

# Úkol: Webová aplikace pro správu filmů pomocí API

Vaším úkolem je vytvořit webovou aplikaci pomocí PHP Nette frameworku, která bude sloužit pro správu filmů pomocí REST API. Veškeré pojmenování, struktura databáze je plně ve vaší režii. K výsledku práce prosím přiložte SQL soubor, kde bude definovaná struktura tabulky pro filmy.

## Požadavky:

### Základní funkce:

- Přehled filmů: Je možné zobrazit seznam všech filmů v knihovně pomocí API
- Detail filmu: Je možné zobrazit detaily o konkrétním filmu, včetně názvu, autora, žánru a popisu.
- Vytvoření filmu: Je možné přidat nový film do systému pomocí API.
- Editace filmu: Je možné upravit existující film pomocí API.
- Smazání filmu: Je možné smazat film pomocí API.

### Volitelné funkce:

- Endpoint pro získání všech filmů je možné rozšířit také o volitelnou filtraci, případné stránkování.
- Docker: Vytvořte Dockerfile a docker-compose.yml pro jednoduchou správu vývojového prostředí.
- Statická analýza kódu: Využijte nástroje pro statickou analýzu kódu (např. PHPStan, CodeSniffer) k identifikaci potenciálních chyb a nedostatků ve vašem kódu.
- Testování: Přidejte Unit testy, pokud je to možné, a zvažte integrační testy pro práci s API.

### Návrh API konfigurace:

- Autorizace: Token-based autorizace / Basic Auth (je to plně na vás)
- **Endpointy:**
  - [GET] /api/v1/movies - Získání seznamu filmů
  - [GET] /api/v1/movies/{id} - Získání detailu konkrétním filmu
  - [POST] /api/v1/movies - Vytvoření nového filmu
  - [PUT] /api/v1/movies/{id} - Editace filmu
  - [DELETE] /api/v1/movies/{id} - Mazání filmu

**Očekávání a hodnocení:**

- Kvalita kódu: Očekává se čistý a dobře strukturovaný kód, vhodně okomentovaný.
- Bezpečnost: Zajistěte, aby aplikace byla bezpečná před útoky jako SQL injection nebo CSRF.
- Dokumentace: Přidejte dokumentaci k funkcím a metodám tam, kde je to vhodné.
- Git: Udržujte historii změn pomocí Gitu a dodržujte vhodnou strukturu commitů.
- Chybové stavy: v případě nevalidních vstupů/výstupu API endpointů se očekávají vhodné chybové stavy, které popisují problém, který nastal.

**Další poznámky:**

- Používejte vhodné návrhové vzory a strukturu kódu.
- Zkuste dbát na to, aby aplikace byla rozšiřitelná a udržitelná.
- Veškerá data (např. autentizační token pro API, struktura JSON atd.) si můžete vymyslet.

**Odevzdání:**

- Ideální forma odevzdání je nasdílený GIT repozitář.
- Nezapomeňte přiložit vaši strukturu DB (tabulky).