

# C# 3

**Zdolej pokročilé techniky  
programování**

Martin Černil, Václav Kučera

28.9. 2024



## Použití materiálů

Toto dílo je licencováno pod

Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International  
License



## C#3

Co nás čeká a jak to bude probíhat 😊

# OSNOVA

- 25. 9.      Organizace, Visual Studio Code, založení GitHub repozitáře, Opakování
- 2. 10.     Webové technologie a WebAPI
- 9. 10.    Interakce s REST API
- 16. 10.   Testování kódu
- 23. 10.   Databáze
- 30. 10.   Generika + Repository design pattern
- 6. 11.    Opakování
- 13. 11.   Refactoring, Dependency Injection
- 20. 11.   Blazor
- 27. 11.   HTTP Klient
- 4. 12.    Asynchronní programování, Refactoring
- 11. 12.   Opakování, závěr

# **STRUKTURA LEKCE**

## Časově

- Začínáme v 18:00, končíme ve 20:30
- Plánujeme vždy dvě přestávky – zhruba v 18:50 a 19:50

## Obsahově

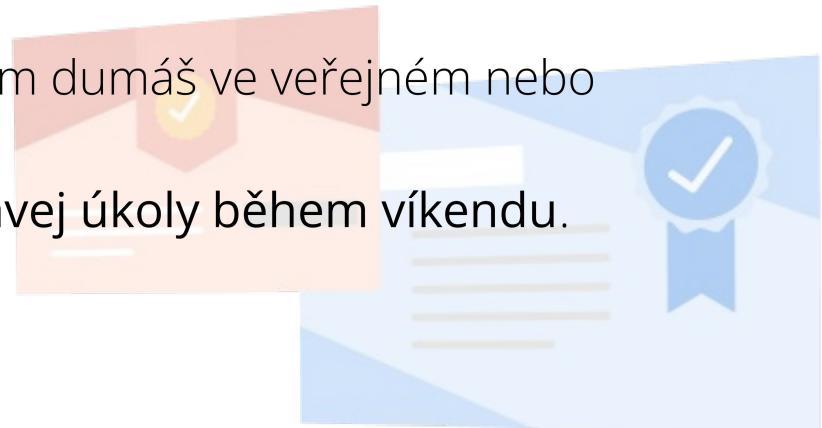
- Rekapitulace úkolů z minулé lekce
- Výklad nové látky
- Breakout rooms - samostatná práce na úkolech s kouči

## **ÚKOLY**

Ale budou samé „jednoduché“, můžete nám  
věřit 😊

# ÚKOLY

- Úkoly jsou **povinné** i nepovinné, nepovinné jsou **silně doporučené**.
- Odevzdávání bude probíhat do tvého vlastního GitHub repozitáře, kde dostaneš od svého kouče **individuální zpětnou vazbu**.
- Snažíme se opravit opravdu vše a odpovědět na všechny dotazy.  
**Využij toho.**
- Připoj se na **Discord** a diskutuj nad čím dumáš ve veřejném nebo skupinkovém kanálu.
- Pro získání včasné odpovědi odevzdávej úkoly během víkendu.



## **BUDE TO BOLET?**

- Doufáme, že nikoliv! Ale přece jen se tento kurz zabývá pokročilými tématy, takže buď na pozoru a připrav si dotazy.
- To, že jsi na úkolu strávila xy hodin, vůbec nevypovídá o tom, jestli na to máš nebo ne, jestli jsi chytrá, nebo ne, atd. Je to běžná součást učení, znamená to, že pracuješ opravdu intenzivně na tom, aby ses posunula dál a jde ti to!
- Neboj se tomu věnovat čas, je to zase úplně nová věc a na její vstřebání je potřeba zažít i „nepříjemné chvilky zoufalství“.
- Nezapomeň ale na pravidelný odpočinek.
- Odměnou ti bude funkční webová aplikace, pochopení, jak ji vytvořit a hlavně skvělý pocit!



**DOTAZY**

## **ŽÁDNÝ DOTAZ NENÍ HLOUPÝ**

- Vždy se najde v místnosti někdo, kdo se na to chce také zeptat, ale nemá odvahu.
- Nezapomeň, že jsou tu i **koučové**, kteří sem přišli, aby ti byli k dispozici. Jsou to tví důležití **mentorové**, kteří tě budou vést správným směrem.
- Vyplňuj prosím zpětnou vazbu. Dává nám to hrozně moc!

## DISCLAIMER

- Toto je první běh nového kurzu. Můžeme se setkat s mnohými obtížemi, technického či edukačního typu.
- Vaše **zpětná vazba je naprosto klíčová!** Potřebujeme vědět, kde přidat a kde ubrat, co se líbilo, co ne. Co bylo nejasné a nad čím jste dumaly.
- Časová dotace kurzu neumožňuje jít do přílišné hloubky některých technologií (vystačily by na samostatný kurz).  
Důležité je si je vysvětlit a pochopit jejich princip.
- Očekávejte, že některé informace si budete dohledávat samy (z anglických zdrojů/dokumentace). Zdroje poskytneme a kouči vám budou k dispozici.  
Vyhledávání informací je zásadní pro vaši budoucí pozici.

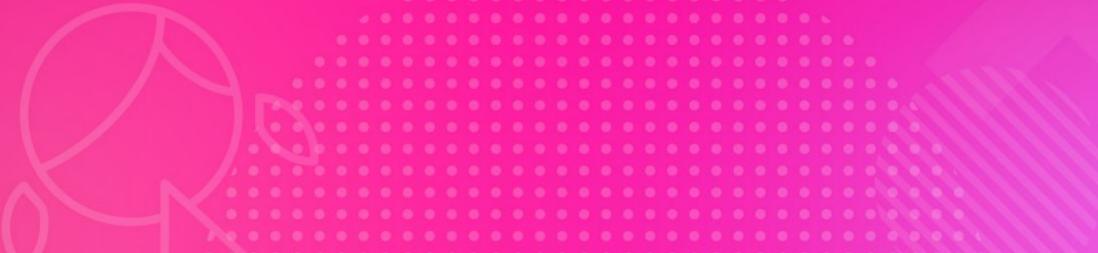
# IT POZICE



# IT POZICE

Computing-Core Disciplines	Computing-Intensive Fields	Computing-Infrastructure Occupations
Artificial intelligence	Aerospace engineering	Blockchain administrator
Cloud computing	Autonomous systems	Computer technician
Computer science	Bioinformatics	Data analyst
Computer engineering	Cognitive science	Data engineer
Computational science	Cryptography	Database administrator
Database engineering	Computational science	Help desk technician
Computer graphics	Data science	Identity theft recovery agent
Cyber security	Digital library science	Network technician
Human-computer interaction	E-commerce	Professional IT trainer
Network engineering	Genetic engineering	Reputation manager
Programming languages	Information science	Security specialist
Programming methods	Information systems	System administrator
Operating systems	Public Policy and Privacy	Web identity designer
Performance engineering	Instructional design	Web programmer
Robotics	Knowledge engineering	Web services designer
Scientific computing	Management information systems	
Software architecture	Network science	
Software engineering	Multimedia design	
	Telecommunications	

# Backend vývojářka



# Lekce 01

Intro

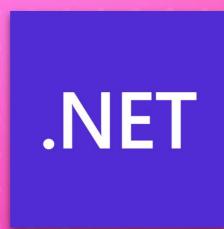
## Lekce 01 - Intro

1. Rozjetí **IDE** (Integrated Development Environment)
  - a. .NET SDK 8.0
  - b. Visual Studio Code
    - i. Import nastavení
  - c. FizzBuzz projektík

Všichni jsme na jedné lodi, máme připravené vývojové prostředí, dokážeme se v něm orientovat a něco si zaprogramujeme ;)



## S ČÍM BUDEME PRACOVAT



.NET 8.0

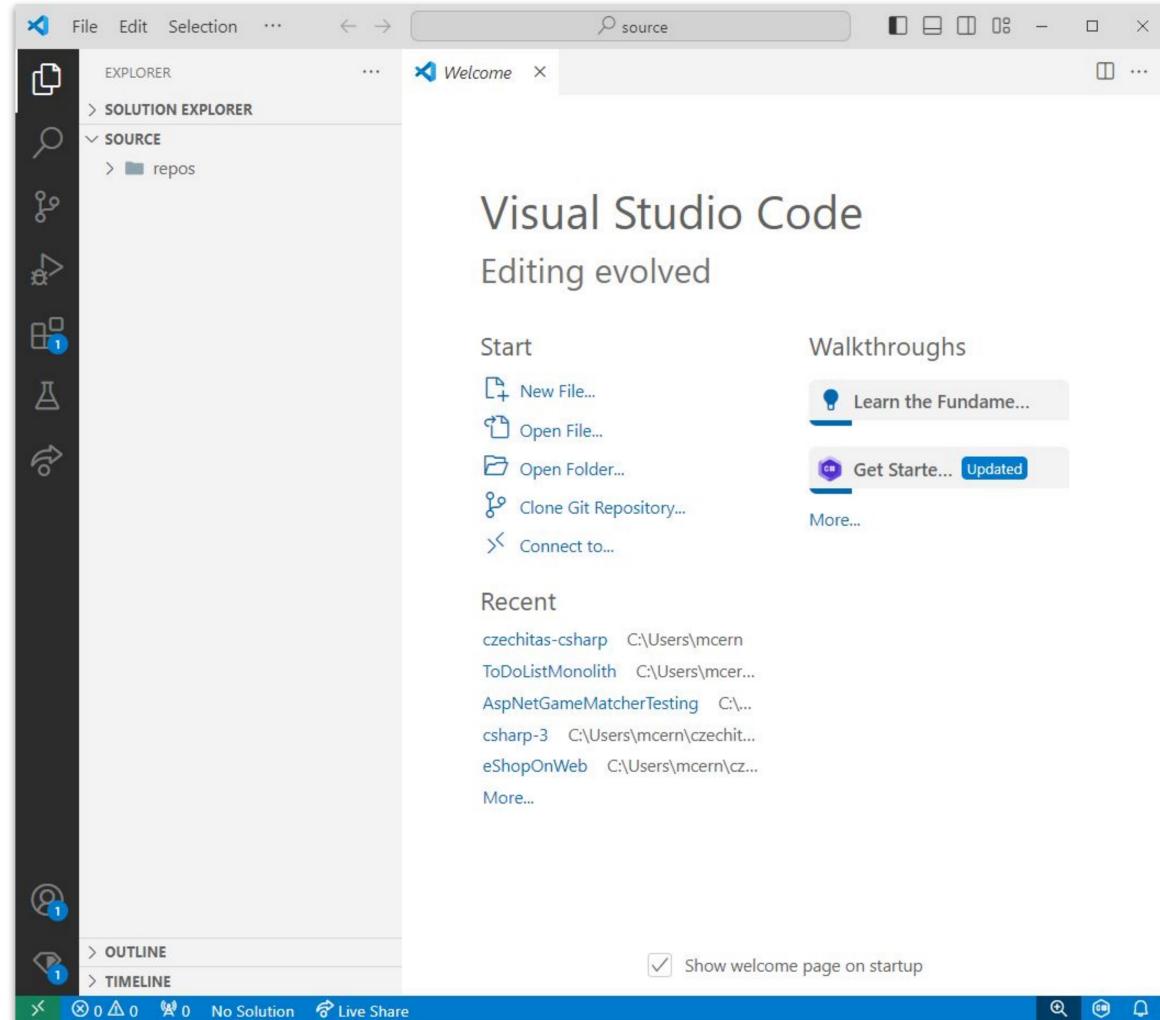
<https://dot.net/download>



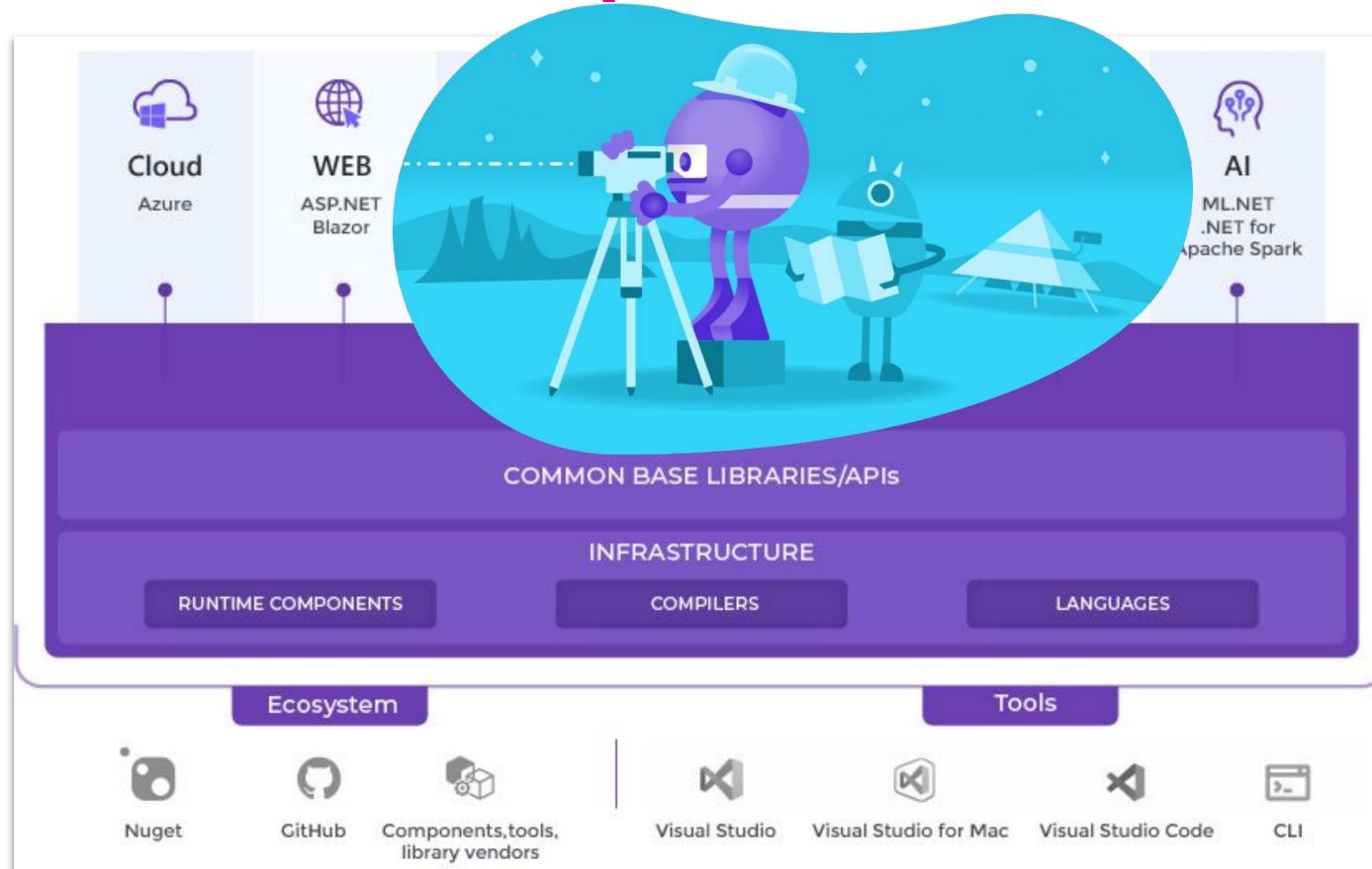
Visual Studio Code

<https://code.visualstudio.com>

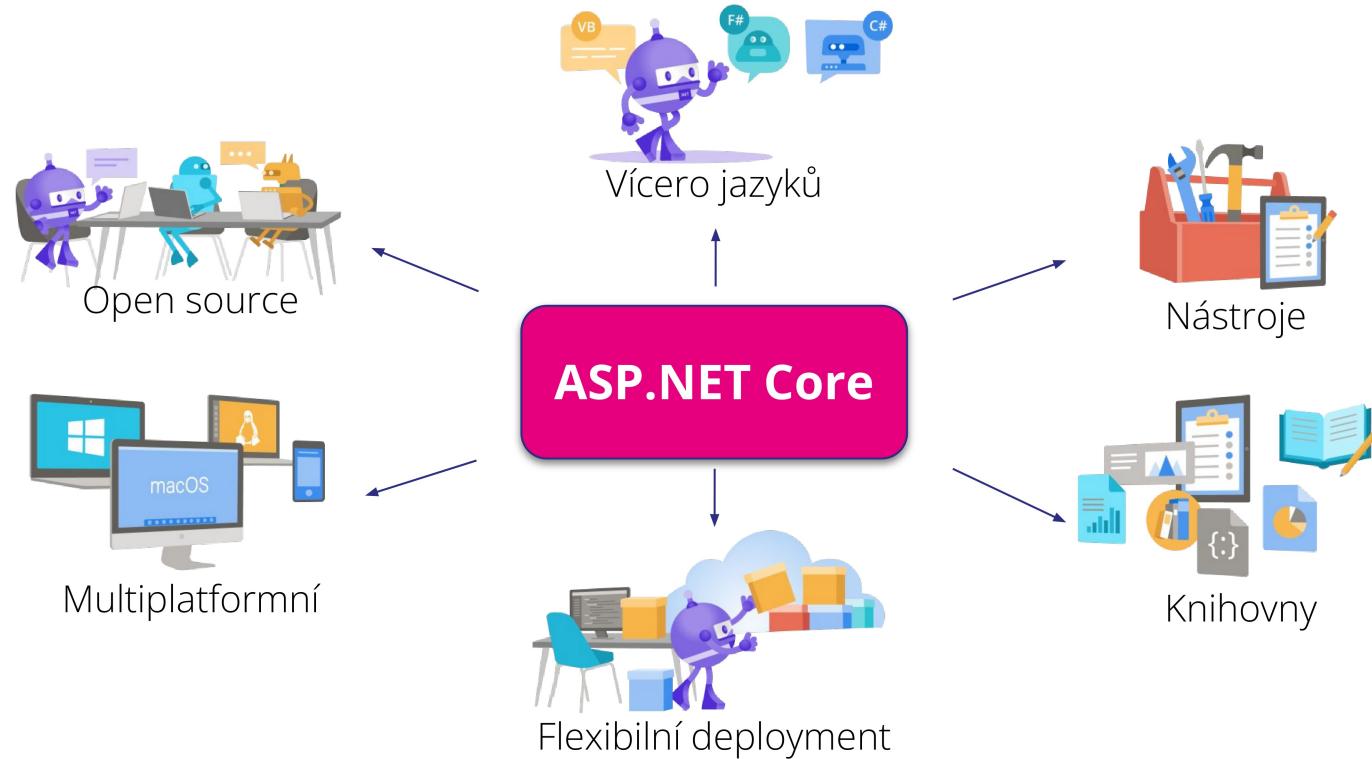




# Na co vše se dá .NET použít



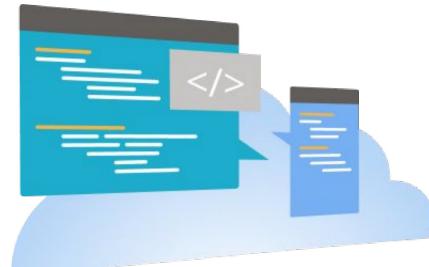
# Co je to ASP.NET Core?



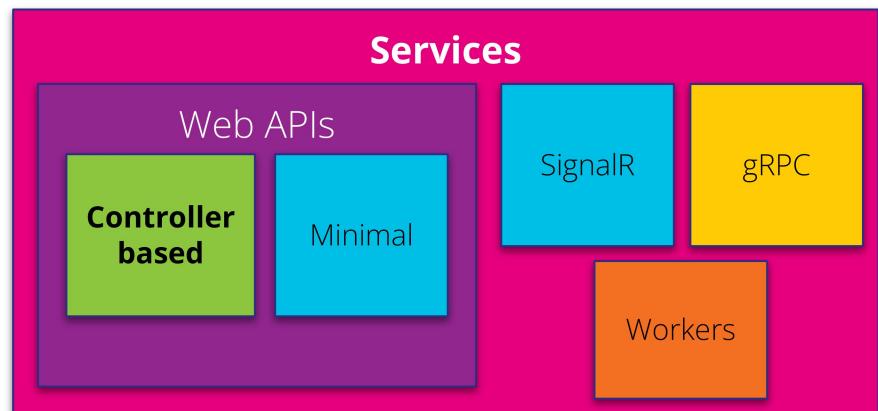
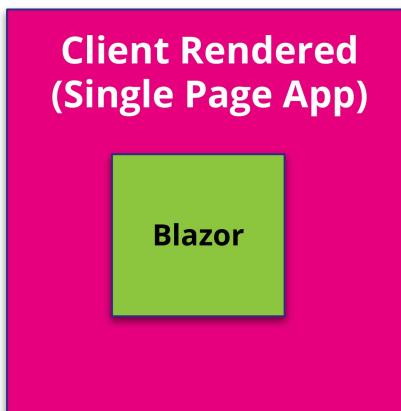
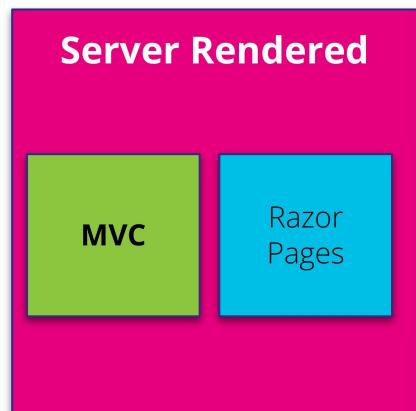
# Co se dá s ASP.NET Core vytvořit?



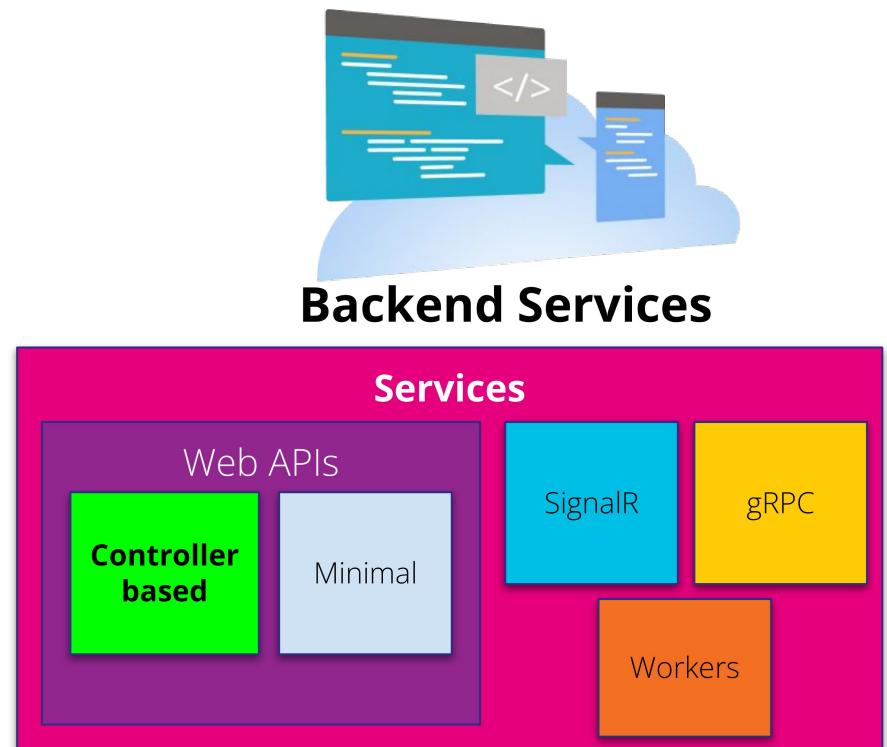
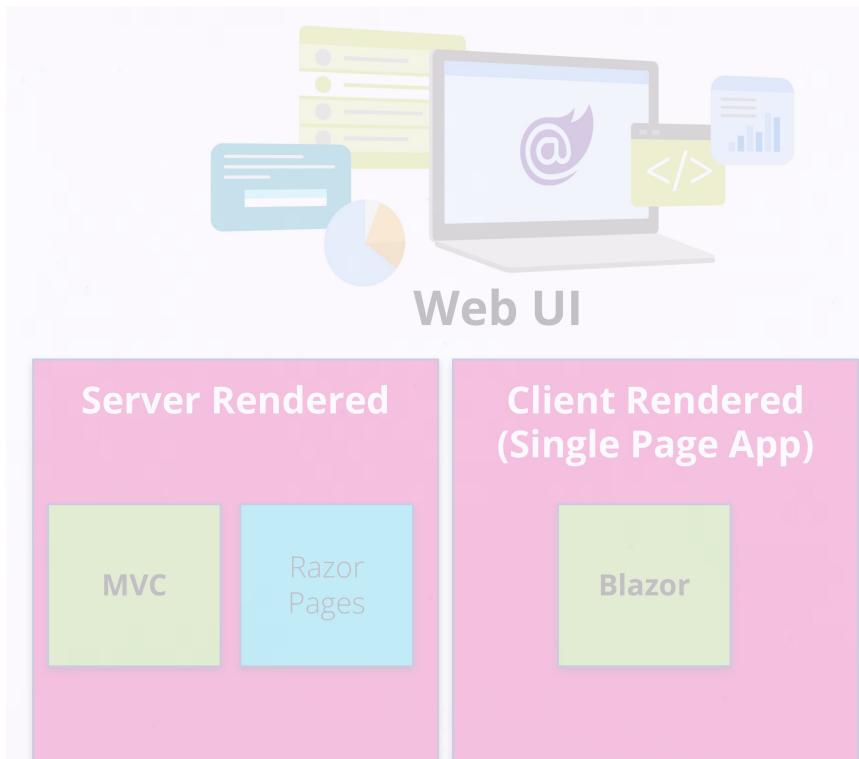
Web UI



Backend Services



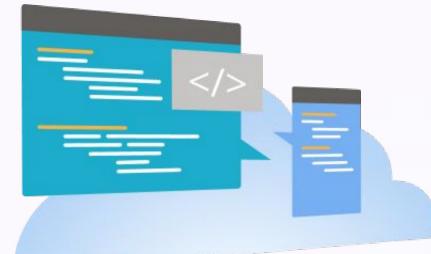
# Co se dá s ASP.NET Core vytvořit?



# Co se dá s ASP.NET Core vytvořit?

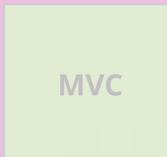


Web UI

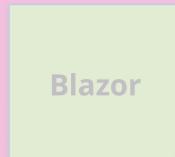


Backend Services

## Server Rendered



## Client Rendered (Single Page App)



## Services

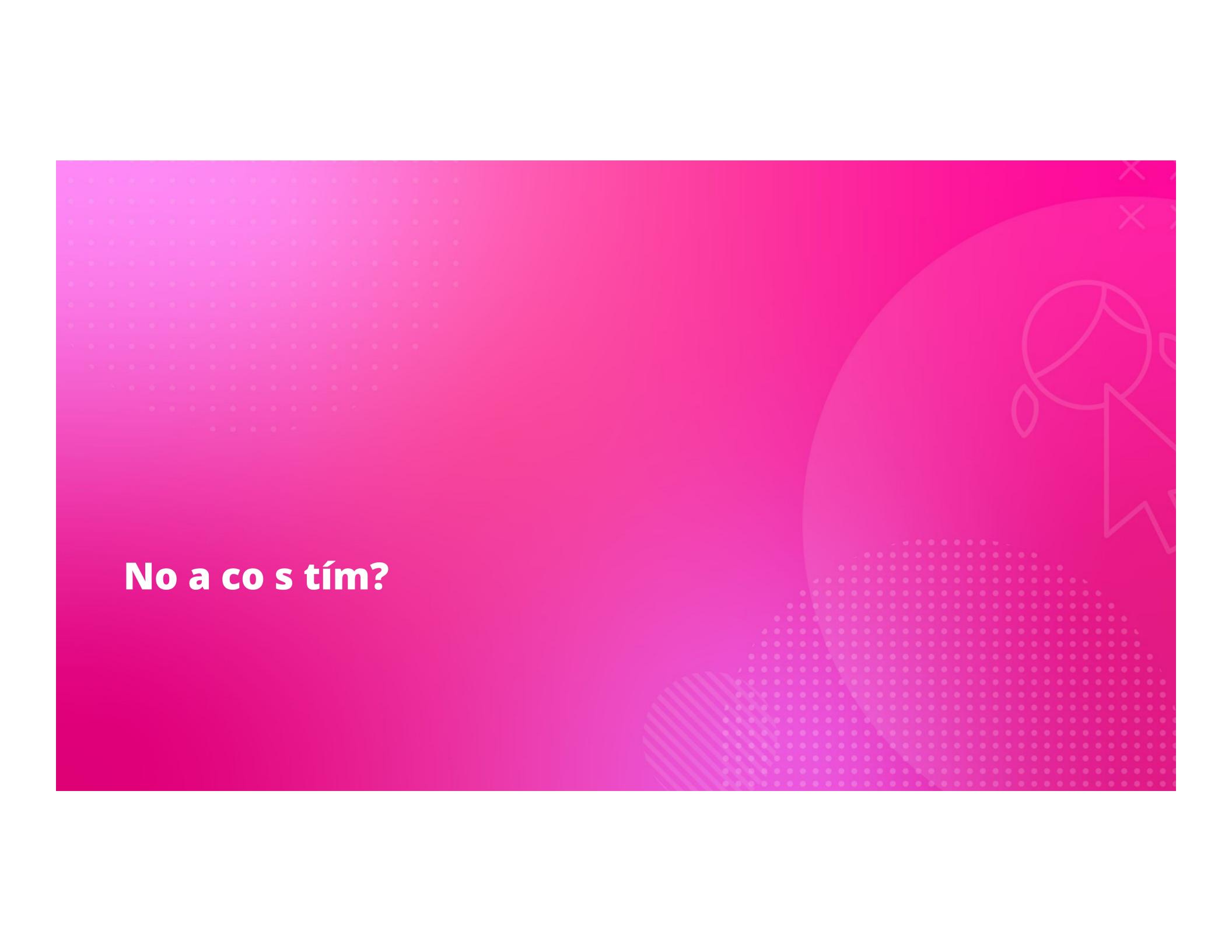
### Web APIs



Minimal



Workers



**No a co s tím?**

# Demo

- Faketsy
- EShopOnWeb
- GameStore
- GameMatcher

# HOSPODA?

Aneb zpětnou vazbu si rádi vyslechneme i ~~piva~~ ústně.



**POJĎME PROGRAMOVAT** 😊

# ROZEHŘÁTÍ

FizzBuzz

# FizzBuzz - Zadání

FizzBuzz je jednoduchá číselná hra, ve které počítáš od 1 do X a nahrazuješ určitá čísla slovy "Fizz" a/nebo "Buzz" podle určitých pravidel.

Vytvoř třídu `FizzBuzz` s metodou `CountTo(int lastNumber)`, která vytiskne do konzole čísla od 1 do `lastNumber`, každé na nový řádek.

Místo čísel **dělitelných 3** by měla metoda vypsat do konzole "**Fizz**". Místo čísel **dělitelných 5** by měla vypsat do konzole "**Buzz**". A místo čísel **dělitelných 3 i 5** by měla vypsat do konzole "**FizzBuzz**".

Odkaz

```
1
2
Fizz
4
Buzz
Fizz
7
8
Fizz
Buzz
11
Fizz
13
14
FizzBuzz
16
```

Očekávaný výstup z konzole

# Začněte



# Začněte?



# Začněme spolu!

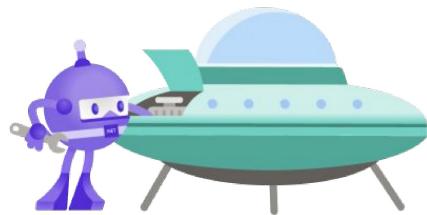


# KONZOLE

- Slouží pro zadávání příkazů a zobrazování výstupů (výpis hodnot) u konzolových aplikací
- Ve VSCode ji zapínáme pomocí **ctrl + ;** (**Cmd + `** pro Mac)



**dotnet --version**  
**dotnet --info**

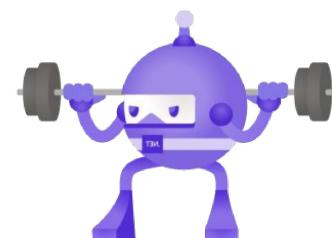


## KONZOLE - Příkazy Dotnet CLI

Pro práci v tomto kurzu budeme potřebovat taky pár příkazů.  
Nebojte se, vždy je budete mít předem připravené na dosah ruky  
a kláves **ctrl+c** ;)

**dotnet new <template name>**  
(přidává nové projekty na základě šablony)

<b>dotnet build</b>	(přeložit program)
<b>dotnet run</b>	(spustit program)
<b>dotnet watch</b>	(interaktivní mód)
<b>dotnet test</b>	(spustí všechny testy)



# KONZOLE - Příkazy Dotnet CLI

Aktuálně potřebujeme jen tyto

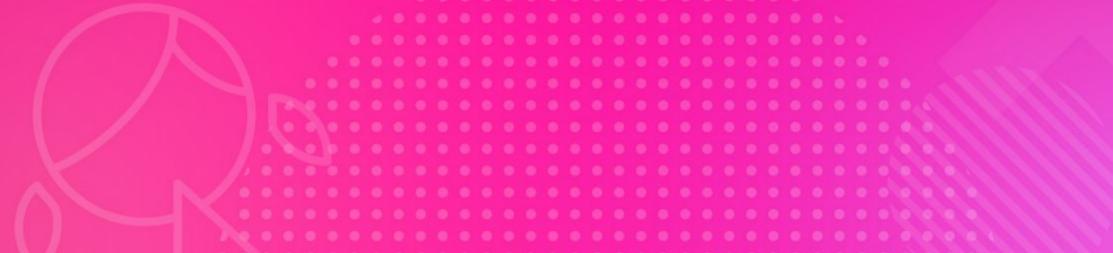
```
dotnet new solution --name FizzBuzz
```

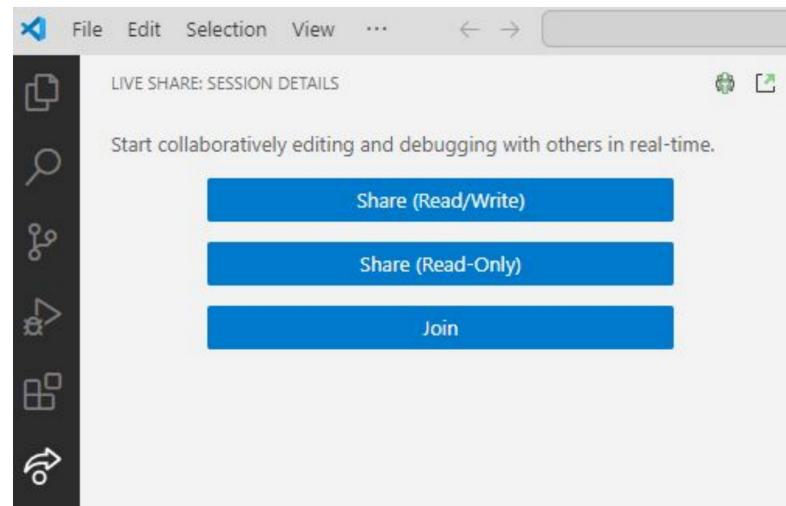
```
dotnet new console --name FizzBuzz
```

```
dotnet sln add FizzBuzz (přidá projekt FizzBuzz do solution)
```

**dotnet watch** (interaktivní mód)

# LiveShare





## **LEKCE 02**

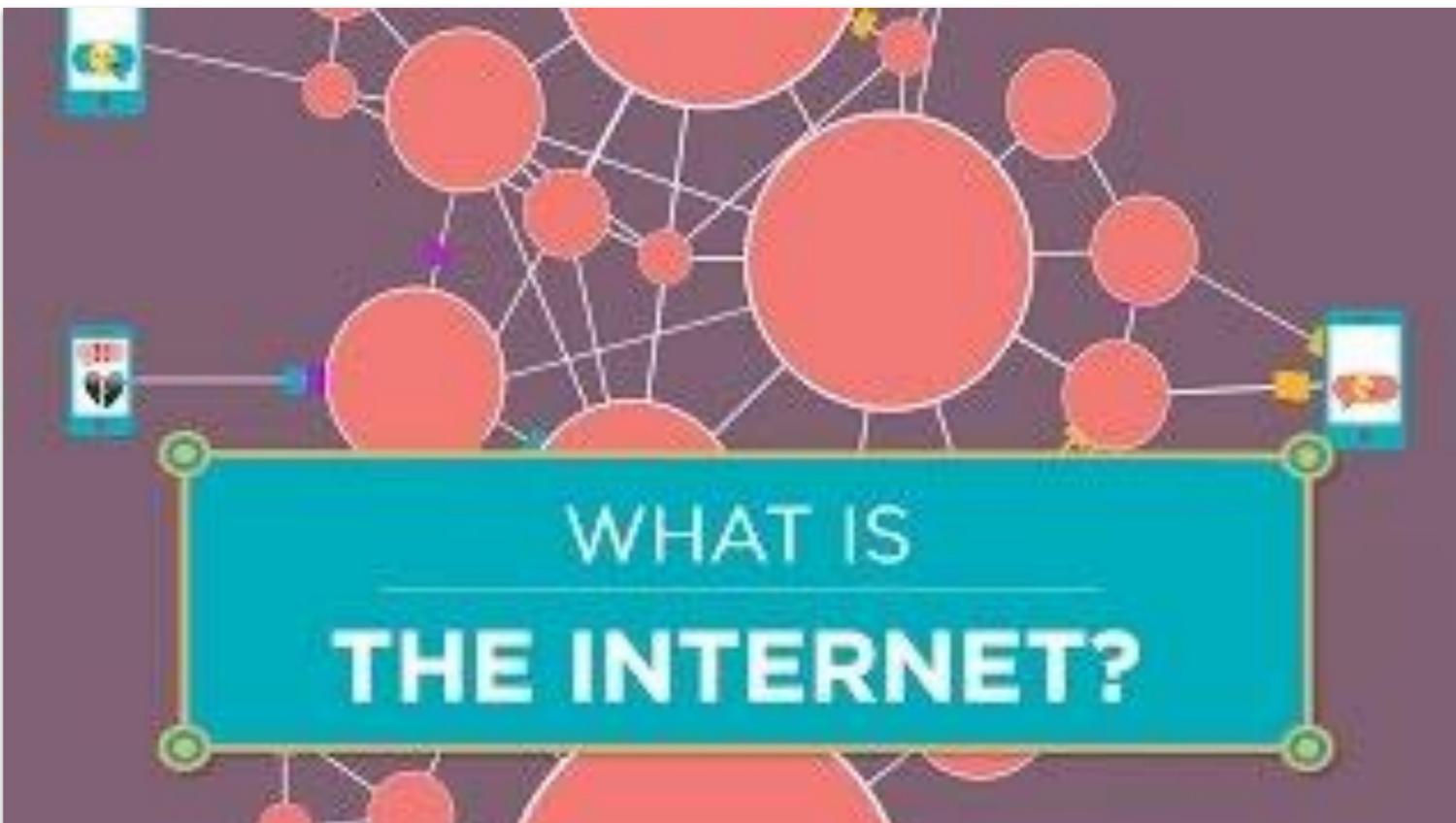
Webové technologie

## Lekce 02 - Webové technologie

- Teorie
  - Internet
  - Odkazy **URL**
  - **Webové stránky**
  - Prohlížeče
  - **Webová API (REST, CRUD)**
- Založení projektu ToDoList.WebApi
  - Orientace v projektu
  - Význam

Máme povědomí o internetových technologiích, víme co jsou webová API a dokážeme se zorientovat v ASP.NET projektu ;)

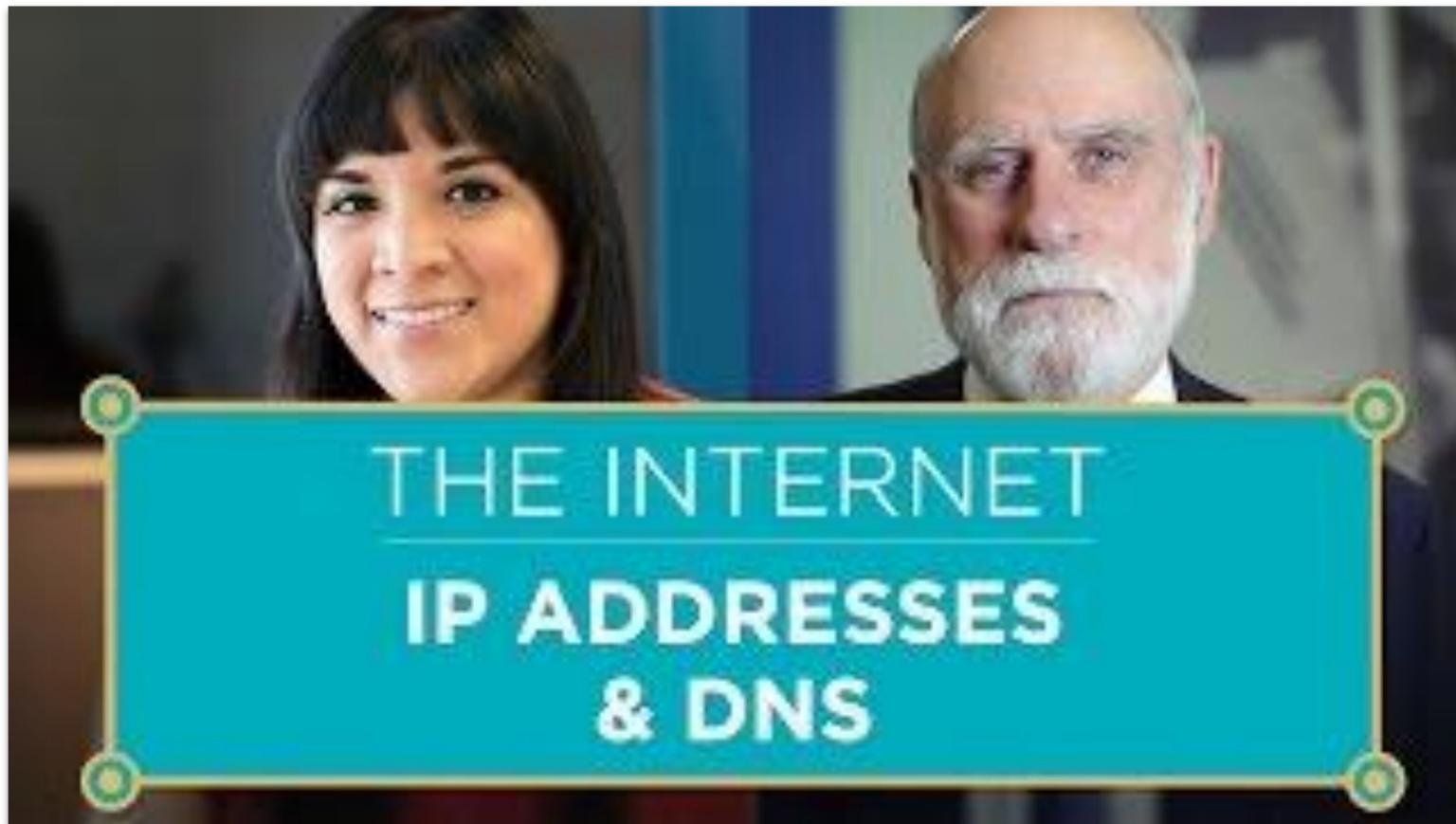
# Internet



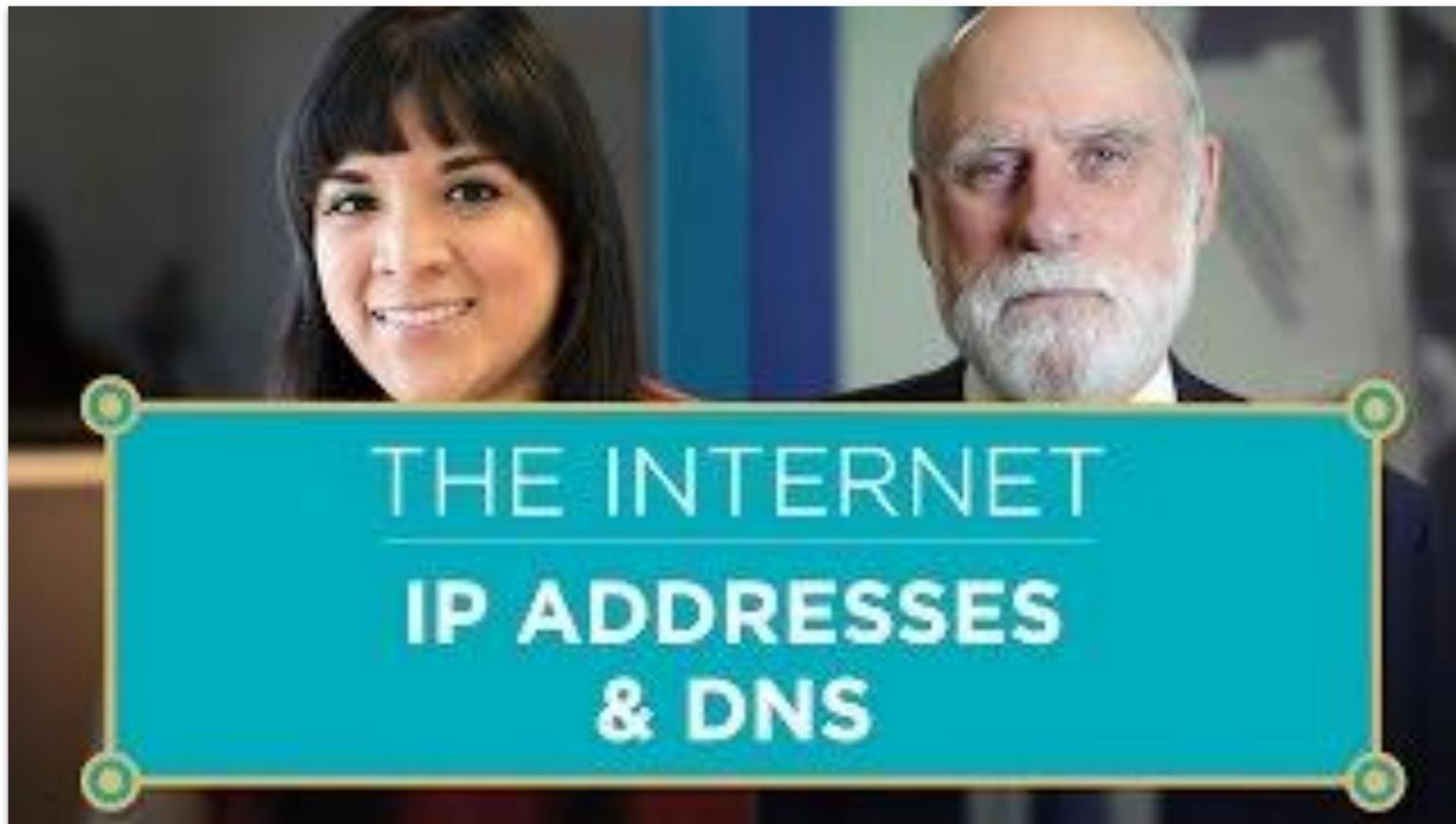
# WHAT IS THE INTERNET?

# **IP Adresa**

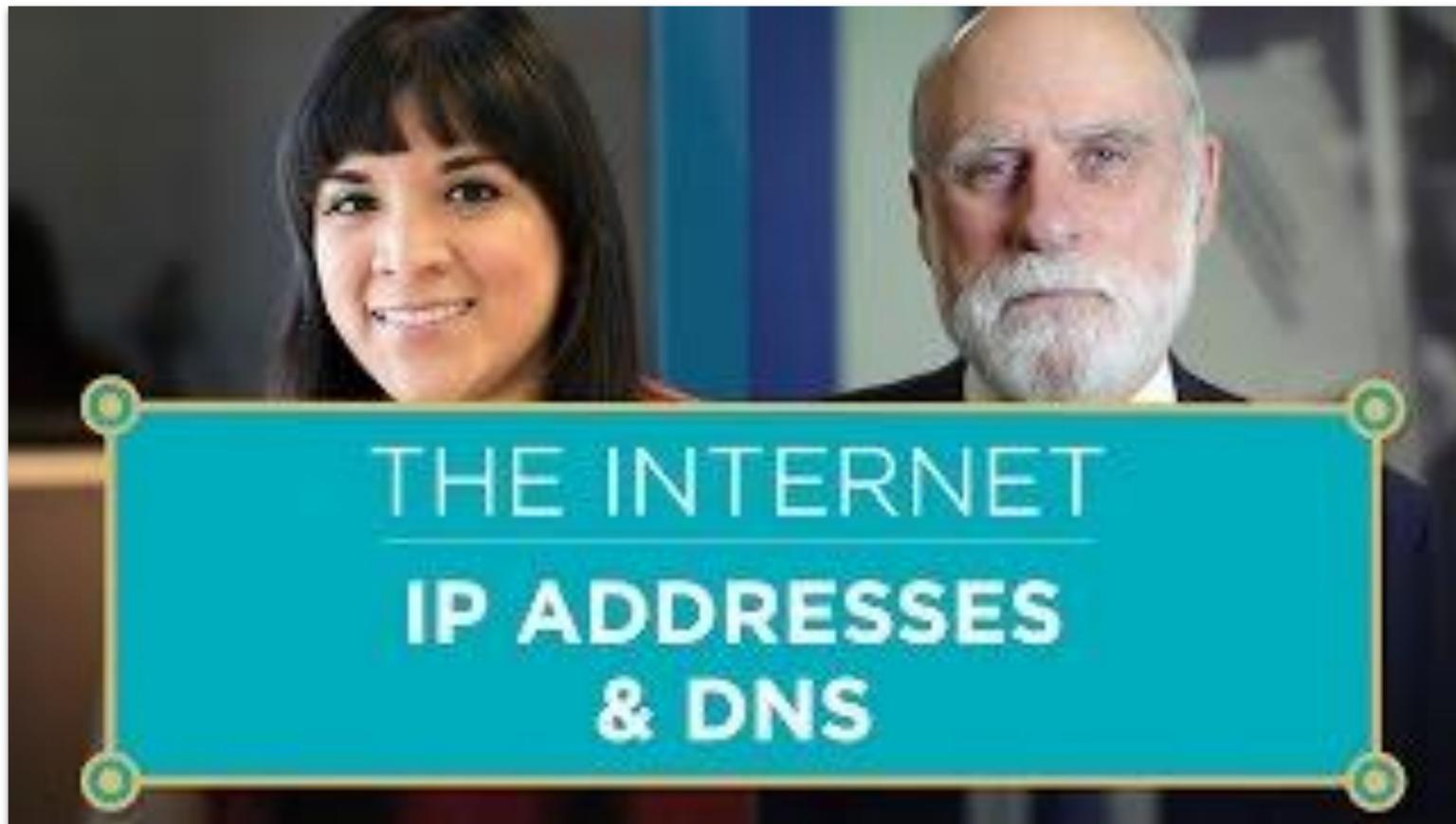
## **Internet Protocol Address**



## THE INTERNET IP ADDRESSES & DNS



## THE INTERNET IP ADDRESSES & DNS



## THE INTERNET IP ADDRESSES & DNS

# DNS

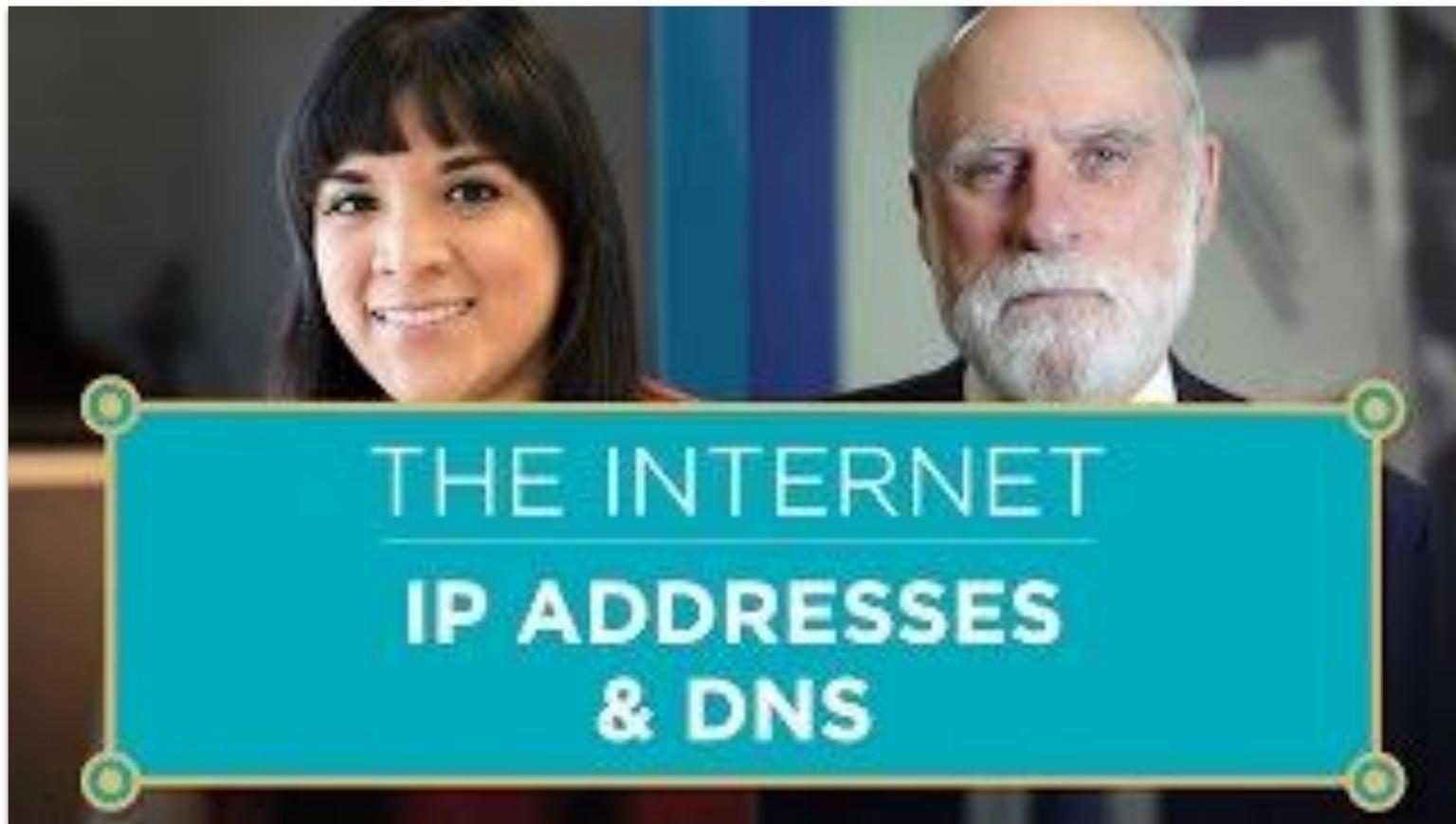
## Domain Name System

I KNOW A GUY  
**WHO KNOWS A GUY**  
WHO KNOWS  
**ANOTHER GUY**

Br  
Ba

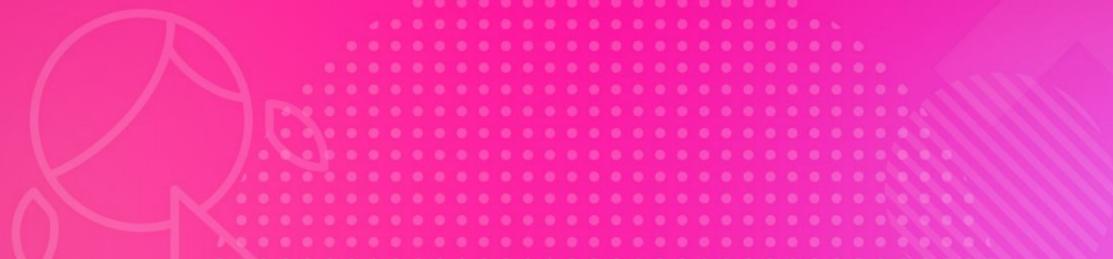
NETFLIX

<https://x.com/NetflixUK/status/559788220658241537/photo/1>



## THE INTERNET IP ADDRESSES & DNS

# Prohlížeče





THE INTERNET

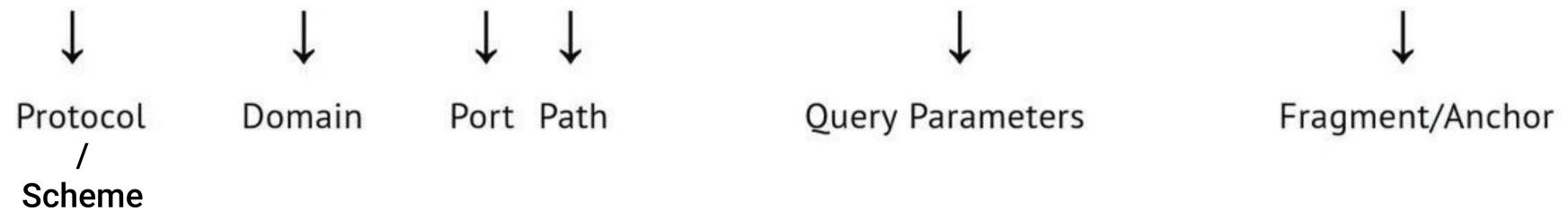
HTTP  
& HTML

# **URL**

## **Uniform Resource Locator**

## URL Anatomy

https://example.com:80/blog?search=test&sort\_by=created\_at#header



<https://medium.com/@ndbeladiya720/what-is-a-url-example-structure-of-url-92cda07a9dcc>

# **HTTP**

## **HyperText Transfer Protocol**



THE INTERNET

HTTP  
& HTML

# HTTP GET

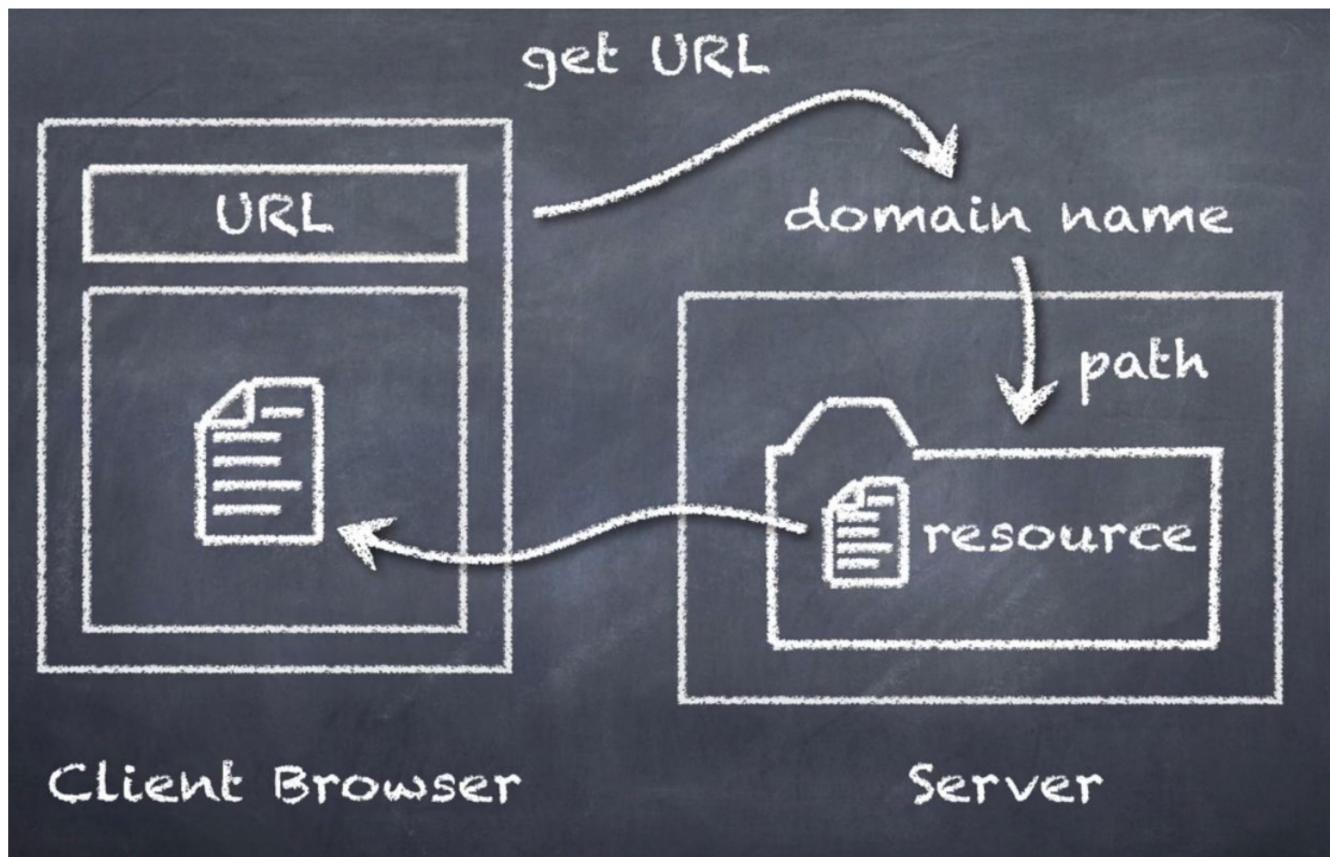


**DEMO**  
**V PROHLÍŽEČI**

<https://www.google.com/search?client=chrome&q=dog>



Zde se  
schovává  
port :80

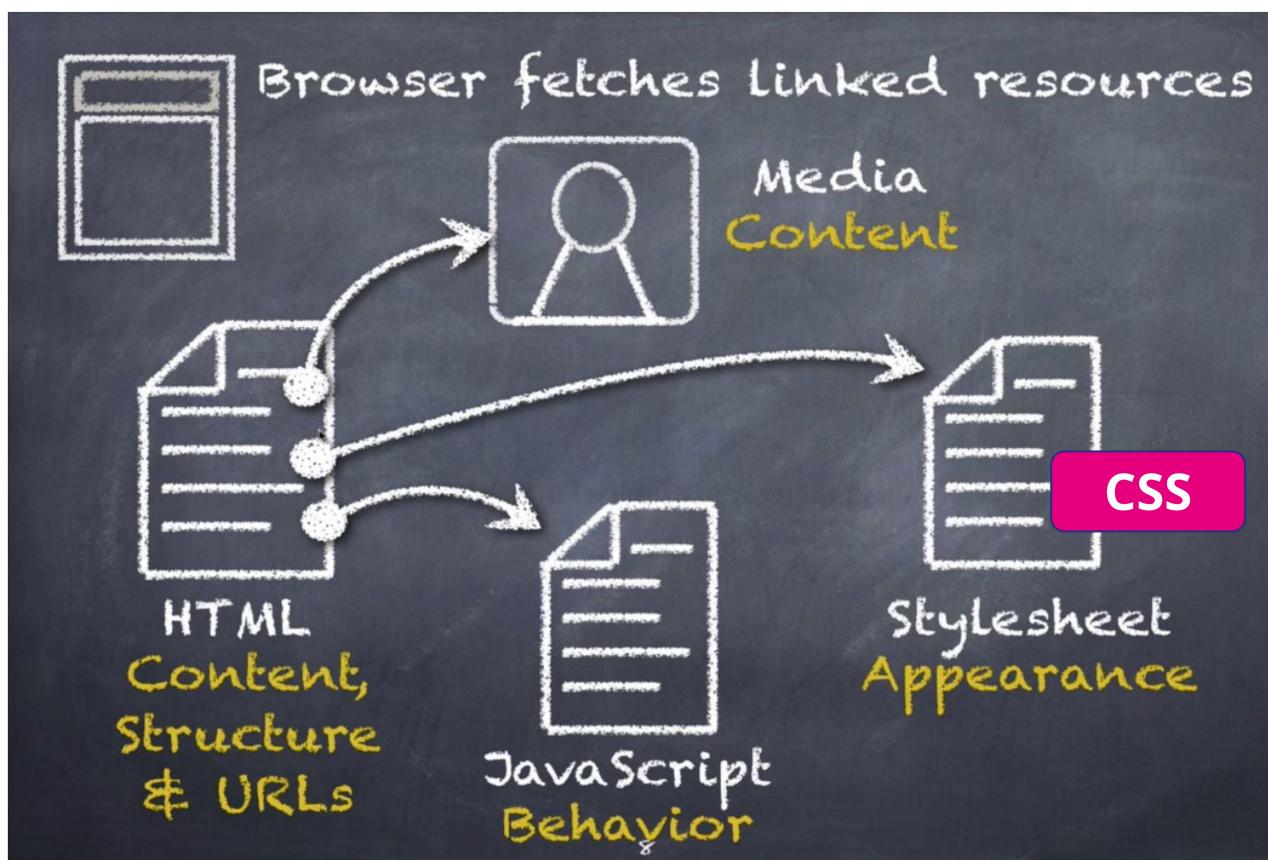


# HTML

## HyperText Markup Language



[https://www.flaticon.com/free-sticker/blue-print\\_11355571](https://www.flaticon.com/free-sticker/blue-print_11355571)

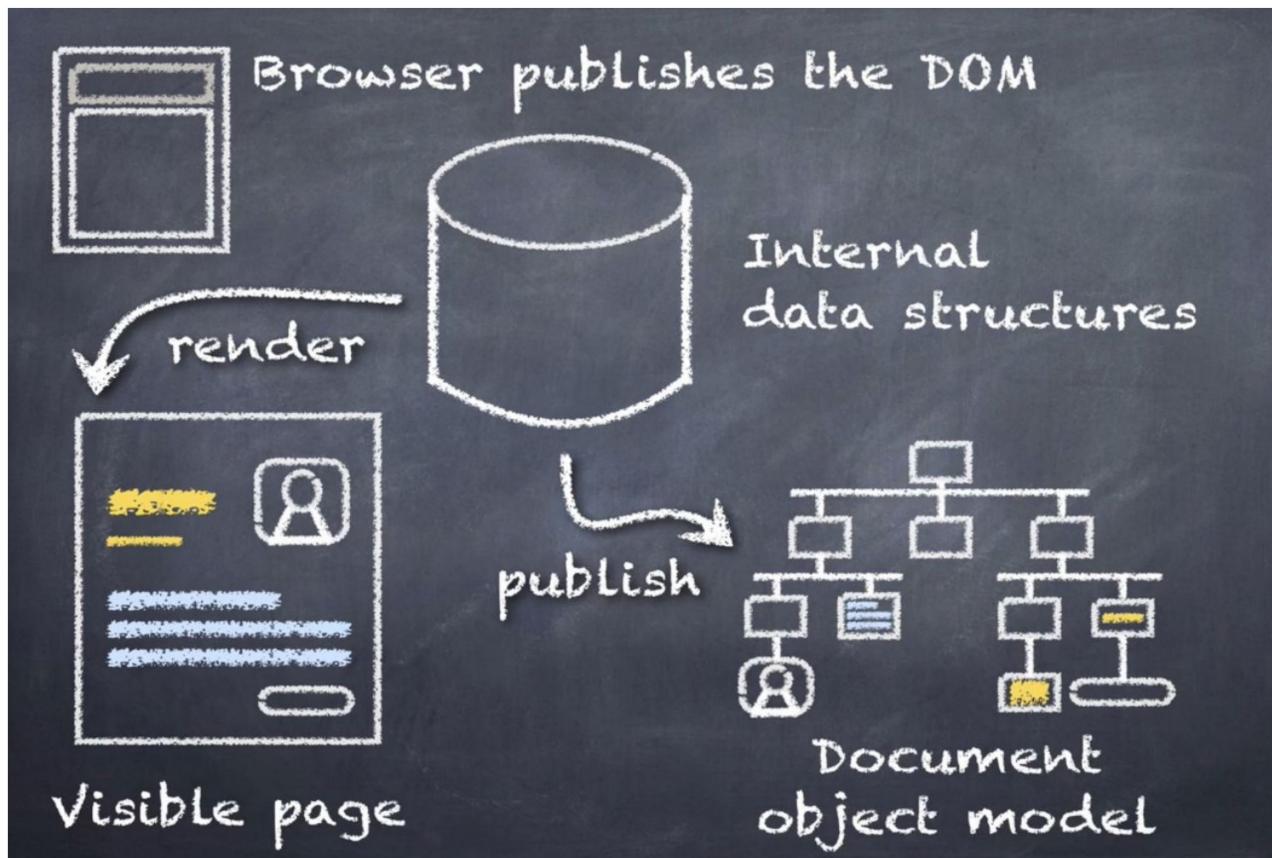


# CSS

## Cascading Style Sheets



[https://www.flaticon.com/free-sticker/web-development\\_11805421](https://www.flaticon.com/free-sticker/web-development_11805421)

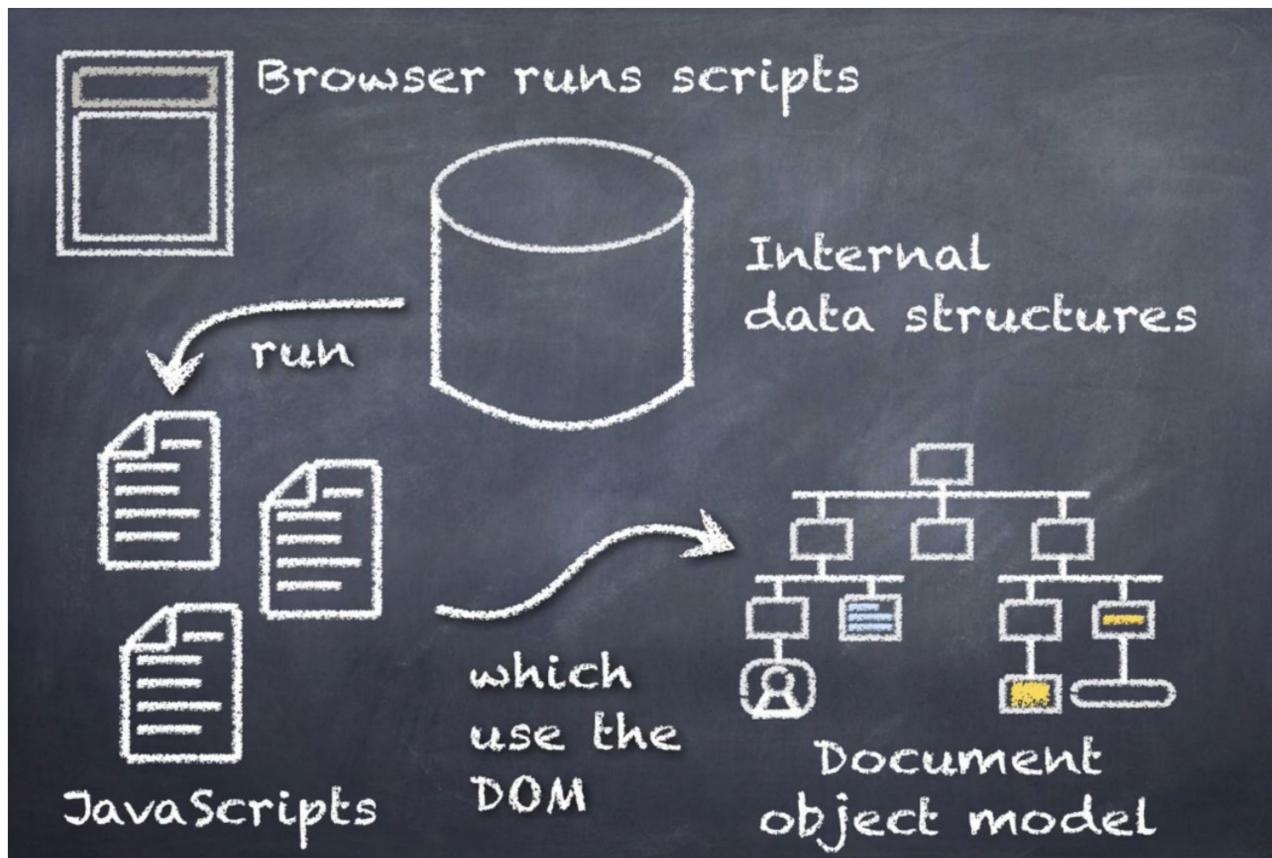


# JavaScript

Interaktivní Weby

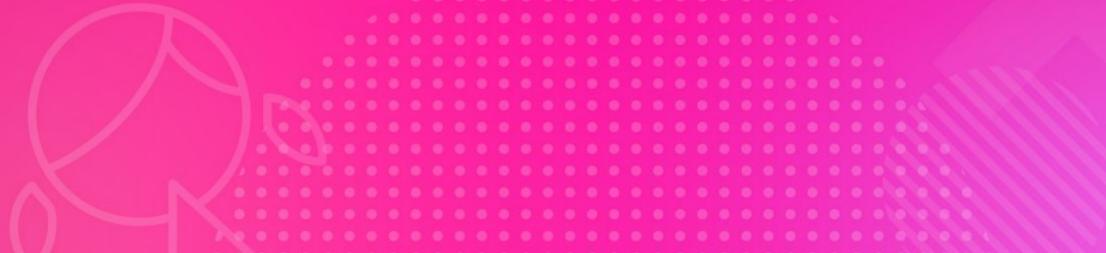


[https://www.flaticon.com/free-sticker/disco\\_11449497](https://www.flaticon.com/free-sticker/disco_11449497)

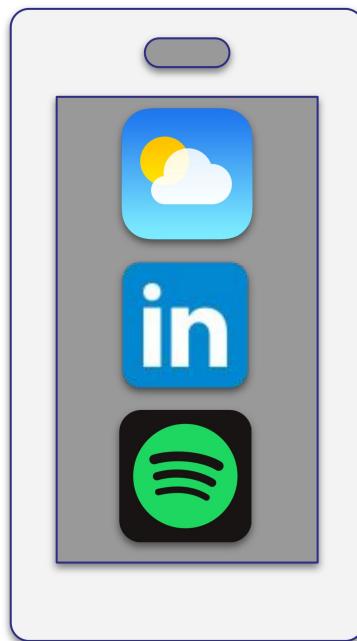




# Co je to Web API?

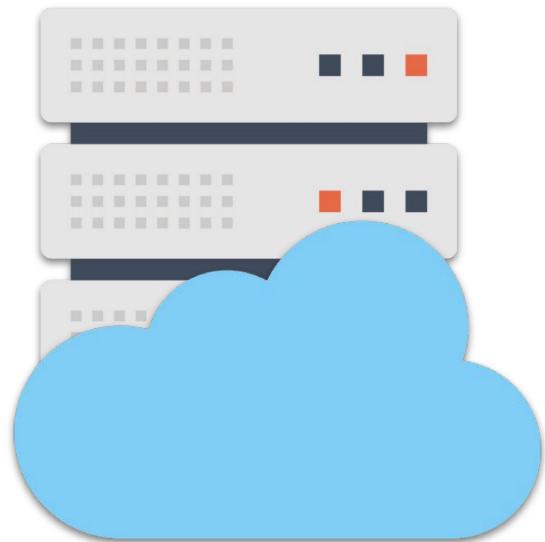


# Klienti a Servery



**Klient**

Jak získat data?



**Server**

## Co je to API?

# Application Programming Interface

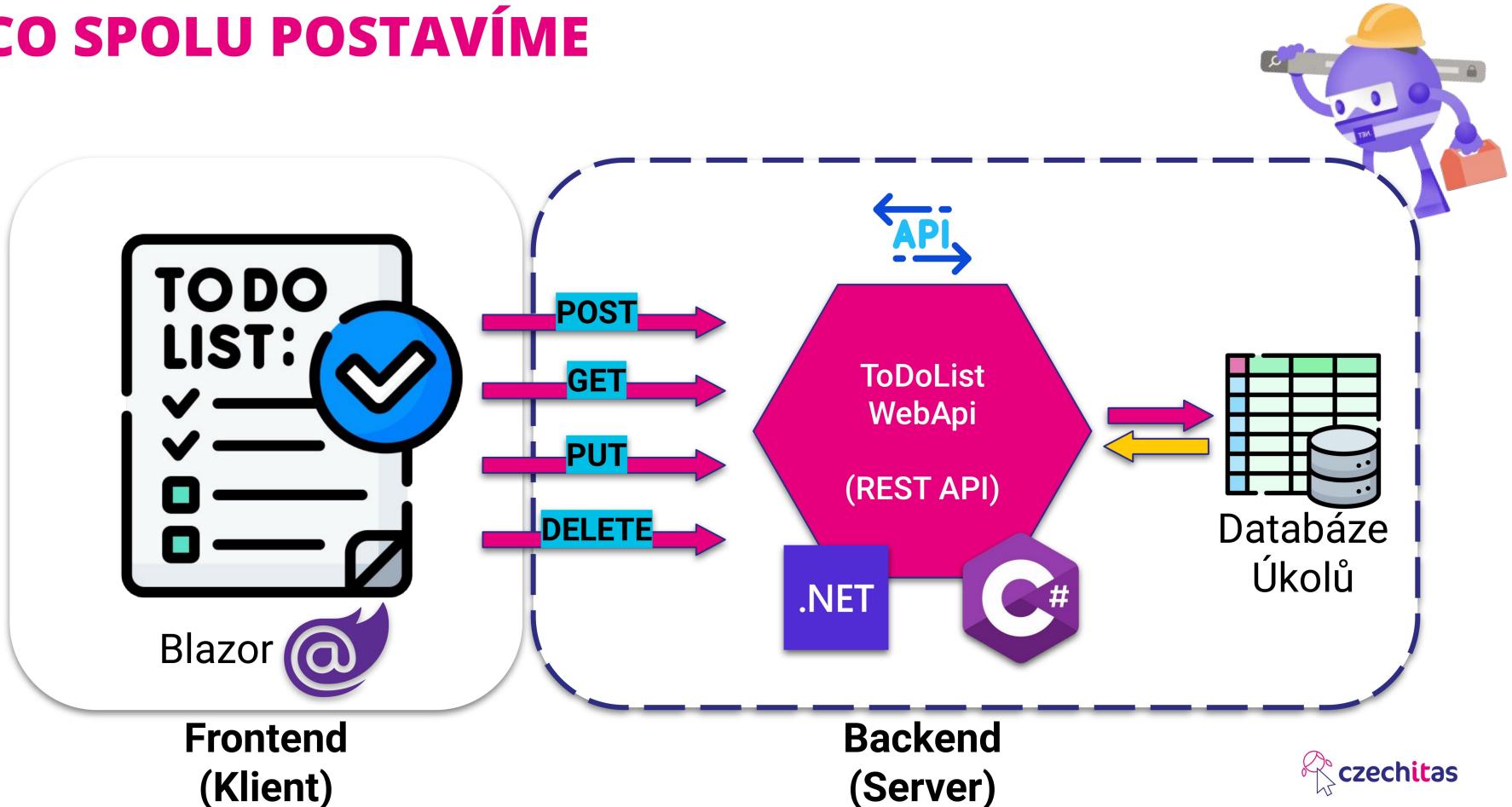


***API pomáhá klientům komunikovat se službami takovým způsobem, aby služba rozuměla a splnila klientem zadaný požadavek (request).***

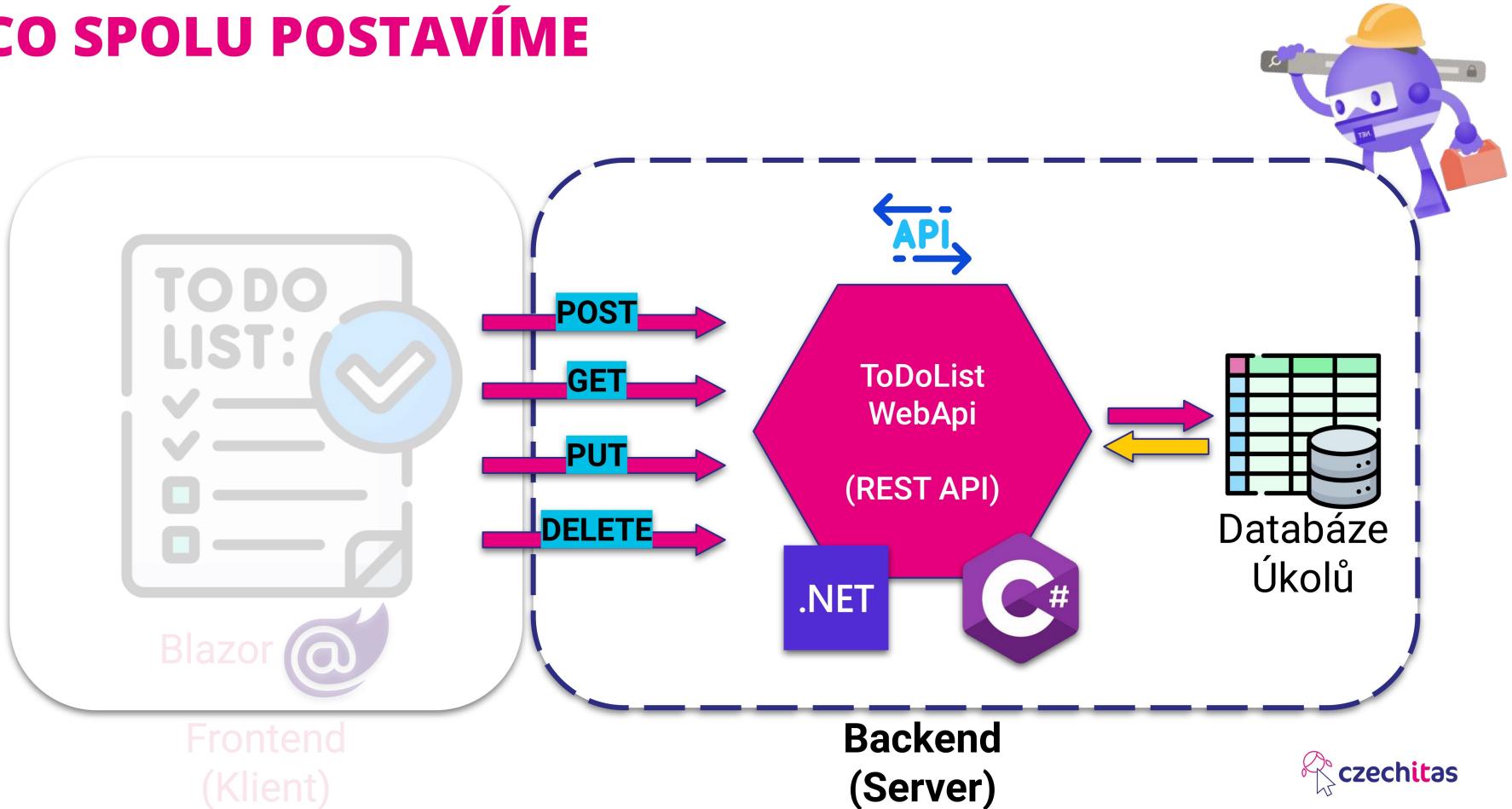
## Co je to Web API?

*Web API je takové API, které je dostupné na počítačové síti (např. internetu). Typicky využívá ke komunikaci protokol HTTP.*

# CO SPOLU POSTAVÍME



# CO SPOLU POSTAVÍME



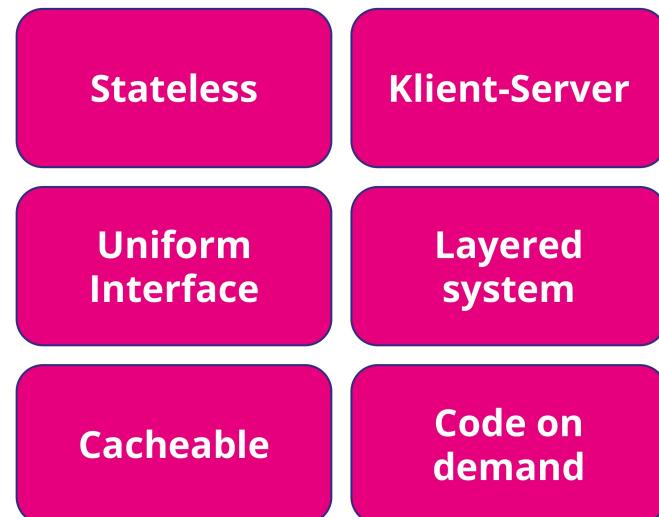
**Pojďme se  
vrhnout do  
programování**



# Co je to REST API?

## Co je to REST?

REpresentational  
State  
Transfer



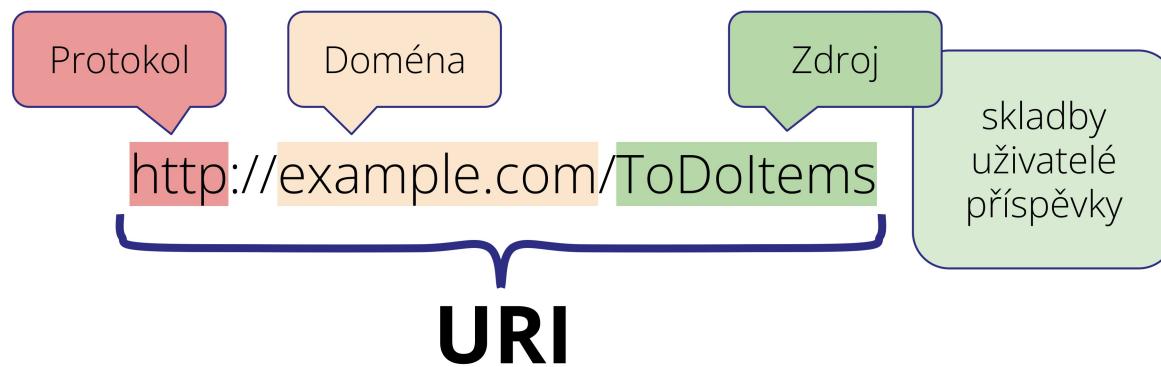
*Soubor principů, které určují podmínky, jak má API fungovat.*

## Co je to REST API?

*REST API nebo RESTFUL API je takové API,  
které dodržuje REST architekturální styl.*

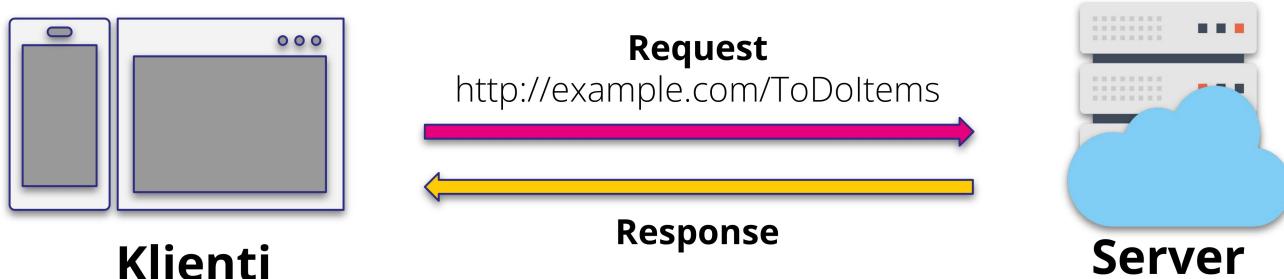
# Jak identifikovat zdroje v REST API?

Zdroj je jakýkoliv objekt, dokument nebo věc, kterou API dokáže přijímat nebo posílat klientům



**Uniform Resource Identifier**

# Jak interagovat s REST API?



CRUD operace

## HTTP Metody

Create	POST	Vytvoří nový zdroj
Read	GET	Získá reprezentaci/stav zdroje
Update	PUT	Modifikace zdroje
Delete	DELETE	Smaže zdroj

# **LEKCE 03**

Interakce s REST API

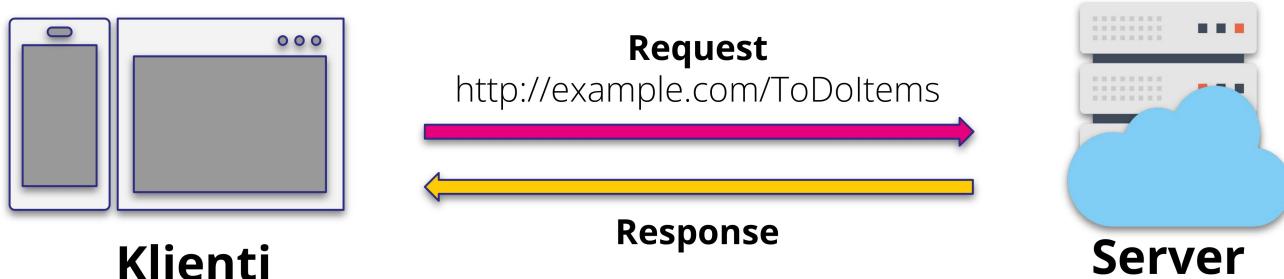
# Lekce 03 - Interakce s REST API

- ToDoList.WebApi
  - **CRUD** metody
  - **Endpoints, Routing**
  - MVC **Controller Endpoint**
- **DTOs**
  - Význam, Definice, Použití
- Testování volání API
  - **REST Client + .http soubory**
- **Implementace** API
  - Nejjednodušší možná

# Práce se šablonou



# Jak interagovat s REST API?



CRUD operace

## HTTP Metody

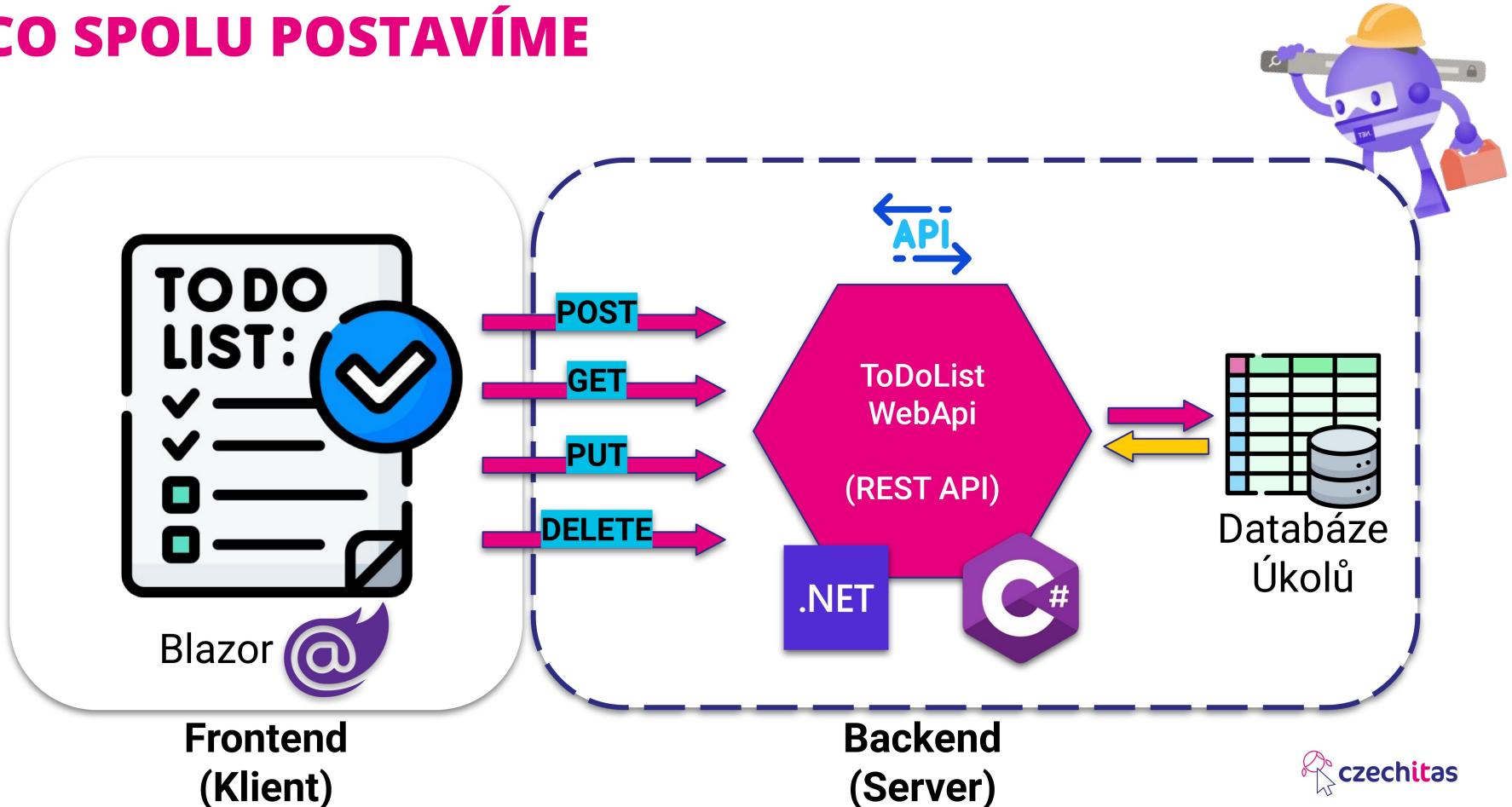
Create	POST	Vytvoří nový zdroj
Read	GET	Získá reprezentaci/stav zdroje
Update	PUT	Modifikace zdroje
Delete	DELETE	Smaže zdroj



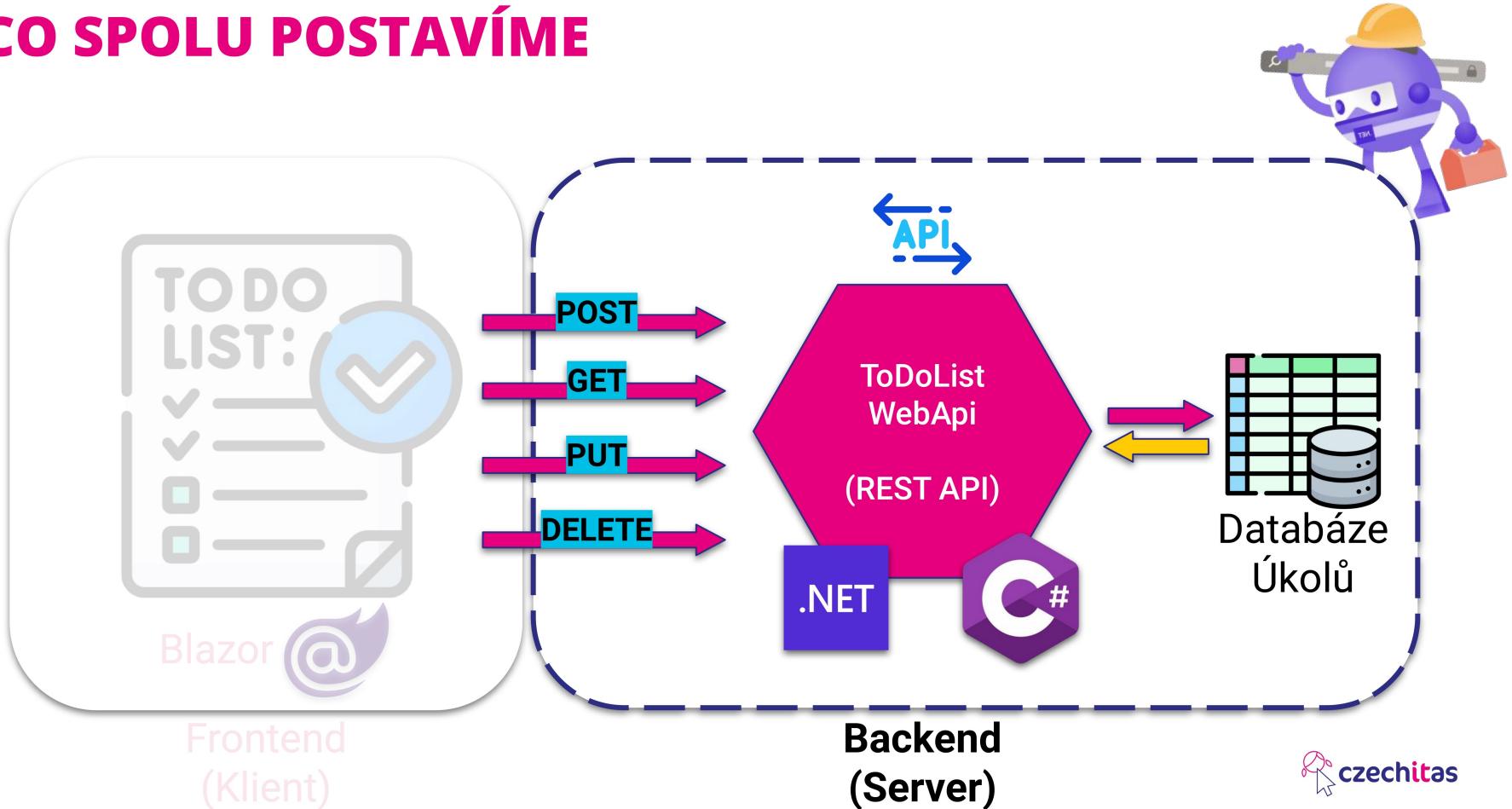
# Pojďme si vyzkoušet

## GET

# CO SPOLU POSTAVÍME



# CO SPOLU POSTAVÍME



# Web API Endpoint

## Co je to Web API Endpoint?

*URL, která reprezentuje konkrétní zdroj (data nebo službu). Každý zdroj má svůj unikátní endpoint, který umožňuje k němu přistupovat.*

## Web API Endpoints: EShop

<https://api.eshop.com/product>

<https://api.eshop.com/users>

<https://api.eshop.com/cart>

<https://api.eshop.com/payment>

## Web API Endpoints: ToDoList

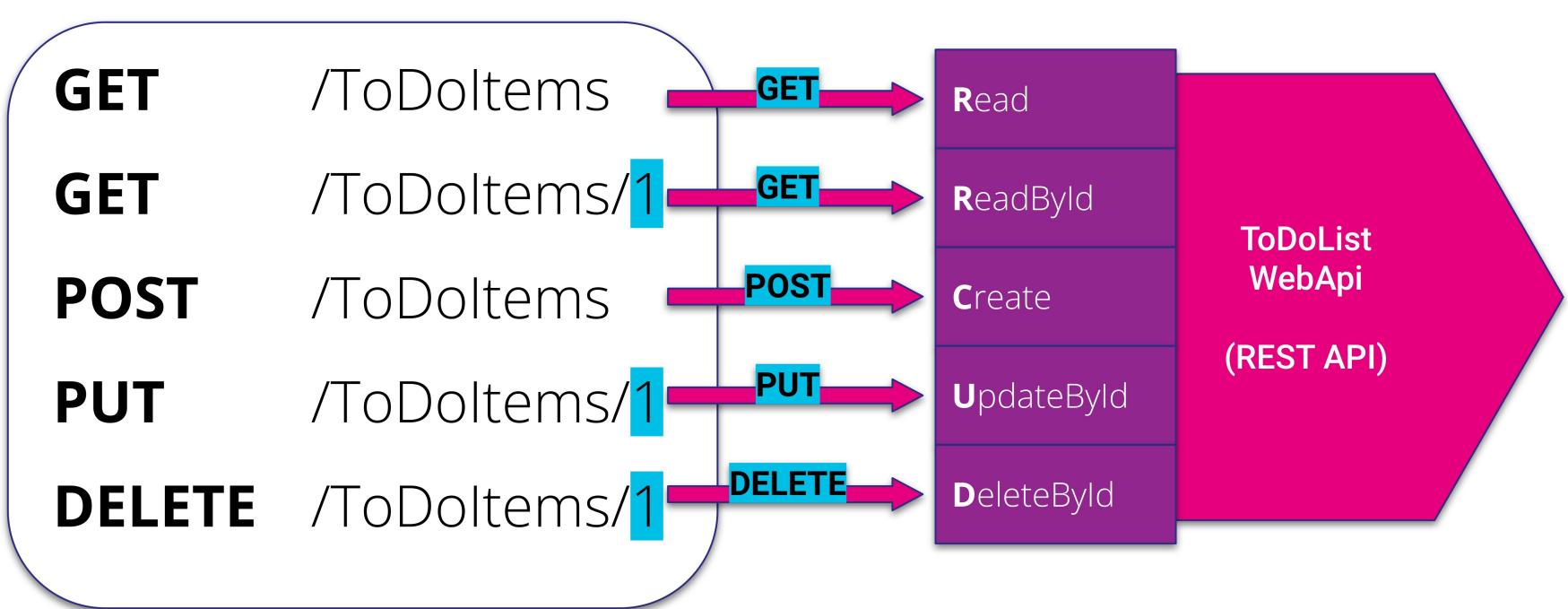
<https://localhost:<port>/ToDoItems>

# REST API Endpoint pro Úkoly

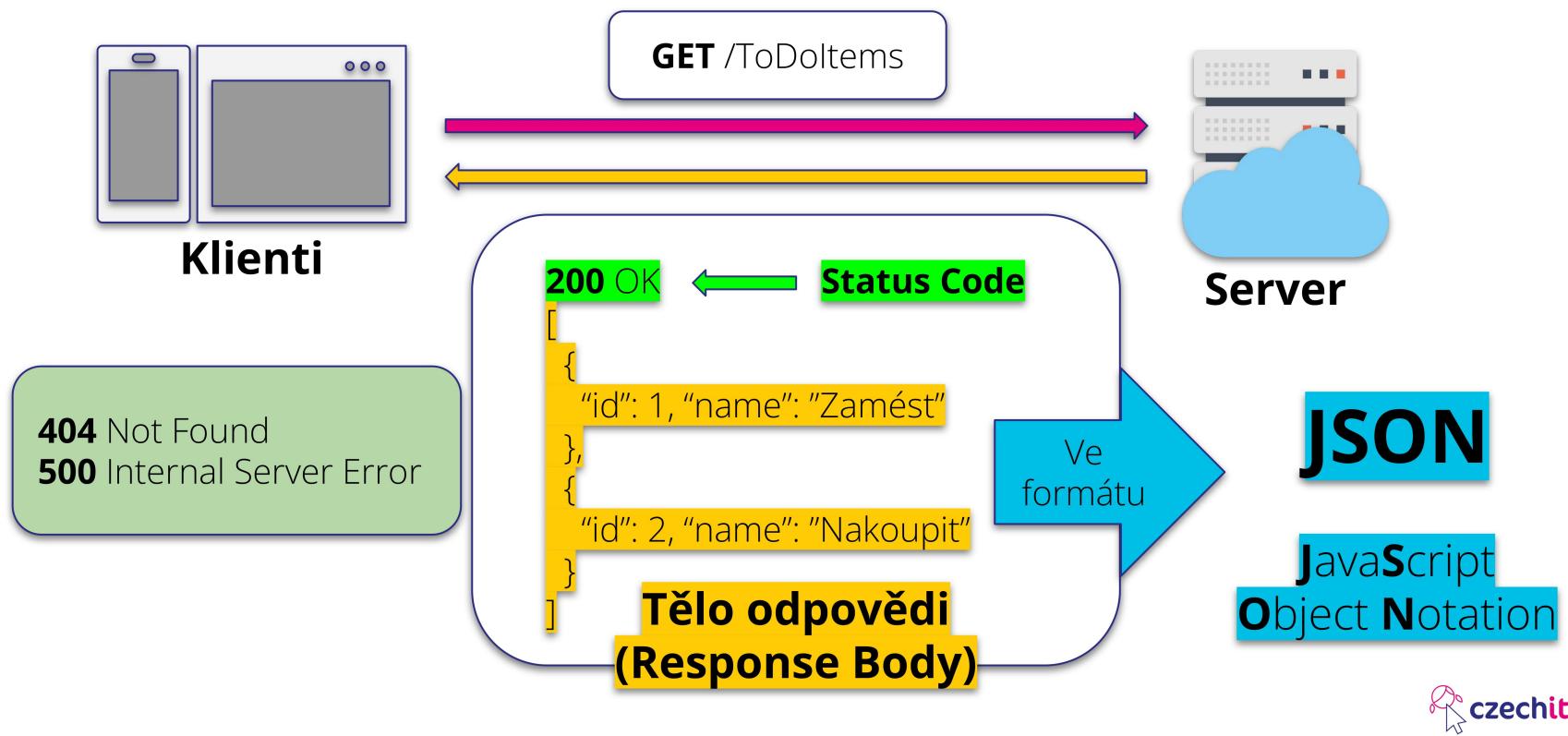
Create	POST	Vytvoří nový zdroj
Read	GET	Získá reprezentaci/stav zdroje
Update	PUT	Modifikace zdroje
Delete	DELETE	Smaže zdroj

**POST** /ToDoItems  
**GET** /ToDoItems  
**GET** /ToDoItems/1  
**PUT** /ToDoItems/1  
**DELETE** /ToDoItems/1

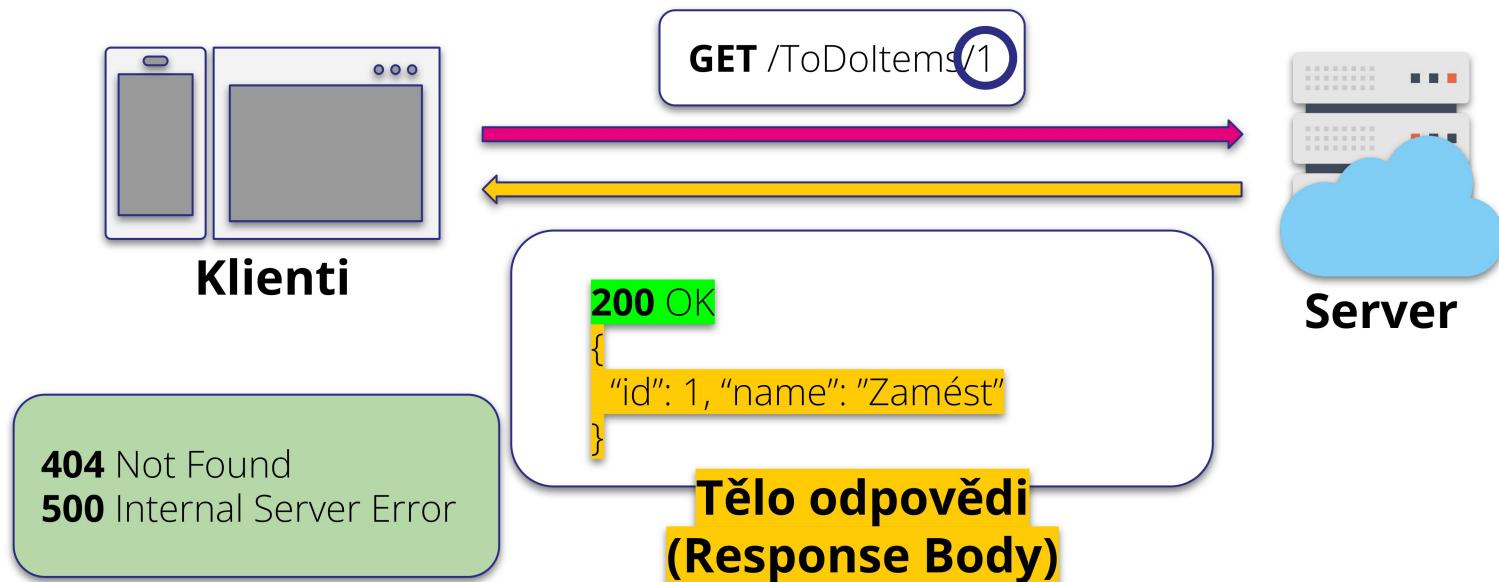
# REST API pro Úkoly



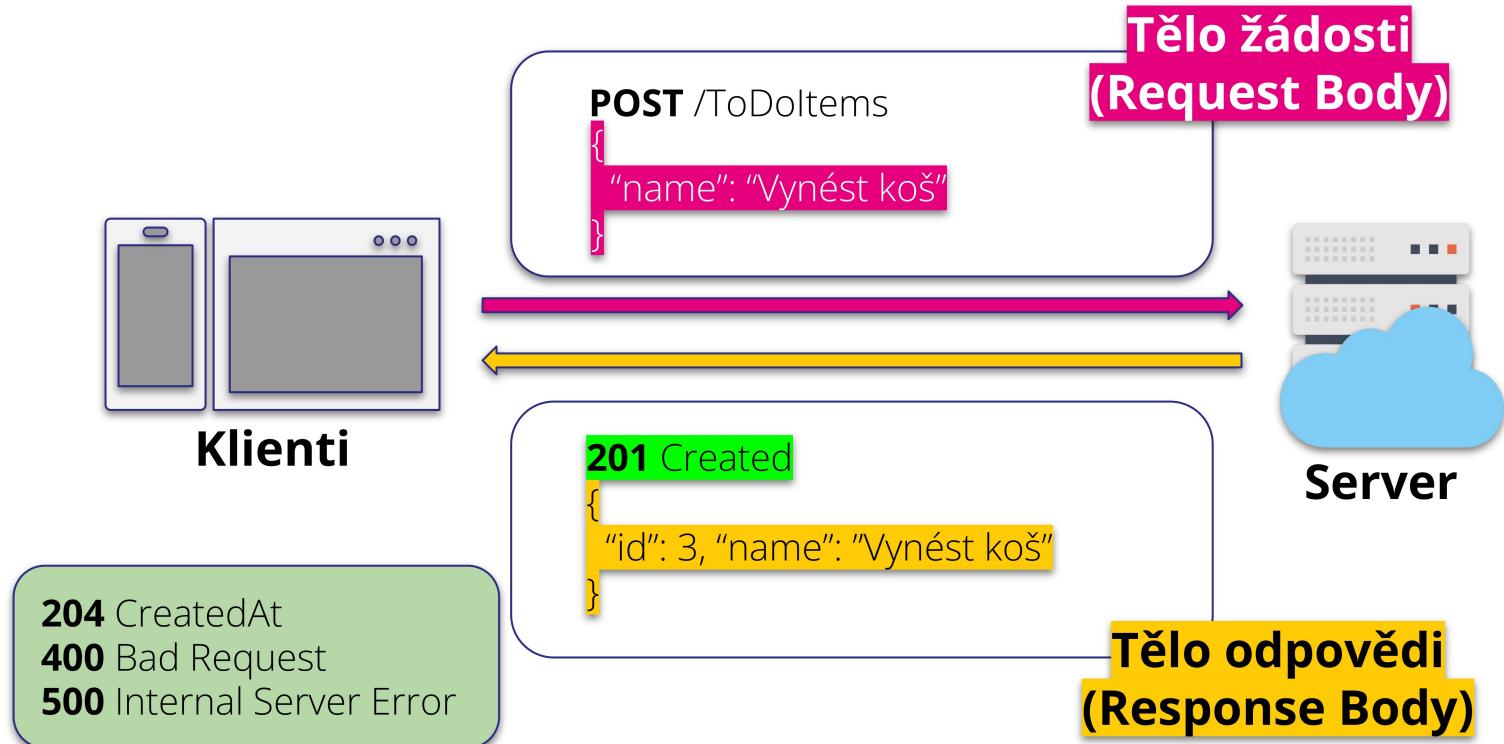
## Získej všechny úkoly - HTTP GET



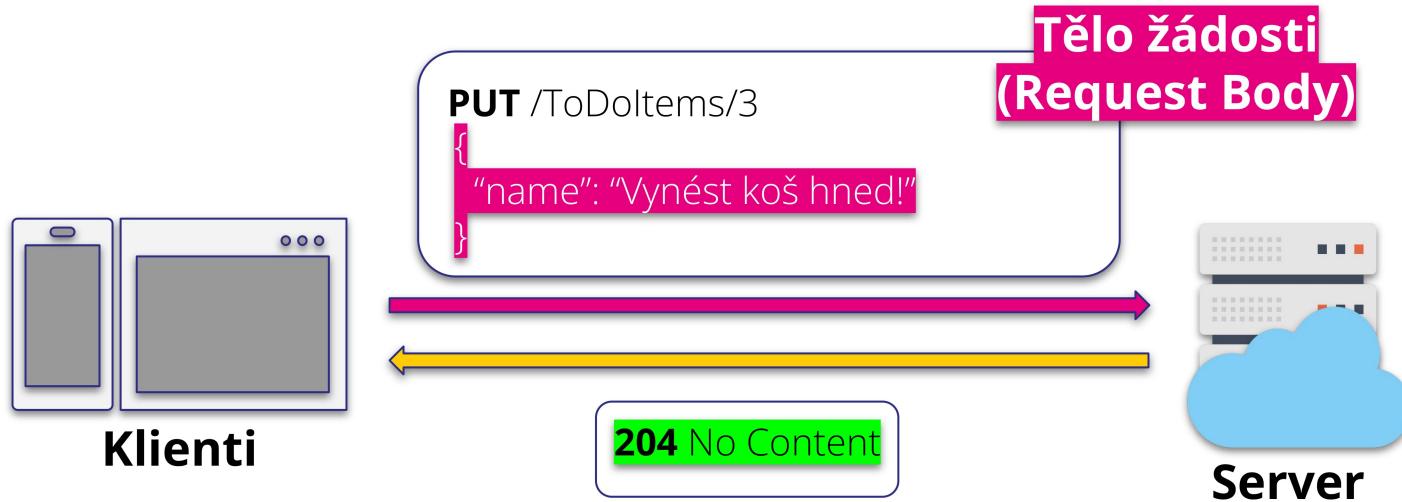
## Získej specifický úkol - HTTP GET



## Vytvoř úkol - HTTP POST



## Modifikuj úkol - HTTP PUT



**404** Not Found  
**400** Bad Request  
**500** Internal Server Error

## Smaž specifický úkol - HTTP DELETE



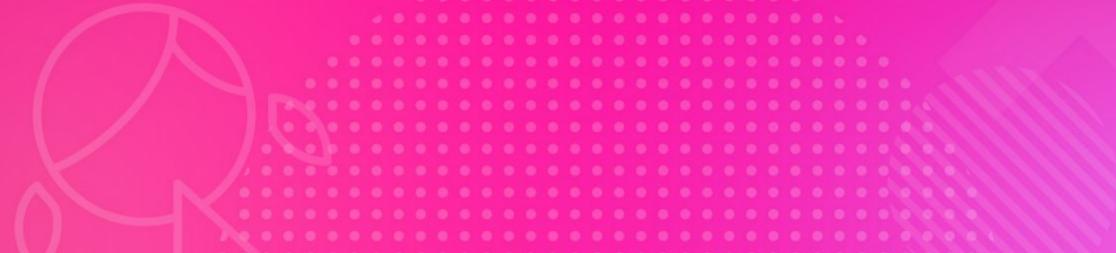
**404** Not Found  
**400** Bad Request  
**500** Internal Server Error

# REST API pro Úkoly

**POST** /ToDoItems  
**GET** /ToDoItems  
**GET** /ToDoItems/1  
**PUT** /ToDoItems/1  
**DELETE** /ToDoItems/1



# Routing



## Co je to Routing?

*Routing rozhoduje, jaká akce nebo část kódu se má spustit na základě cesty zadané v URL a podle použité HTTP metody (GET, POST, PUT, DELETE apod.).*

# **DTO**

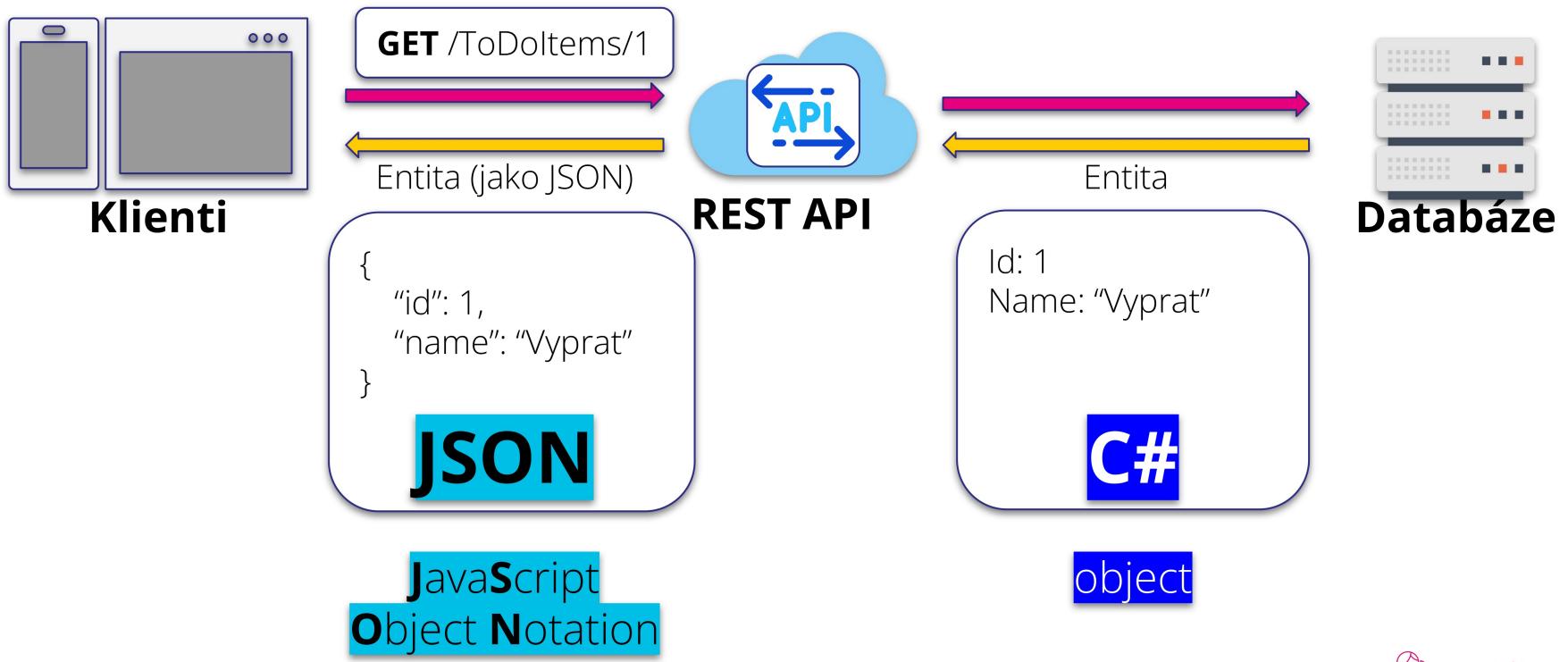
## Data Transfer Object

## Co je to DTO?

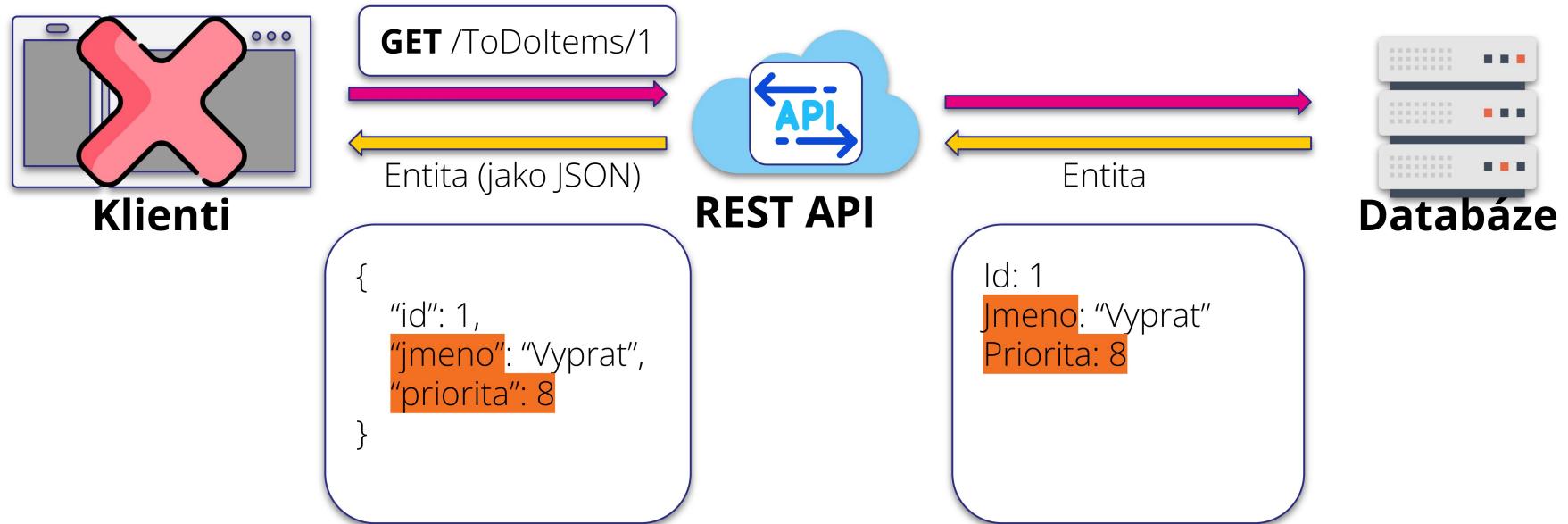
*Data Transfer Object (DTO) je takový objekt, který přenáší data mezi procesy nebo aplikacemi.*

*V kontextu REST API, DTO může být považován jako kontrakt mezi klientem a serverem*

# Proč používat Data Transfer Objects?



# Proč používat Data Transfer Objects?



# Proč používat Data Transfer Objects?

