MINIMAL API – C#6, C#7

OBSAH

[Rozprava 2](#_Toc129594413)

[API dnes 2](#_Toc129594414)

[Zvyky 2](#_Toc129594415)

[Jak by to melo byt 2](#_Toc129594416)

[Specification first 2](#_Toc129594417)

[Výhody 2](#_Toc129594418)

[Open API Standardizace (Swagger) 2](#_Toc129594419)

[Vytvoření specifikace 2](#_Toc129594420)

[Specifikace ve Wordu 2](#_Toc129594421)

[Stoplight studio 3](#_Toc129594422)

[DEMO 3](#_Toc129594423)

[Specifikace - Word 3](#_Toc129594424)

[GITHUB 3](#_Toc129594425)

[Stoplight studio (STLS) 3](#_Toc129594426)

[Tvorba API 3](#_Toc129594427)

[POSTMAN 3](#_Toc129594428)

[Import 4](#_Toc129594429)

[REDOC 4](#_Toc129594430)

[Mock server 4](#_Toc129594431)

[PRISM 4](#_Toc129594432)

[Stopligh studio 4](#_Toc129594433)

# Rozprava

Popis tech dvou dilu, o cem to dnes je

* O cem byl prvni, specifikace + api, vsechny nastroje, c# 6,
* O cem byl druhy – api mini, grupovani, c# 7

# API dnes

Zvyky

* Postavit API
* Z API vygenerovat dokumentaci

Jak by to melo byt

* API si nejdříve specifikujeme
* Využijeme standardizované specifikace k vygenerování MOCKů
* Specifikace k vygenerování dokumentace
* Validace samotného API
* Na základě specifikace se provádí implementace

# Specification first

Myšlenkový postup, shoda sw architektu, pristup, kdy je nahlizeno na API jako samostatný produkt, samostatne rozhrani pro vyvojare, rozhradni pro vyvojare navrzene, resime potreby vyvojari

## Výhody

* Jediný bod pravdy (pokryte standardem), veskera pravidla, endpointy
* Rychlejší něž vývoj
* Díky specifikaci lze paralelně vyvíjet BE i FE
* Každý specifikaci rozumí
* Lze ji použít pro automatizaci, mocků, dokumentaci

## Open API Standardizace (Swagger)

* Swagger do verze 2, od verze 3.0 OpenAPI (unor 2021)
* Podpora u většiny moderních nástrojů
* Popis API pomocí yaml nebo json, moderni nastroje preferuji yaml

## Vytvoření specifikace

### Specifikace ve Wordu

1. Ideální je začít nějakou word šablonou
2. naprototypovat si budoucí API specifikace
3. šablona by měla co nejvíce reflektovat OpenApi specifikace a co nejvíce se jí přiblížit
4. být stále co nejvíce čitelná pro uživatele
5. jakmile bude hotová, vygenerovat specifikaci k příslušném nástroji

#### Výhody Wordu

* Mnohem rychlejší, než rovnou psát v nějakém komplexním nástroji
* Pro porady nebo meetingy se zákazníky je to srozumitelnější

### Stoplight studio

Jedná se o nejkomplexnějsí a nejpraktičtějsí nástroj pro tvorbu OpenApi specifikace

* Ondemand aplikace
* online služna
* propojeni s githubem, kde mame ulozenou samotnou specifikaci

# DEMO

## Specifikace - Word

Krok č.1 – co nejvíc reflektovat OpenApi specifikaci

* https://swagger.io/specification/
* Začíná úvodními společnými vlastnostmi
* Podporovaný formát – př. appliacation.json
* Deklarování stavových kodu
* Formátování – pravidla, pokrýt vešker hodnoty standardy
* Path – endpointy – popis
* Odkazování – ctrl+k
* + povinne, o optional

## GITHUB

* Před prací se stoplightstudiem si založme GtHub repo
* STLS vytvoří adresář **reference**
* Sem se budou ukladat yaml api file
* StopLight studio se umí na github napojit

## Stoplight studio (STLS)

* Napojení na repositář
* Ukazka pripravene specifikace
* Prepnout na code – ukazaka yaml, specfikace API

### Tvorba API

* Modely
  + - komponent schema z wordu - popis
* Responses
  + - Na zaklade modelu lze definovat respo, + code, reference na model
* Paths
  + - url, metoda, parametry, komentare, nazev operace, navratovy typ – reference, 400,404,500 – nastaveni referenci
* Commit na git
  + - Msg, push, adresar reference, yaml, raw – surova data

## POSTMAN

* Díky yaml můžeme jednoduše vygenerovat kolekci v postmanu

### Import

* Import
* Link
* Název yaml kolekce – url na raw na githubu
* Check – Generate collection – import
* Automaticky se vytvori API

## REDOC

* Slouží pro vygenerování dokumentace, nejrozšířenější nástroj pro dokumentaci
* HTML soubor (default) – prida se JavaScript s linkem na REDOC + odkaz na GitHUB s yaml filem
* Spustit v browser: je vygenerovaná dokumentace na základě specifikace ze StopL st.

## Mock server

### PRISM

* Nástroj pro vytvoření lokálního api-serveru na základě opět yaml dokumentace
* Spustí na url server s naším API včetně dat
* Za pomocí Postmana lze tak ihned provolávat API a FE může vyvíjet nezávisle na BE
* Podpora jen pro linux, na win jsem nenasel

### Stopligh studio

* STLS nutné zapnout v konfiguraci podporu test a mock serveru
* Settings / Docs Settings / Enable Try It
* Settings / Docs Settings / Show Mock Servers