

Univerzita Jana Evangelisty Purkyně v Ústí nad Labem (UJEP)

Generování REST API specifikace pomocí LLM

Seminární práce (KI/SWI)

Ondřej Švorc (F23209)

17. 5. 2025

Úvod

Tato práce zkoumá, zdali je LLM o4-mini-high¹ (dále jen LLM) schopen extrahovat ze vstupu (přírozený jazyk) strukturu REST API (endpointy, metody, parametry) a převést ji do OpenAPI dokumentace (JSON) dodržující OpenAPI 3.0 specifikaci². Dále se zaměřuje na schopnost LLM generovat návrhy na zlepšení OpenAPI dokumentace a tranzitivně i výsledného REST API. Nakonec se pak zaměřuje na jeho dovednost generovat REST API v C# .NET 8 ASP.NET Web API (Minimal APIs) na základě dané dokumentace pro jeden konkrétní business scénář (správa knih).

V běžné praxi je OpenAPI dokumentace často implicitně generována zpětně z existujícího REST API kódu, typicky pomocí nástrojů jako Swagger³. Slouží jednak jako dokumentační rozhraní, ale také pro automatické generování klientského kódu pro komunikaci s daným REST API (Swagger Codegen⁴). Cílem je zjistit, zda opačný přístup, tedy generování samotného návrhu a implementace REST API z OpenAPI dokumentace vygenerované pomocí LLM, může být z pohledu času rychlejší na realizaci.

Téma práce považuji za vysoce aktuální. Používání velkých jazykových modelů razantně mění způsob, jakým přemýšlíme o softwarovém návrhu – včetně návrhu REST API. Co bývalo introspektivním monologem vývojáře, případně kolektivním rozhodováním v týmu, se častokrát mění v dialog s LLM. Domnívám se, že role vývojáře jako jednotlivce se bude postupně posouvat od samotného psaní kódu k formulaci jeho záměru, a to vše pro následnou interpretaci LLM. Vývojář nebude primárně autorem konkrétních implementací, ale průvodcem v procesu iterativní elicitace s LLM, jejímž cílem bude výstup, se kterým bude subjektivně spokojen.

¹ OPENAI. Introducing OpenAI o3 and o4-mini [online]. 16. dubna 2025 [cit. 12. května 2025]. Dostupné z: <https://openai.com/index/introducing-o3-and-o4-mini/>

² OpenAPI Initiative. OpenAPI Specification v3.0.0. [online]. 2020 [cit. 2025-05-15]. Dostupné z: <https://spec.openapis.org/oas/v3.0.0>

³ SmartBear Software. Swagger – Open Source API Tool. [online]. [cit. 2025-05-15]. Dostupné z: <https://swagger.io/>

⁴ SMARTBEAR SOFTWARE. Swagger Codegen [online]. 2025 [cit. 2025-05-17]. Dostupné z: <https://swagger.io/tools/swagger-codegen/>

Výzkumná rešerše

Zkoumaná oblast tohoto výzkumu se ukazuje být málo probádaná. Jedním z důvodů může být vysoce specifická povaha výzkumu. Dále, kvůli zvolení stále relativně nedávno vydaného LLM (31. ledna 2025⁵), se mi nepodařilo najít relevantní zdroje, které by měly stejnou nebo alespoň podobnou metodiku výzkumu, a to ani s jinými, staršími LLM. Naopak jsem narazil na několik studií zabývajících se opačným přístupem, tedy vytvářením OpenAPI dokumentace z již existujícího REST API kódu.

Jedním z příkladů takového přístupu je švédská bakalářská práce z roku 2024 od autorů Esbjörn Lyster a Golawski James Taylor. Autoři v ní využili LLM GPT-4 k automatickému generování OpenAPI dokumentace z již existujících REST API implementací. Generované specifikace následně porovnávali s ručně vytvořenými verzemi od vývojářů a hodnotili tři kritéria, a to jak je výstup přesný, úplný a konzistentní s požadavky.⁶

Další zajímavou prací je čínská studie z roku 2025 od autorů Sida Deng, Rubing Huang a dalších. Autoři v ní představují nástroj LRASGen, který využívá mimo LLM GPT-4 také LLM DeepSeek v3⁷ k automatickému generování OpenAPI dokumentace přímo ze zdrojového kódu existujících REST API.⁸

⁵ OpenAI. OpenAI O3 mini. In: *OpenAI* [online]. 2024 [cit. 2025-05-15]. Dostupné z: <https://openai.com/index/openai-o3-mini/>

⁶ Kamali, M. Evaluating GPT-4 for OpenAPI Generation and Conversion. Bachelor's Thesis. Uppsala University, 2024. Dostupné z: <https://www.diva-portal.org/smash/get/diva2:1877570/FULLTEXT01.pdf>

⁷ DEEPSEEK AI. *DeepSeek V3* [online]. [cit. 2025-05-17]. Dostupné z: <https://deepseekv3.org/>

⁸ DENG, Sida; HUANG, Rubing; ZHANG, Man; CUI, Chenhui; TOWEY, Dave; WANG, Rongcun. *LRASGen: LLM-based RESTful API Specification Generation* [online]. 2025 [cit. 2025-05-17]. Dostupné z: <https://arxiv.org/abs/2504.16833>

Výzkumné otázky

1. Jak lze automaticky vytvářet OpenAPI dokumentaci ve formátu JSON k REST API, který je popsán přirozeným jazykem, pomocí LLM o4-mini-high?
 - Validita výstupu
 - výstupem je JSON (ano/ne)
 - JSON je well-formed⁹ (ano/ne)
 - JSON odpovídá OpenAPI 3.0 specifikaci (ano/ne)
 - Počet endpointů se správnými parametry a návratovými typy / celkový počet očekávaných
 - Subjektivní skóre (0-5, 0 = nejhorší, 5 = nejlepší) hodnotící to, jak dobře výstup odpovídá zadání
2. Jak lze generovat návrhy na zlepšení OpenAPI dokumentace ve formátu JSON pomocí LLM o4-mini-high?
 - Výstup zůstává validní podle OpenAPI 3.0 specifikace (ano/ne)
 - Počet nově doplněných prvků (např. summary, examples, constraints)
 - Subjektivní skóre (0-5, 0 = nejhorší, 5 = nejlepší) hodnotící, jak je výsledek přínosný
3. Jak lze generovat REST API v C# .NET 8 ASP.NET Web API (Minimal APIs) na základě OpenAPI dokumentace ve formátu JSON pomocí LLM o4-mini-high?
 - Validita výstupu
 - Výstupem je C# kód
 - Kód je po vložení do Program.cs¹⁰ kompilovatelný
 - Počet endpointů se správnými parametry a návratovými typy / celkový počet očekávaných
 - Subjektivní skóre (0-5, 0 = nejhorší, 5 = nejlepší) hodnotící čitelnost a využitelnost kódu jako základní implementace

⁹ SyncRO Soft. JSON well-formedness. In: *Oxygen XML Editor User Guide* [online]. Version 27.1. [cit. 2025-05-15]. Dostupné z: <https://www.oxygenxml.com/doc/versions/27.1/ug-json/topics/json-well-formedness.html>

¹⁰ *Fundamentals*. ASP.NET Core [online]. [cit. 17. 5. 2025]. Dostupné z: <https://learn.microsoft.com/en-us/aspnet/core/fundamentals/?view=aspnetcore-9.0&tabs=windows>

Návrh experimentů

Metodologie této práce spočívá v empirickém testování schopností LLM ve třech oblastech odpovídajících výzkumným otázkám. LLM, konkrétně tedy o4-mini-high, je dotazován výhradně v angličtině. Pro každý úkol jsou vytvořeny samostatné promptovací šablony. Šablony jsou navrženy tak, aby jasně vymezily očekávaný výstup. Dále jsou strukturovány dle stejného vzoru, a proto každá z nich obsahuje úvod definující roli LLM s kýženým výstupním formátem a následně požadavky na tento výstup. Model nebyl žádným způsobem upravován ani laděn. Pro validaci JSON výstupu je používán Swagger Editor¹¹.

```
1 You are an assistant that generates OpenAPI 3.0 documentation in JSON format.
2
3 The API should allow users to manage a collection of [EntityName].
4
5 Each [EntityName] has the following properties:
6 - [field1Name] ([type], [required/optional])
7 - [field2Name] ([type], [required/optional])
8 - [field3Name] ([type], [required/optional])
9 - ...
10
11 The system also includes related entities:
12 - [RelatedEntity1Name]: [field1] ([type]), [field2] ([type, optional])
13 - [RelatedEntity2Name]: [field1] ([type])
14
15 Relationships:
16 - A [EntityName] can have multiple [RelatedEntity1Name] and [RelatedEntity2Name]
17 - [any other relationship, e.g. "Authors can be shared across books"]
18
19 The API should support the following operations:
20 - Create new [EntityName]
21 - View all [EntityName]
22 - View specific [EntityName] by ID
23 - Update [EntityName] by ID
24 - Delete [EntityName] by ID
25 - List all [RelatedEntity1Name]
26 - Filter [EntityName] by [field or related entity]
27
28 Constraints and assumptions:
29 - Use JSON for requests and responses
30 - Use RESTful HTTP conventions and standard status codes
31 - The API is public and does not require authentication
32 - Include reusable components under `components.schemas`
33 - Add a `summary` for each operation
```

Obrázek 1 Příklad promptovací šablony k prvnímu experimentu

¹¹ SMARTBEAR Software. Swagger Editor [online]. [cit. 14. 5. 2025]. Dostupné z: <https://editor.swagger.io/>

Výsledky a diskuze

Byly provedeny celkem 3 experimenty, jejíž výsledky považuji za nutné interpretovat. Podrobné vstupy a výstupy jednotlivých experimentů jsou uvedeny v přílohách.

Experiment 1

První experiment spočíval v generování OpenAPI dokumentace z přirozeného jazyka, resp. z vyplněné promptovací šablony pro tento experiment.

Tabulka 1 Hodnocení výstupů prvního experimentu

Výstupem je JSON	ano
JSON je well-formed	ano
JSON odpovídá OpenAPI 3.0 specifikaci	ano
Počet správných endpointů / počet očekávaných	6/6
Subjektivní skóre	3/5

Výstup LLM v experimentu 1 je sice technicky korektní a validní podle OpenAPI 3.0 specifikace, ale postrádá jemnější návrhové detaily. LLM úspěšně vytvořil základní CRUD operace pro entitu Book i její závislé entity Author a Genre, včetně filtrování podle autory a genreId. Chybí ale například stránkování, typické pro produkční API. Schémata sice odpovídají základnímu zadání, ale nejsou rozšířená o validační pravidla (minLength, enum, pattern). Výstup neobsahuje ani jednotný model pro chybové odpovědi, což považuji za chybné. Oddělení modelů BookInput a Book ukazuje snahu LLM o čistý návrh, i když volba názvu BookInput mi přijde spíše nešťastná. Celkově výstup dobře reaguje na prompt, ale neprojevuje žádnou vlastní iniciativu, resp. vyšší úroveň návrhové inteligence, což je na jednu stranu dobře, protože plně respektuje prompt uživatele, ale zároveň neodhalí nedostatky. Hodí se jako výchozí bod, ale bez zásahu vývojáře za mě není okamžitě připraven k produkci.

Použitý prompt a jeho výstup: <https://github.com/ondrejsvorc/SWI/blob/main/Experiment1.md>

Experiment 2

Druhý experiment spočíval v generování vylepšené verze OpenAPI dokumentace ve formátu JSON, která byla výstupem prvního experimentu.

Tabulka 2 Hodnocení výstupů druhého experimentu

Výstup zůstává validní dle OpenAPI 3.0 specifikace	ano
Počet nově doplněných prvků	2
Subjektivní skóre	4/5

Výstup LLM v experimentu 2 představuje kvalitativní posun oproti prvnímu. Struktura zůstává zachována, ale dokumentace je obohacena o example bloky. LLM správně doplnil i validační constraints (např. minLength, maxLength, format), což původní verze postrádala, protože to nebylo explicitně specifikováno. Vzniklo také nové jednotné Error schéma. Přesto se mi některé části jeví jako zbytečné. Sekce s příklady jsou příliš rozsáhlé a opakují se. Dokumentace by šla ještě dále optimalizovat pomocí \$ref nebo externalizací opakujících se struktur. Celkově však výstup představuje dobrý výchozí bod pro veřejně publikovatelné API, pokud by byla dále přidána forma autentizace, která není předmětem tohoto experimentu.

Použitý prompt a jeho výstup: <https://github.com/ondrejsvorc/SWI/blob/main/Experiment2.md>

Experiment 3

Třetí experiment spočíval v generování REST API endpointů z OpenAPI dokumentace ve formátu JSON, která byla výstupem prvního druhého experimentu, resp. mohlo se jednat o výstup prvního experimentu, ale byl zvolen výstup druhého, protože se jedná o lepší verzi toho prvního.

Tabulka 3 Hodnocení výstupů třetího experimentu

Výstupem je C# kód	ano
Kód je po vložení do Program.cs kompilovatelný	ano
Počet správných endpointů / počet očekávaných	7/7
Subjektivní skóre	2/5

LLM dokázal vytvořit kompilovatelný kód s použitím Entity Framework Core¹². Kód ale bohužel není zcela funkční. Třídy pro entitní modely vypadají na první pohled v pořádku, ale odkazy na jiné entity by měly být navigačními vlastnostmi, tedy typu virtual ICollection místo List. Dále chybí definice relací mezi entitami ve FakeDbContext. Dále chybí validace vstupních dat do REST API endpointů a kód neřeší edge-cases (např. neexistující AuthorId při POST/PUT), ani optimalizaci na úrovni ORM (např. AsNoTracking). Co nechybí je dependency injection ani Include(...) pro eager loading vztahů. Výstup bych označil za spíše nedostačující a rozhodně se nejedná o základ, který by stačilo vývojářem doplnit pouze o jemné detaily.

Použitý prompt a jeho výstup: <https://github.com/ondrejsvorc/SWI/blob/main/Experiment3.md>

¹² Microsoft. Entity Framework Core. [online]. [cit. 2025-05-15]. Dostupné z: <https://learn.microsoft.com/en-us/ef/core/>

Závěr

LLM o4-mini-high prokázal schopnost generovat validní i použitelnou OpenAPI dokumentaci na základě přirozeného jazyka. Také ukázal svůj potenciál pro tvorbu základního C# REST API kódu dle této dokumentace. Výsledky naznačují, že opačný proces (od dokumentace ke kódu) může být vhodný pro zrychlení počátečních fází návrhu tím, že poskytne výchozí bod, od něž se mohou vývojáři odrazit a dále jej zlepšovat.

Omezením výzkumu bylo spoléhání se na jediný LLM (o4-mini-high) a statické promptovací šablony. LLM se řídí pouze explicitně zadanými instrukcemi a nevnímá implicitní očekávání (např. stránkování), které si vývojář osvojí praxí a myslí na ně. Výsledky tak stále vyžadují následnou revizi zkušeným vývojářem, který výstup kriticky zhodnotí. Celkově si ale LLM vedlo dobře při generování OpenAPI dokumentace, u generování kódu to už bylo horší.

Na výzkum lze navázat porovnáním více modelů (např. o4-mini-high vs DeepSeek v3) nebo srovnat výstupy s reálnými návrhy vývojářů různé seniority. Lze také zkusit vymyslet lepší promptovací šablony či přijít s jinou strategií a porovnávat ji s tou stávající.

Seznam obrázků

Obrázek 1 Příklad promptovací šablony k prvnímu experimentu 5

Seznam tabulek

Tabulka 1 Hodnocení výstupů prvního experimentu..... 6

Tabulka 2 Hodnocení výstupů druhého experimentu 7

Tabulka 3 Hodnocení výstupů třetího experimentu..... 8

Zdroje

OPENAI. Introducing OpenAI o3 and o4-mini [online]. 16. dubna 2025 [cit. 12. května 2025]. Dostupné z: <https://openai.com/index/introducing-o3-and-o4-mini/>

OpenAPI Initiative. OpenAPI Specification v3.0.0. [online]. 2020 [cit. 2025-05-15]. Dostupné z: <https://spec.openapis.org/oas/v3.0.0>

SmartBear Software. Swagger – Open Source API Tool. [online]. [cit. 2025-05-15]. Dostupné z: <https://swagger.io/>

SMARTBEAR SOFTWARE. Swagger Codegen [online]. 2025 [cit. 2025-05-17]. Dostupné z: <https://swagger.io/tools/swagger-codegen/>

OpenAI. OpenAI O3 mini. In: *OpenAI* [online]. 2024 [cit. 2025-05-15]. Dostupné z: <https://openai.com/index/openai-o3-mini/>

Kamali, M. Evaluating GPT-4 for OpenAPI Generation and Conversion. Bachelor's Thesis. Uppsala University, 2024. Dostupné z: <https://www.diva-portal.org/smash/get/diva2:1877570/FULLTEXT01.pdf>

DEEPSEEK AI. *DeepSeek V3* [online]. [cit. 2025-05-17]. Dostupné z: <https://deepseekv3.org/>

DENG, Sida; HUANG, Rubing; ZHANG, Man; CUI, Chenhui; TOWEY, Dave; WANG, Rongcun. *LRASGen: LLM-based RESTful API Specification Generation* [online]. 2025 [cit. 2025-05-17]. Dostupné z: <https://arxiv.org/abs/2504.16833>

SyncRO Soft. JSON well-formedness. In: *Oxygen XML Editor User Guide* [online]. Version 27.1. [cit. 2025-05-15]. Dostupné z: <https://www.oxygenxml.com/doc/versions/27.1/ug-json/topics/json-well-formedness.html>

Fundamentals. ASP.NET Core [online]. [cit. 17. 5. 2025]. Dostupné z: <https://learn.microsoft.com/en-us/aspnet/core/fundamentals/?view=aspnetcore-9.0&tabs=windows>

SMARTBEAR Software. Swagger Editor [online]. [cit. 14. 5. 2025]. Dostupné z: <https://editor.swagger.io/>

Microsoft. Entity Framework Core. [online]. [cit. 2025-05-15]. Dostupné z: <https://learn.microsoft.com/en-us/ef/core/>