

ASP.NET Core Web API

Jakub Čermák

jakub@jcermak.cz

 [@cermakj](https://twitter.com/cermakj)

Osnova

1. RESTové služby
2. ASP.NET Core platforma
3. API služba
 1. Controller, návratové hodnoty
4. Formatters, XML, vlastní
5. Novinky v 2.1 (preview)

REST

- Representational State Transfer
- Datově orientovaný – unifikovaný přístup k datům
 - Práce se zdroji (vs. RPC) – CRUD operace
- Jednoduchý, čisté HTTP
 - Snadno škálovatelný – bezestavovost, cachovatelný, distribuovatelný
- JSON / XML
- Adresace pomocí URI
 - Kolekce – <http://něco/images>
 - Element – <http://něco/images/5436>
- Bez metadat

Proč použít?

- Stačí mi CRUD operace
- Chci multiplatformního klienta (JS)
- Chci jednoduché

■ Proč nepoužít?

- Chci neCRUD operace
- Chci více protokolů
- Chci session, transakce a další vymoženosti SOAP

REST operace

- GET (ála SQL SELECT)
 - Seznam URI elementů v kolekci
 - Získání konkrétní položky
- POST (ála SQL INSERT)
 - Vytvoření nové položky
- PUT (ála SQL UPDATE / UPSERT)
 - Update položky pokud existuje, jinak vytvoření.
 - Update kolekce nějakou jinou.
- DELETE (ála SQL DELETE)
 - Smazání kolekce
 - Smazání položky

REST návratové hodnoty

- 200 OK - vrací obsah
 - GET, POST (nově přidaná položka včetně přiděleného ID)
- 201 Created; 202 Accepted
 - PUT
- 204 No content
 - DELETE

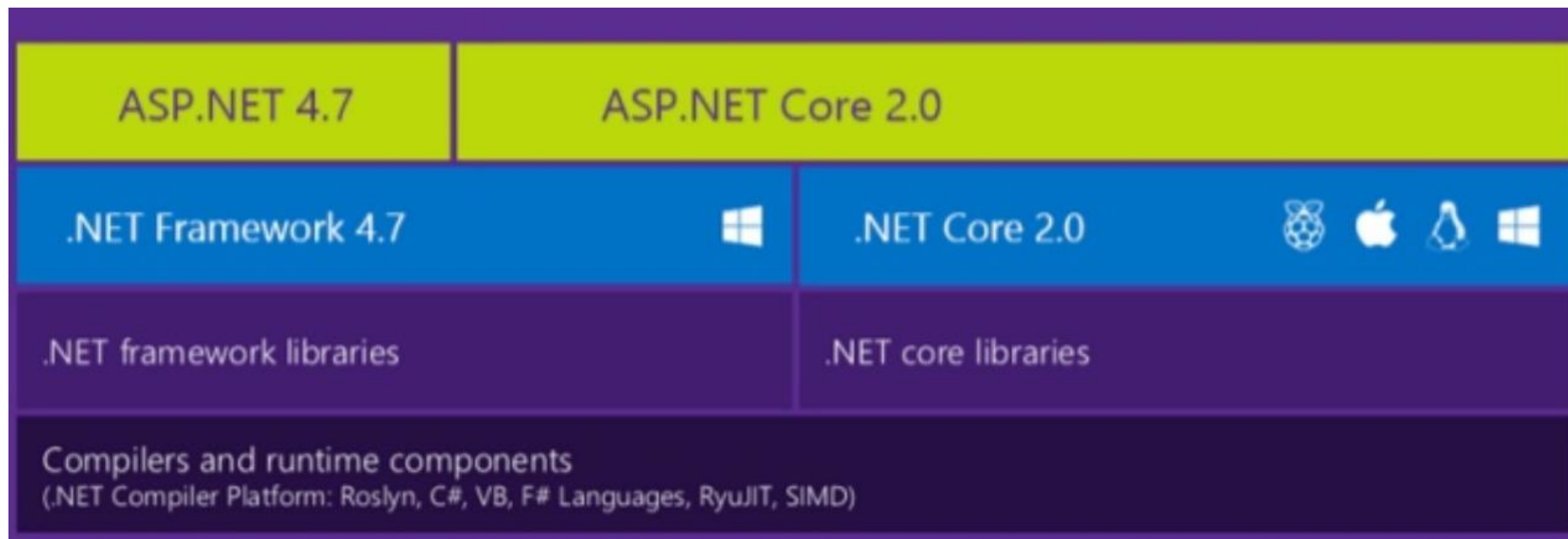
- 400 Bad Request
- 409 Conflict

HTTP 418: I'm a teapot



ASP.NET Core

- Přepsání celého .NETu a ASP.NETu
- Crossplatform, modular, extensible, DI container, high performance,
- Kestrel



Startup.cs

- Úvodní třída aplikace
- Konfigurace knihoven
- Instalace middlewarů
- Registrace aplikačních služeb - DI

ASP.NET MVC Core

- Port původního MVC nad Core
- Funkcionalita navíc
 - Integrace Web API

ASP.NET MVC Web API

- Součást MVC
- => píšeme controllery a modely
- Pro každý endpoint a HTTP metodu -> separátní metoda
 - Atributy [HttpGet], [HttpPost],
- Adresa endpointu – bazová route pro controller + pro akci

Návratové hodnoty

- 1) void 😊
- 2) konkrétní data
- 3) IActionResult
 - Zabalená odpověď
 - HTTP status code + hlavičky + data



Validace

- Validace správnosti dat na modelu
- 1) atributy data annotationn
 - Možnost vlastních
- 2) IValidatableObject



Doporučení pro controllery

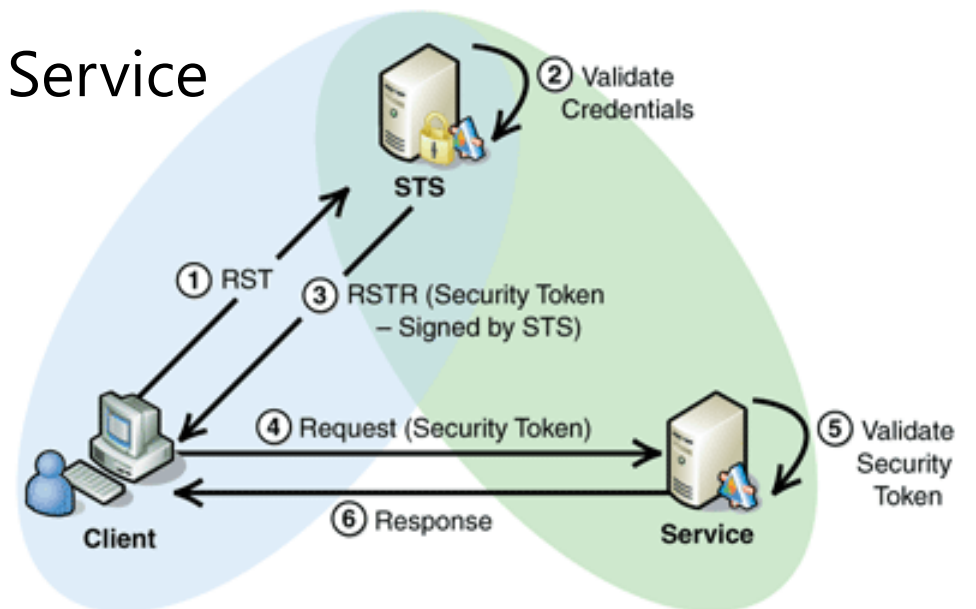
- Tenké controllery
- Co nejmenší modely
 - Overposting
- Dodržovat HTTP metody
 - Bezpečnost
- Nepřenášet citlivé údaje v URI
 - Logy
- Využívat DI
 - Když už ji konečně máme 😊

Servisně orientované API

- Smečka POST akcí, které znázorňují jednotlivé metody

Autentizace + autorizace

- Ověření identity uživatele
- Pomocí ASP.NET Core Identity – mnoho možností
 - Pro API:
 - Převzít webovou (pro AJAX)
 - Oauth / OpenID connect
 - ◆ Vyžaduje identity server / Security Token Service
 - Basic auth -> pozor na bezpečnost



Dokumentace a metadata

- Swagger / Open API
- Metadata o REST službě – ve formátu JSON, YAML
 - => dokumentace strojová
 - => generování proxy klientů (autoREST, <http://editor.swagger.io/>)
- UI
 - => lidská dokumentace
 - => testovací rozhraní
- Nuget Swashbuckle.AspNetCore

Autorest

- Npm install -g autorest
- Autorest -reset
- autorest --namespace=MyCoolClient --input-file=http://localhost:54624/swagger/v1/swagger.json --output-folder=MyCoolClient -csharp
- Nuget Microsoft.Rest.ClientRuntime



Podpora formátů

- Řídí se dle hlavičky Accept
 - Lze i podle URL
- JSON
 - V základu
- XML
 - Microsoft.AspNetCore.Mvc.Formatters.Xml nuget
- Vlastní
 - Input / output formatters



Novinky v 2.1 ohledně API

- IActionResult<T>
 - Náhrada za ProducesResponseType
 - Přetížený implicitní konverzní operátor z T
- [FromBody] je nepovinný
- Možnost validačních atributů přímo na parametru akce
- Podpora [RFC 7808 – Problem Details for HTTP APIs](#)
 - Standardizované podání validačních chyb namísto BadRequest

Dotazy

Jakub Čermák

jakub@jcermak.cz

 [@cermakj](https://twitter.com/cermakj)