10 - AJAX, REST

# Klasická komunikace

* Uživatel chce vidět podrobnosti o nějaké věci na stránce, rozklikne ji v prohlížeči
* Klient (prohlížeč) žádá server o celou stránku
* Server zpracuje požadavek, vrací celou novou stránku
* Výhody:
  + Jednoduchá implementace
* Nevýhody:
  + Náročné na data
    - Kvůli načítání celé stránky a přenášení dat jako CSS, JS
  + Náročné na výkon
    - Pokaždé je nutno stránku sestavit (server) a vyrenderovat (klient)

# Asynchronní komunikace

* Uživatel chce vidět podrobnosti o nějaké věci na stránce, rozklikne ji v prohlížeči
* Klient (prohlížeč) žádá server o konkrétní data
* Server zpracuje požadavek, vrací pouze požadovaná formátovaná data
* Klient obdrží data a aktualizuje jen nutné části stránky
* Výhody:
  + Přenos méně dat (ovšem tendence k častějšímu přenosu)
  + Není potřeba sestavovat HTML stránku na straně serveru
* Nevýhody:
  + Nutná podpora v prohlížečích
  + Čtečky webových stránek mají s aktualizovaným obsahem problém
  + Některé web crawlery nemusí spouštět JS – neindexuje se AJAX obsah (Google crawler JS podporuje)

# AJAX

* Asynchronous JavaScript and XML (nebo JSON)
* Technologie sloužící k vytváření asynchronních webových aplikací
* Získávání dat po nahrání stránky
* Získání dat bez nutnosti refreshe
* Odesílání dat serveru na pozadí
* Změna části stránky
* Kombinace známých technologií (HTML, CSS, JS, JSON)
* Dříve byl problém s přepisováním historie
* Složité pro vyhledávače
* React, našeptávače, chat

# Promise

* Promise vrací hodnotu o úspěchu a nebo selhání asynchronní funkce
* Má pouze 3 stavy:
  + Pending - čekání
  + Fulfilled – data přišla
  + Rejected – chyba

# Fetch API

* Rozhraní pro načítání dat
* Vylepšené XMLHttpRequest
* Vyžádá data, asynchronní request
* Request, response
* Vrací objekt Promise
* Nefunguje v IE
* Globální Fetch()

# API

* Application Programming Interface
* Rozhraní pro programování
* Svírka funkcí, metod a různých knihoven
* Existují grafická API (OpenGL, DirectX), WEB API
* API na serveru – webové API s endpointy

# http Dotazy

* http slouží ke komunikaci mezi klientem a serverem
* Request-Response
* GET – žádá o data
* POST – vytváří
* PUT – aktualizace celé entity
* PATCH –aktualizace části entity
* DELETE – mazání
* OPTIONS – dotaz na server, jaké podporuje metody

# REST API

* Druh API
* Representational State Transfer
* Pracujeme přímo s daty
* Používá protokol http
* Implementování CRUD (Create, Read, Update, Delete)