

# Design aplikace

Miroslav Hlávka

2021



# Obsah

1. Historie vývoje webu
2. Aktuální design
3. Front-end
4. Back-end
5. REST API
6. AWS Lambda BE

# Historie vývoje webu

- ▶ Definice HTML (1990)
- ▶ PHP, JavaScript (1995), CSS (1998), Macromedia Flash s Action Scriptem (2000), jQuery (2006)
- ▶ AngularJS (2010), Bootstrap (2011), ReactJS (2013), HTML5 (2014)

# Historie vývoje webu

- ▶ Multiple Page Applications - statické HTML se server-side renderingem (PHP, Java, apod.)
- ▶ AJAX a jQuery řešily upload stránky bez nutnosti reload
- ▶ Rich Internet Applications - Flash, Silverlight, Java Applets, JavaFX
- ▶ Single Page Applications - client-side UI renderování a back-end zdroj dat

# Front-end

- ▶ Aplikace běžící na straně klienta
- ▶ Nejpoužívanější je JS (alternativa WebAssembly)
- ▶ Podpora mobilních zařízení
- ▶ Data jsou získávána z backendu asynchronním voláním

## Obliba JS frameworků

<https://blog.bitsrc.io/top-5-javascript-frameworks-past-present-and-future-8b6fda39de02>

# Back-end

- ▶ Podpora mnoha jazyků (Java, Go, PHP, apod.)
- ▶ Potřeba běhu 24\*7
- ▶ Často implementuje business logic, správu dat, management uživatelů apod.

# REST API

- ▶ Komunikace mezi FE a BE pomocí různých metod (SOAP, CORBA - 1998)
- ▶ REST API (2000)
- ▶ Využívá HTTP protokol a jeho request types (GET, POST, PUT, OPTIONS, DELETE, apod.)
- ▶ Data jsou předávána nejčastěji formou JSON
- ▶ Alternativou jsou GraphQL, gRPC apod.

# Swagger

- ▶ Jazyk pro popis REST API (2011)
- ▶ Představuje jednoznačnou specifikaci API
- ▶ Umožňuje použití mock nástrojů při vývoji FE/BE
- ▶ Je implementován řadou jazyků a umožňuje automatické generování modelu

## Swagger editor

<https://editor.swagger.io>



# AWS Lambda BE

- ▶ Vývoj pro serverless aplikace se značně liší od zažitých způsobů
- ▶ AWS Cloud9 IDE
- ▶ Lokálně pak potřebujete kód pokaždé nahrát do cloudu nebo použijete např. AWS SAM Local CLI