## Zadání úlohy pro výběrové řízení na senior .Net programátora

### Úvod

Vaším úkolem je implementovat REST službu v C# pomocí .NET frameworku, která bude mít dvě hlavní operace. Tato služba bude pracovat s entitami "Osoba", "Bydliště" a "Kontakt" a bude umožňovat základní CRUD operace GET a Insert.

### Specifikace úlohy

#### 1. Vytvoření RESTové služby

* Použijte .NET framework 4.8 nebo vyšší
* Vytvořte nový projekt ASP.NET Web API.

#### 2. Definice entit "Osoba", "Bydliště" a "Kontakt"

* Entita "Osoba" by měla mít následující vlastnosti:
  + Id (typ: int, automaticky generované, povinný)
  + Jmeno (typ: string, max. délka: 50 znaků, povinný)
  + Prijmeni (typ: string, max. délka: 50 znaků, povinný)
  + RodnePrijmeni (typ: string, max. délka: 50 znaků, nepovinný)
  + RodneCislo (typ: string, přesně 10 znaků, validace formátu pro české rodné číslo, povinný)
  + DatumNarozeni (typ: DateTime, povinný)
  + Narodnost (typ: string, číselník národností, povinný)
  + Bydliste (typ: Bydliště, vztah 1:1 s entitou Bydliště, povinný)
  + Kontakty (typ: kolekce Kontaktů, vztah 1:N s entitou Kontakt, povinný)
* Entita "Bydliště" by měla mít následující vlastnosti:
  + Id (typ: int, automaticky generované, povinný)
  + Ulice (typ: string, max. délka: 100 znaků, nepovinný)
  + Mesto (typ: string, max. délka: 50 znaků, povinný)
  + PSC (typ: string, max. délka: 10 znaků, povinný)
  + Stat (typ: string, číselník států, povinný)
* Entita "Kontakt" by měla mít následující vlastnosti:
  + Id (typ: int, automaticky generované, povinný)
  + TypKontaktu (typ: string, číselník typů kontaktů, např. Telefon, Email, povinný)
  + Hodnota (typ: string, max. délka: 100 znaků, povinný)
  + OsobaId (typ: int, cizí klíč na entitu Osoba, povinný)

#### 3. Implementace operací

Implementujte následující operace:

##### a) Vytvoření nové osoby (POST /api/osoby)

* API by mělo přijímat JSON objekt obsahující vlastnosti osoby, jejího bydliště a případně i kontakty bez Id.
* Příklad vstupního JSONu:

json

* {

"jmeno": "Jan",

"prijmeni": "Novák",

"rodneCislo": "8501011234",

"datumNarozeni": "1985-01-01",

"narodnost": "Česká",

"bydliste": {

"ulice": "Hlavní 123",

"mesto": "Praha",

"psc": "11000",

"stat": "Česká republika"

},

"kontakty": [

{

"typKontaktu": "Telefon",

"hodnota": "+420123456789"

},

{

"typKontaktu": "Email",

"hodnota": "jan.novak@example.com"

}

]

}

* Implementujte validace pro rodné číslo (musí být ve formátu českého rodného čísla) a další vhodné validace pro ostatní pole.
* API by mělo vrátit JSON objekt nově vytvořené osoby včetně Id.
* Navrhněte vhodné validace a ošetřete návratové kódy s ohledem na další parametry

##### b) Získání detailu osoby podle ID (GET /api/osoby/{id})

* API by mělo vrátit JSON objekt obsahující všechny vlastnosti osoby včetně jejího bydliště a kontaktů.
* Pokud osoba s daným Id neexistuje, API by mělo vrátit status 404 Not Found.
* Navrhněte vhodné validace a ošetřete návratové kódy s ohledem na další parametry

#### 4. Úložiště dat

* Pro účely této úlohy můžete použít in-memory úložiště. Není potřeba implementovat databázové připojení.
* Doporučený způsob je použití služby (service) pro správu osob, která bude zajišťovat CRUD operace v paměti.

#### 5. Dokumentace a testování

* Vytvořte základní dokumentaci API pomocí Swagger
* Přiložte unit testy pro obě operace pomocí frameworku dle vašeho výběru
* Doplňte logování pomocí frameworku dle vašeho výběru

### Kritéria hodnocení

* Kvalita a čistota kódu.
* Dodržení zadání a správná implementace požadovaných funkcionalit.
* Struktura a organizace projektu.
* Kvalita dokumentace.
* Pokrytí unit testy.

### Odevzdání

* Kód projektu nahrajte do veřejného GitHub repozitáře a pošlete nám odkaz, nebo zdrojové kódy zabalte do archivu a zašlete emailem.
* Přiložte stručný README soubor s popisem projektu a instrukcemi k jeho spuštění.