

Домашние задание #6

#21 Учебный проект: pimp my mage

В этом задании мы сделаем так, чтобы в окне настройки персонажа под волшебником загружались похожие персонажи других игроков с сервера.

Задача

1. Создайте модуль `backend.js`, который экспортирует в глобальную область видимости функции `backend.load` и `backend.save`, которые будут работать с сервером данных:

- получать с сервера данные с помощью объекта `XMLHttpRequest` или с помощью `JSONP`, обрабатывать полученные запросы и передавать полученную информацию в функцию обратного вызова;
- отправлять данные игрока на сервер, обрабатывать ошибки и скрывать форму редактирования персонажа, если ошибок не произошло;

Функция `backend.load` должна принимать на вход следующие параметры:

- `onLoad` — функция обратного вызова, которая срабатывает при успешном выполнении запроса. При вызове функции `onLoad` в её единственный параметр передается набор полученных данных;
- `onError` — функция обратного вызова, которая срабатывает при неуспешном выполнении запроса. При вызове функции `onError` в её единственный параметр передается сообщение об ошибке;

Функция `backend.save` должна принимать на вход следующие параметры:

- `data` — объект `FormData`, который содержит данные формы, которые будут отправлены на сервер
- `onLoad` — функция обратного вызова, которая срабатывает при успешном выполнении запроса
- `onError` — функция обратного вызова, которая срабатывает при неуспешном выполнении запроса. При вызове функции `onError` в её единственный параметр передается сообщение об ошибке или объект с описанием ошибки полученный с сервера

2. Подключите модуль `backend.js` в `index.html`
3. Перепишите код проекта таким образом, чтобы для блока похожих волшебников в качестве данных использовались не случайно сгенерированные объекты, которые вы делали в задании «#12 Нас опда», а те данные, которые вы загрузите с сервера <https://js.dump.academy/code-and-magick/data>. Для отрисовки в блоке похожих волшебников используйте 4 произвольных записи из полученных данных.
4. Доработайте обработчик отправки формы, который вы делали в задании «#14 Учебный проект: одеть Надежду», так чтобы он отменял действие формы по умолчанию и отправлял данные формы на сервер <https://js.dump.academy/code-and-magick>. А при успешной загрузке данных на сервер закрывал окно редактирования персонажа.
5. Добавьте обработку возможных ошибок при загрузке: создайте DOM-элемент, который будет показывать сообщения об ошибках, произошедших по ходу загрузки данных. Дизайн DOM-элемента предлагается вам придумать самостоятельно.

Работа с JSONP

Сервер, на котором находится информация, поддерживает JSONP запросы, поэтому, при желании, вы можете попрактиковаться в их выполнении и обработке данных. Для этого, нужно отправить запрос по тому же адресу, что указан в задании, добавив к нему GET-параметр `callback`. Так, запрос

на <https://js.dump.academy/code-and-magick/data> отдаст данные для Кодымагии как обычный JSON, а запрос на <https://js.dump.academy/code-and-magick/data?callback=callbackName> отдаст JS-код, который вызывает функцию `callbackName`. Вы можете изменять имя коллбэка, меняя значение GET-параметра. Проверить как это работает можно прямо в браузере, скопировав адрес в адресную строку.

Для того чтобы вы могли закрепить работу с JSONP на практике, мы сделали так, чтобы запросы к любому адресу, в том числе и к тем, которые используются в личных проектах, умели отдавать JSONP. Но помните, что в личных проектах запрос нужно делать через `XMLHttpRequest`, а возможность использовать JSONP добавлена только для того, чтобы вы могли самостоятельно попрактиковаться.

#22 Личный проект: надо подкачаться

В этом задании мы создадим модуль для работы с сервером и скачаем с его помощью данные, которые мы будем использовать вместо тех, которые сгенерировали случайно несколькими заданиями ранее, а также отправим данные формы при помощи AJAX так, чтобы они не перезагружали страницу.

Задача

1. Создайте модуль `backend.js`, который экспортирует в глобальную область видимости функции для взаимодействия с удаленным сервером через XHR.
 - получать с сервера данные с помощью объекта `XMLHttpRequest`, обрабатывать полученные запросы и передавать полученную информацию в функцию обратного вызова;
 - отправлять данные на сервер.

Функция получения данных с сервера должна принимать на вход следующие параметры:

- `onLoad` — функция обратного вызова, которая срабатывает при успешном выполнении запроса. При вызове функции `onLoad` в её единственный параметр передаётся набор полученных данных;
- `onError` — функция обратного вызова, которая срабатывает при неуспешном выполнении запроса. При вызове функции `onError` в её единственный параметр передаётся сообщение об ошибке.

Функция для отправки данных на сервер должна принимать на вход следующие параметры:

- `data` — объект `FormData`, который содержит данные формы, которые будут отправлены на сервер;
- `onLoad` — функция обратного вызова, которая срабатывает при успешном выполнении запроса;
- `onError` — функция обратного вызова, которая срабатывает при неуспешном выполнении запроса. При вызове функции `onError` в её единственный параметр передается сообщение об ошибке или объект с описанием ошибки полученный с сервера.

2. Подключите модуль `backend.js` в `index.html`.

Кекстаграм

3. Доработайте модуль для отрисовки фотографий так, чтобы в качестве данных использовались не случайно сгенерированные объекты, которые вы делали в задании «[#13 Пока все дома](#)» а те данные, [которые вы загрузите](#) с удалённого сервера через XHR: <https://js.dump.academy/kekstagram/data>.
 4. Доработайте обработчик отправки формы, который вы делали в задании «[#16 Личный проект: доверяй, но проверяй](#)», так чтобы он отменял действие формы по умолчанию и [отправлял данные](#) формы посредством XHR на сервер <https://js.dump.academy/kekstagram>. При успешной загрузке данных на сервер закрывал окно редактирования фотографии и сбрасывал значения формы на те, что были поставлены по умолчанию.
-

Кексобукинг

3. Доработайте модуль для отрисовки меток на карте так, чтобы в качестве данных использовались не случайно сгенерированные объекты, которые вы делали в задании «[#13 Пока все дома](#)» а те данные, которые [вы загрузите](#) с помощью модуля `backend.js` с сервера: <https://js.dump.academy/keksobooking/data>.
 4. Доработайте обработчик отправки формы, который вы делали в задании «[#16 Личный проект: доверяй, но проверяй](#)», так чтобы он отменял действие формы по умолчанию и [отправлял данные формы на сервер посредством XHR](#) <https://js.dump.academy/keksobooking>. А при успешной загрузке данных на сервер сбрасывал значения формы на те, что были по умолчанию.
-
5. Добавьте обработку возможных ошибок при загрузке: создайте DOM-элемент, который будет показывать сообщения об ошибках, произошедших по ходу загрузки данных. Дизайн DOM-элемента придумайте самостоятельно.