда указано «после 14», то время выезда будет равно «до 14» и наоборот.

3.6. Поле «Количество комнат» синхронизировано с полем «Количество мест» таким образом, что при выборе количества комнат вводятся ограничения на допустимые варианты выбора количества гостей:

- 1 комната — «для 1 гостя»;
- 2 комнаты — «для 2 гостей» или «для 1 гостя»;
- 3 комнаты — «для 3 гостей», «для 2 гостей» или «для 1 гостя»;
- 100 комнат — «не для гостей».

«**Обратите внимание:** допускаются разные способы ограничения допустимых значений поля «Количество мест»: удаление из разметки соответствующих элементов `option`, добавление

элементам `option` состояния `disabled` или другие способы ограничения, например, с помощью метода `setCustomValidity`.

3.7. Значением полей «Ваша фотография» и «Фотография жилья» может быть только изображение.

## 4. Выбор адреса на карте:

4.1. Приблизительный адрес квартиры указывается перемещением специальной метки по карте Токио. Содержимое поля адреса должно всегда соответствовать координатам метки.

4.2. Поле адреса должно быть заполнено всегда, в том числе сразу после активации страницы. По умолчанию используются координаты центра Токио.

4.3. Формат значения поля адреса: `lat, lng`, где `lat` и `lng` — это координаты метки (от англ. latitude и longitude, широта и долгота). Данные предоставляются API карт. Дробные координаты округляются до пяти символов после запятой.

## 5. Сравнение с похожими объявлениями

5.1. Полный список похожих объявлений загружается после перехода страницы в активное состояние с сервера <https://23.javascript.htmlacademy.pro/keksobooking/data>. **Данные с сервера могут быть получены не в полном объёме.**

5.2. Если при загрузке данных с сервера произошла ошибка запроса, нужно показать соответствующее сообщение. Дизайн блока с сообщением нужно придумать самостоятельно.

5.3. Каждое из объявлений показывается на карте в виде метки. Размеры метки похожего объявления — 40 на 40, размеры специальной метки выбора адреса — 52 на 52.

5.4. При нажатии на метку похожего объявления, показывается балун (предоставляется API карт) с подробной информацией об объявлении (далее — карточка). Разметка карточки должна

создаваться на основе шаблонного элемента `.popup`, расположенного в элементе `template` с идентификатором `#card`. Данные в карточку вставляются по аналогии с данными, вставленными в шаблонную карточку в качестве примера. **Если данных для заполнения не хватает, соответствующий блок в карточке скрывается. Например, если в объявлении не указано никаких удобств, нужно скрыть блок `.popup__fc_types`. При отсутствии полей не должно возникать ошибок.**

5.5. Нажатие на специальную метку выбора приводит к показу балуна.

5.6. В каждый момент времени может быть показано только одно объявление. Это означает, что при нажатии на метку другого похожего объявления скрывается текущий балун, если он показан и появляется соответствующий другому объявлению.

5.7. Открытую карточку с подробной информацией можно закрыть нажатием на крестик в углу балуна.

5.8. Объекты, расположенные неподалёку, могут быть скрыты. Фильтрация производится при изменении значений полей формы `.map__filters` по тем же параметрам, которые указываются для объявления:

- тип жилья;
- цена за ночь;
- число комнат;
- число гостей;
- дополнительные удобства.

5.9. Как до изменения фильтров, так и при изменении на карте должны показываться все подходящие объявления, но не более 10 меток (не считая специальной), от выбранного фильтра. Показанные метки должны соответствовать всем выбранным фильтрам.

5.10. Форма, с помощью которой производится фильтрация объявлений на момент открытия страницы, заблокирована и становится доступной только после окончания загрузки карты.

похожих объявлений, которые в свою очередь начинают загружаться только после загрузки и успешной инициализации карты.

5.11. Отрисовка соответствующих выбранным фильтрам меток должна происходить не чаще, чем раз в полсекунды (устранение дребезга).

5.12. При изменении фильтров открытый балун (при наличии) должен быть скрыт.

## 6. Необязательная функциональность

6.1. В форме подачи объявления показывается превью аватарки пользователя и фотографии помещения при изменении значений соответствующих полей.

### Поиск по материалам

Git

[Все материалы](#)

