

22 - AJAX, REST

Klasická a asynchronní komunikace

- rozdíl v **asynchronní komunikaci** oproti normální:
 - Requesty můžou probíhat na pozadí a aplikace stále poběží
 - nejčastěji použito na webových stránkách v client-side kódu
 - JavaScript + AJAX
 - možno aktualizovat data bez nutnosti znovunačítání stránky

AJAX

- Asynchronous Javascript And XML
- technologie pro asynchronní komunikaci JavaScriptu na clientovi a serveru
- lze díky němu měnit obsah stránky bez nutnosti znovunačtení stránky
- AJAX pošle Request na server a poté se zachová rozdílně podle odpovědi (HTTP kódu)
- přenáší se méně dat, ale častěji

Promise

- objekt, který reprezentuje příslib dokončení asynchronní operace
- díky němu může aplikace běžet dál a pracuje s Promise jako výsledkem operace, který ještě není známý
- po dokončení aplikace se zavolá tzv. callback
 - = metoda, která se volá po dokončení a obsahuje předem definované akce
- více asynchronních operací lze jednoduše přes callbacky řetěžit

Fetch API

- nahrazuje v JavaScriptu `XMLHttpRequest` globálně přístupnou metodou `fetch()`
- funguje pouze na moderních prohlížečích
- využívá Promise

```
Promise = fetch(URL, data)
```

REST

- Representational state transfer
- způsob, jak pracovat s daty na serveru pomocí HTTP metod
- dělí se podle metod např. na:
 - `GET` - pro získání dat
 - `POST` - pro nahrání dat
 - `PUT` - pro nahrazení dat
 - `DELETE` - pro smazání dat
- "RESTful aplikace" - využívají princip REST