

Front-end development course

JavaScript

Lecture #11

Async and sync programming

JavaScript is single threaded language by design, but some things may go in parallel (script loading) - that is called service thread, also you can run parallel tasks inside [WebWorkers](#), but it is also async by design.

Async (also called non-blocking) means that creation of the event and execution of that event are separated in time.

Example of blocking programming: loading `<script>` tag with src attribute and without **async** or **defer**.

Example of non-blocking: making a request Ajax request, adding an event listener and so on.

HTTP

HTTP - (HyperText Transfer Protocol) is application layer (see [OSI](#) model) protocol that works on top of TCP/IP.

Protocol is like a language - it has its nouns, verbs and grammar.

HTTP primitives are **request** and **response**, both of them contain Headers with some data in specific format, also **request** contains Request Method (common methods are GET, POST, PUT and DELETE) and **response** contains Status Code. List of all status codes can be found [here](#).

HTTP server - is a software that fully implements protocol specification and can be used to host static files or proxy requests. Most popular servers are [Apache](#) and [nginx](#), they are extremely fast and mature.

Good introduction and overview [here](#). In depth guide to HTTP for front-end developers [part 1](#) and [part 2](#).

XMLHttpRequest and fetch

To work with HTTP servers from browsers we have [XMLHttpRequest](#) class. It's pretty old and has a verbose API. You can use it for asyn or sync requests, for async requests it can be used only with callbacks and that may lead you to [callback hell](#).

Modern replacement to XMLHttpRequest is [fetch](#) it returns [Promise](#) object

Also there is a thing called [WebSockets](#) and it can be used for full-duplex communication with server.

JSON

JSON - JavaScript Object Notation, simple format that looks very similar to object literal in JS. JSON gain a huge popularity because of JavaScript spreading.

JSON is one of the global objects available in browsers and Node, it has 2 methods **parse()** and **stringify()** to convert from JSON to JS object and vise versa.

HTTP nowadays supports a big variety of formats and JSON is also supported.

REST

REST - Representational State Transfer is an architecture pattern, systems that implements REST must meet following requirements:

- Uniform Interface
- Stateless
- Cacheable
- Client-Server
- Layered System

REST explained [here](#) and [here](#) .

Working with backend in React app

Most common technique for working with backend in React app - is to make request using fetch or [axios](#) library and then updating state of component directly or updating Redux store and then triggering state update.

There is a short article about how to work with axios [here](#).

Q&A

Thank You

