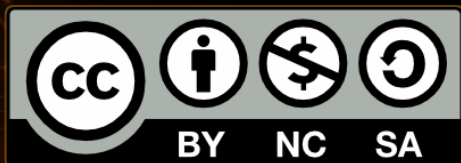


AJAX

Asynchronous JavaScript and XML



Учителски екип

Обучение за ИТ кариера

<https://it-kariera.mon.bg/e-learning/>



Разработка на софтуер



Съдържание

1. Какво е AJAX?

1. XMLHttpRequest

2. Какво е Fetch?

1. GET заявка

2. POST заявка





Какво е AJAX?

Asynchronous JavaScript And XML

Какво е AJAX?

- Зареждане на заден план на динамично съдържание / данни
- Зарежда данни от уеб сървър и ги визуализира
- Два вида AJAX
 - Частично изобразяване на страница
 - Зарежда HTML фрагмент + показва го в <div>
 - JSON услуга
 - Зарежда JSON обект и го показва с JS / jQuery

AJAX: Работен процес

1. HTTP заявка (първоначално зареждане на страницата)



2. HTTP отговор (HTML страница)



AJAX заявка

UI Interaction



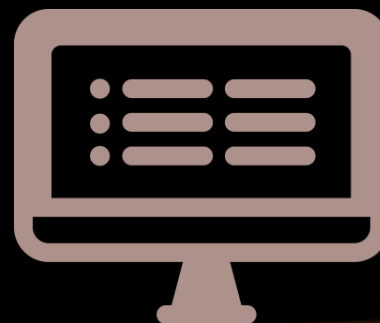
AJAX отговор (асинхронен)

AJAX handler



Връща данни като JSON / HTML

Modify the
page DOM



Web Client



Web Server

XMLHttpRequest – стандартен API за AJAX

```
<button onclick="loadRepos()">Load Repos</button>  
<div id="res"></div>
```

```
function loadRepos() {  
    const req = new XMLHttpRequest();  
    req.onreadystatechange = function() {  
        if (this.readyState == 4 && this.status == 200)  
            document.getElementById("res").textContent =  
                this.responseText;  
    };  
    req.open("GET",  
        "https://api.github.com/users/testnakov/repos", true);  
    req.send();  
}
```

Fetch API

Какво е Fetch?

Какво е Fetch?

- Методът **fetch()** позволява извикване на уеб заявки
- Той е подобен на **XMLHttpRequest (XHR)**. Основната разлика е, че **Fetch API**:
 - Използва обещания (Promises)
 - Е по-прост и чист API

```
fetch('./api/some.json')  
  .then(function(response) {...})  
  .catch(function(err) {...})
```


Проста Request Заявка

- Отговорът на заявката от **fetch()** е **Stream** обект
- Четенето на потока се случва асинхронно
- Когато се извика методът **json()**, се връща обещание
 - Статуса на отговора се проверява (трябва да е 200), преди да се преобрази отговора в JSON

```
if (response.status !== 200) {  
  // handle error  
}  
response.json()  
  .then(function(data) { console.log(data)})
```

GET Заявка

- GET заявката с **Fetch API** изглежда така:

```
fetch('https://swapi.co/api/people/4')  
  .then((response) => response.json())  
  .then((data) => console.log(JSON.stringify(data)))  
  .catch((error) => console.error(error))
```

POST Заявка

- За да направим **POST** заявка, можем да зададем параметри на метода и тялото в опциите на **fetch()**

```
fetch('/url', {  
  method: 'post',  
  headers: { 'Content-type': 'application/json' },  
  body: JSON.stringify(data),  
})
```

Методи на Тялото

- **clone()** - създаване на копие на отговора
- **json()** - разрешава обещания с JSON
- **redirect()** - създаване на ново обещание, но с различен URL адрес
- **text()** - решава обещанието със низ

Задача: Load GitHub Repos with AJAX

GitHub username:

```
<input type="text" id="username" value="k1r1L" />  
<button onclick="loadRepos()">Load Repos</button>  
<ul id="repos"></ul>
```

```
function loadRepos() {  
    // AJAX call ...  
}
```

GitHub username:

Load Repos

- [k1r1L/Angular-2-Demos](#)
- [k1r1L/Angular-Sli.do](#)
- [k1r1L/awesome-interview-questions](#)
- [k1r1L/CSharp-Web-Development-Basics](#)
- [k1r1L/CSharp-Web-MVC-Frameworks-ASP.NET](#)
- [k1r1L/Databases-Advanced-Entity-Framework](#)
- [k1r1L/Databases-MS-SQL-Server-Exercises](#)
- [k1r1L/Express-Demo-Server](#)
- [k1r1L/express-js-exercises](#)

Решение: Load GitHub Repos with AJAX

```
function loadRepos() {  
  const repos = document.getElementById("repos");  
  repos.innerHTML = '';  
  const username = document.getElementById("username").value;  
  const url = `https://api.github.com/users/${username}/repos`;  
  
  fetch(url)  
    .then((response) => response.json())  
    .then((data) => displayRepos(data))  
    .catch((err) => displayError(err))  
  
  // Continues on the next slide...
```

Решение: Load GitHub Repos with AJAX

```
function createRepo(name, url) {  
  // ...  
}  
function displayRepos(repoItems) {  
  repoItems.forEach(repo => {  
    const { full_name, html_url } = repo;  
    const repoItem = createRepo(full_name, html_url);  
    repos.appendChild(repoItem)  
  });  
}  
function displayError(err) {  
  const listItem = document.createElement('li');  
  listItem.textContent = err;  
  repos.appendChild(listItem); }}
```

Какво научихме в този час?

- Какво е AJAX?
 - XMLHttpRequest
- Какво е Fetch?
 - GET заявка
 - POST заявка



AJAX



Въпроси?



Министерство на образованието и науката (МОН)

- Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма "**Обучение за ИТ кариера**" на МОН за подготовка по професия "Приложен програмист"



Министерство
на образованието
и науката



Национална
програма
„Обучение за
ИТ кариера“

- Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под свободен лиценз **CC-BY-NC-SA**



SoftUni
Foundation

