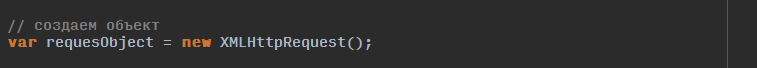
**Практическое занятие №19 AJAX**

**Задание №1**

Проверьте запущен ли у Вас Open Server. Папка с материалами урока (jquery19.loc) должна быть размещена в директории /domains сервера. Заходим на <http://jquery19.loc> и начинаем урок.

**Занятие №2**

Ajax нам нужен для того что бы придать web-странице интерактивности. В чистом JS есть объект «XMLHttpRequest», который по сути моделирует поведение браузера и может посылать запросы на сервер, а также получать ответ от сервера. В этом задании мы разберём простой пример получения данных с сервера. Набирайте код ниже:

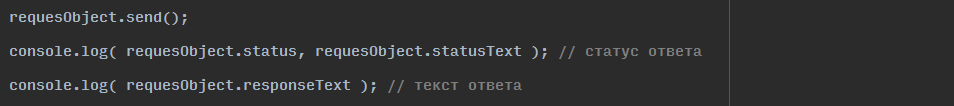


Далее нам необходимо настроить данные для соединения с сервером. Мы используем следующие параметры:

* GET – метод соединения. Можно еще задать POST и другие экзотические методы соединения
* «/task-2/» - адрес к которому мы обращаемся, в данном случае это и есть наша страница задания.
* FALSE (может быть TRUE) – устанавливаем запрос синхронным (асинхронным)



Несмотря на то, что метод .open имеет такое название, он только настраивает будущий запрос, выполняет запрос совсем другой метод – send.



Какой ответ мы сейчас получили? Что содержится в тексте ответа и почему?

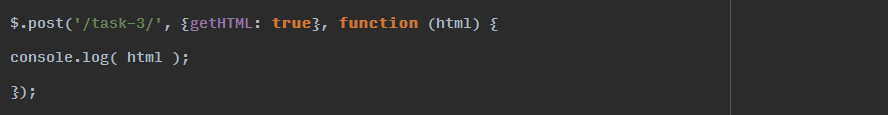
Самостоятельно поменяйте настройку запроса, что бы он стал асинхронным. Посмотрите, что произойдёт и почему?

ЭТО ВАЖНО! В браузере Chrome в панели dev tool есть вкладка «Network» и в фильтре есть пункт «XHR» именно там можно посмотреть все AJAX запросы, которые посылает страница. Найдите наш запрос в панели dev tools.

**Задание №3**

Библиотека JQuery имеет свои инструменты для работы с AJAX. В этом задании мы рассмотрим их.

Отправить запрос AJAX можно с помощью метода $.post или $.get. Рассмотрим и протестируем код ниже:



Рассмотрим этот код более подробно:

* '/task-3/' – это адрес к которому мы обращаемся
* {getHTML: true} – это объект с параметрам и их значениями. Фактически это аналог атрибутов элемента формы {name: value}
* Далее идет функция-обработчик ответа, первым аргумент «html» это строка ответа от сервера.

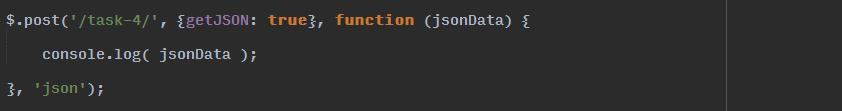
Протестируйте данный код и убедитесь, что Вы получили строку ответа от сервера.

Строка ответа от сервера представляет собой кусок вёрстки. Самостоятельно напишите скрипт, который будет вставлять эту верстку в произвольном месте страницы.

**Задание №4**

В AJAX мы получает ответ в виде одной строки. Это очень неудобно, когда нам нужно передавать множество параметров между сервером и клиентом. Поэтому при использовании AJAX очень популярен формат кодирования данных JSON. JSON позволяет закодировать данные со сложной структурой (массивы и объекты) в строку. Более подробную информацию о формате JSON Вы можете почитать по ссылке <http://www.json.org/json-ru.html>.

Наберите и протестируйте следующий код:



Убедитесь, что Вы получили данные с сервера.

**Задание №5**

В этом задании Вам необходимо самостоятельно написать скрипт обработки формы авторизации с помощью AJAX. Для начала Вы должны перехватить событие отправки формы и собрать данные с полей формы (login и pass). Дальше Вам нужно передать эти данные по AJAX на URL данного урока методом POST. Серверный скрипт настроен таким образом, что если login=admin и pass=admin, то он возвращает сообщение об успешной авторизации, выведите данное сообщение на страницу при её срабатывании.

**Задание №6**

В этом задании Вам нужно самостоятельно создать страницу «Список сотрудников». На этой странице должен быть выведен список всех сотрудников, так же можно создавать нового сотрудника, редактировать зарплату каждого отдельного сотрудника и делать пометку, что сотрудник уволен.

Всё это нужно будет сделать через AJAX. Серверная часть уже настроена, начните с запроса списка всех сотрудников. Вёрстку Вы определяете сами.

Все запросы должны идти на URL «/task-6/» через POST.

Все запросы к серверу сведены в таблице:

|  |  |  |
| --- | --- | --- |
| **Запрос** | **Параметры** | **Ответ** |
| Получение списка сотрудников | command:getList | Список сотрудников в формате JSON |
| Добавление нового сотрудника | command:add  name, family, age, sex, salary | status: true  new\_id = <number> |
| Изменение зарплаты сотрудника | command:changeSalary  id, newSalary | status: true |
| Увольнение сотрудника | command: dismissal  id | status: true |