

# Zadání - TypeScript/MongoDB Developer

**Kontext:** V rámci našeho telemedicínského řešení potřebují lékaři možnost vytvářet a spravovat úkoly pro své pacienty. Tyto úkoly budou pacientům zobrazovány v mobilní aplikaci, přičemž server bude zajišťovat jejich synchronizaci a zasílání push notifikací prostřednictvím mobilní aplikace. Lékaři budou úkoly vytvářet a spravovat prostřednictvím webové aplikace. Tento úkol se zaměřuje na návrh a implementaci serverové části, která bude tyto funkce zajišťovat.

## Zadání:

1. **Struktura objektu úkolu:** Lékař bude mít možnost v **webové aplikaci** vytvořit úkol pro pacienta, který bude obsahovat:
  - **Nadpis úkolu** – Krátký text (např. „Změř si tlak“).
  - **Popis úkolu** – Detailnější informace o úkolu.
  - **Datum zahájení úkolu** – Datum, od kdy je úkol aktivní.
  - **Datum ukončení úkolu** – Datum, do kdy úkol platí.
  - **Frekvence úkolu** – Variabilní rozvrh, který může být např.:
    - Obden.
    - 1x týdně.
    - 1x za 4 týdny.
    - Každý druhý týden v pondělí.
    - 2x denně v pondělí, středu a čtvrtku.
    - V pondělí, středu a pátek jednou za 4 týdny.
  - **Čas úkolu** – Úkoly se vždy zobrazují pacientovi v 9:00. Každý úkol může být v daný den maximálně jednou.
  - **Notifikace** – Možnost nastavit více upozornění (např. 1 hodinu před a 3 hodiny před časem úkolu).
2. **REST API pro mobilní aplikaci:**
  - Navrhněte **REST API**, které bude mobilní aplikace používat k (mobilní aplikace umí i po omezenou dobu fungovat offline):
    - Získání seznamu aktuálních a budoucích úkolů pro konkrétního pacienta.
    - Oznámení, že pacient úkol splnil (označení úkolu jako splněného).
    - Získání seznamu splněných úkolů.
  - API stačí definovat, ideálně v OpenAPI, ale pokud OpenAPI nepoužíváte, můžete strukturu popsat i textově nebo graficky.
3. **REST API pro webovou aplikaci (pro lékaře):**
  - Navrhněte **REST API**, které bude webová aplikace používat k:
    - Vytváření nových úkolů.
    - Aktualizaci stávajících úkolů.
    - Mazání úkolů.
    - Získání seznamu vytvořených úkolů včetně jejich detailů.

- API stačí definovat, ideálně v OpenAPI, ale pokud OpenAPI nepoužíváte, můžete strukturu popsat i textově.
- 4. **Generování jednotlivých úkolů:**
  - Vytvořte server v **TypeScriptu**, kterému bude webová aplikace prostřednictvím navrženého REST API posílat úkoly zadané lékařem. Server uloží úkoly do MongoDB a následně vygeneruje konkrétní instance úkolů pro pacienta, které by se v budoucnu synchronizovaly do mobilní aplikace.
  - Implementace API pro mobilní aplikaci pacienta není v tomto zadání potřeba.
- 5. **K promyšlení:**
  - **Označení úkolu jako splněného:** Jakým způsobem bude mobilní aplikace informovat server o tom, že pacient úkol splnil? Jak bude tato informace uložena do databáze a jak bude možné sledovat historii splněných úkolů?
  - **Notifikace:** Jak bude server generovat a spravovat push notifikace? Notifikace není třeba implementovat, ale je důležité promyslet jejich logiku a strukturu.
  - **Autentizace:** Jakým způsobem by mohla být implementována autentizace, například pomocí OpenID Connect?
  - **Synchronizace úkolů do mobilní aplikace:** Jak byste řešili synchronizaci úkolů do mobilní aplikace, aby se úkoly nemusely neustále stahovat znovu a znovu, pokud je aplikace již má stažené?
  - **Testování:** Promyslete základní scénáře pro testování API a logiky generování úkolů. Implementace testů není nutná, stačí promyslet možné testovací případy a popsat je textově.

## Očekávané výstupy:

1. **Návrh dvou REST API** (pro webovou a mobilní aplikaci) – ideálně v OpenAPI, ale postačí i textový popis.
2. **Server v TypeScriptu**, kterému bude webová aplikace posílat úkoly od lékaře, uloží je do MongoDB a generuje konkrétní instance úkolů, které by se v budoucnu synchronizovaly do mobilní aplikace.

## Technologie:

- **Backend:** TypeScript
- **Databáze:** MongoDB
- **API:** REST (OpenAPI výhodou, ale ne nutností)
- **Autentizace:** OpenID Connect (k promyšlení)
- **Testování:** Promyšlení základních testovacích scénářů (implementace není nutná)

## Časový rámec

Na toto zadání si vyhradte maximálně 16 hodin. Chápeme, že zadání je obsáhlé, proto je prioritou definice REST API a následně implementace schedulingu a přijímání úkolů od lékařů. Ostatní části můžeme dále prodiskutovat na základě toho, co se vám podaří stihnout.

Těšíme se na vaše řešení!