

## Лабораторная 6

Цель лабораторной работы – познакомиться с сущностью Promise и научиться создавать http-запросы и корректно обрабатывать ответы.

В качестве практикума студентам предлагается добавить «живых» данных на уже имеющийся прототип. В качестве поставщика данных использовать сервис с Mock данными, например: [JSONPlaceholder](#)

Перед началом работы необходимо выбрать ту часть проекта, в которую в дальнейшем будут загружаться данные. Это может быть профиль пользователя, прошедшего авторизацию, комментарии под той или иной фото в галерее, список пользователей в одной из уже созданных таблиц и т.п.

В качестве данных, которые предоставляет сервис из примера, предлагается 6 видов ресурсов на выбор:

- Сообщения для гостевой книги/форума  
(Поля: body (само сообщение), title (заголовок))
- Комментарии под фото либо любым другим материалом  
(Поля: name (автор комментария), email (email автора комментария), body (комментарий))
- Альбомы и фотографии  
(Поля: title (название), thumbnailUrl (ссылка на маленькое изображение), url (ссылка на изображение))
- Список дел  
(Поля: title (описание), completed (boolean флаг))
- Пользователи  
(Поля: username, name, email, address, phone, website, company)

- 1) Добавить отдельный скрипт и подключить на страницу где подразумевается сетевое взаимодействие.
- 2) Добавить gif-анимацию либо свою произвольную css-анимацию для элемента типа preloader под элементом где планируется отобразить запрашиваемый контент
- 3) Дождаться события загрузки страницы и инициализировать обращение к поставщику данных используя Fetch API ([https://developer.mozilla.org/ru/docs/Web/API/Fetch\\_API](https://developer.mozilla.org/ru/docs/Web/API/Fetch_API))
- 4) После получения ответа, скрыть preloader, десериализовать данные в JSON объект и отрендерить полученные данные.
- 5) Добавить псевдо-случайную фильтрацию к запросам (например при первом обращении получать комментарии с id 100 и выше, а при втором с id 200 и меньше).
- 6) Добавить обработку ошибок (например – сеть перестала быть доступна и запрос не был выполнен), в случае исключительной ситуации добавить под элементом заплатку, например, «⚠ Что-то пошло не так».