

## 2. REST házi feladat

---

*Dr. Simon Balázs (sbalazs@iit.bme.hu), BME IIT, 2023.*

A továbbiakban a NEPTUN szó helyére a saját Neptun-kódot kell behelyettesíteni, csupa nagybetűvel.

### 1 A feladat leírása

A feladat egy filmadatbázis rendszer elkészítése, és ennek RESTful szolgáltatásként való publikálása.

A szolgáltatás által kapott és visszaadott adatstruktúrát az alábbi pszeudo-kód írja le:

```
struct Movie
{
    string Title;
    int Year;
    string Director;
    string[] Actor; // list of actor names
}
```

### 2 A szolgáltatás

A szolgáltatást egy Maven vagy egy .NET Core webalkalmazásban kell megvalósítani. A két technológia közül szabadon lehet választani.

A webalkalmazás neve a következő: **Rest\_NEPTUN**

A szolgáltatás báziscíme a következő URL:

**http://localhost:8080/Rest\_NEPTUN/resources/MovieDatabase**

A szolgáltatásnak adatokat kell tárolnia az egyes hívások között. Egy valódi alkalmazás esetén az adatok tárolására adatbázist kéne használni. A házi feladatban a könnyebbség kedvéért *statikus változóknak* tároljuk a szükséges adatokat!

#### 2.1 Maven választása esetén

A szolgáltatást egy ugyanolyan Maven webalkalmazásban kell megvalósítani, mint amilyen a tutorialban szerepel. A **pom.xml**-nek kötelezően a csatolt **service/pom.xml**-nek kell lennie, de a NEPTUN szót le kell cserélni benne a saját Neptun-kódra.

A forrásfájlok készülhetnek Java vagy Kotlin nyelven is.

#### 2.2 .NET Core választása esetén

A szolgáltatást egy ugyanolyan ASP.NET Core webalkalmazásban kell megvalósítani, mint amilyen a tutorialban szerepel. A projektfájlnak kötelezően a csatolt **service/Rest\_NEPTUN.csproj**-nek kell lennie, de a fájl nevében a NEPTUN szót le kell cserélni a saját Neptun-kódra.

## 2.3 Interfész specifikáció

Az alábbi táblázat írja le, hogy a szolgáltatásnak milyen műveleteket kell támogatnia. A táblázatban szereplő URL-ek a fenti báziscímtől relatív értendők.

HTTP metódus	Relatív URL	Paraméterek	Tevékenység
GET	/movies	(nincs)	visszaadja az összes Movie objektum listáját
GET	/movies/{id}	az {id} egy Movie azonosítója	visszaadja az adott azonosítójú Movie objektumot ha ilyen azonosítójú objektum nem létezik, akkor HTTP 404-es (not found) státuszkóddal tér vissza
POST	/movies	HTTP body-ban egy Movie struktúra	beszúrja a kapott Movie objektumot (azonosítót a szerver rendel hozzá), visszaadja a szerver által hozzárendelt azonosítót
PUT	/movies/{id}	az {id} egy Movie azonosítója, HTTP body-ban egy Movie struktúra	beszúrja vagy frissíti a kapott Movie objektumot, azonosítót a kliens rendel hozzá (ha még nem létezik ilyen azonosítójú objektum, akkor beszúr, egyébként frissít)
DELETE	/movies/{id}	az {id} egy Movie azonosítója	törli az adott azonosítójú Movie objektumot
GET	/movies/find?year={year}&orderby={field}	a {year} egy évszám, a {field} értéke lehet <b>Title</b> vagy <b>Director</b>	visszaadja az összes {year} évhez tartozó Movie objektum azonosítóját egy listában a listát rendezni kell a {field} paraméterben megadott kritérium alapján ( <b>Title</b> esetén cím szerint, <b>Director</b> esetén a rendező neve szerint)

A szolgáltatásnak XML és JSON formátumú adatokat is kell tudnia fogadni és visszaadni is. Az alábbi példák megadják az egyes kérések és válaszok pontos formátumát.

Példa egy új film felvételére:

POST	http://localhost:8080/Rest_NEPTUN/resources/MovieDatabase/movies
Request HTTP body:	<pre>{   "title": "Batman Begins",   "year": 2005,   "director": "Cristopher Nolan" }</pre>
Response HTTP body:	<pre>{"id":61}</pre>

Példa egy új film felvételére:

<b>POST</b>	<code>http://localhost:8080/Rest_NEPTUN/resources/MovieDatabase/movies</code>
<b>Request HTTP body:</b>	<pre>&lt;movie&gt;   &lt;title&gt;Batman Begins&lt;/title&gt;   &lt;year&gt;2005&lt;/year&gt;   &lt;director&gt;Cristopher Nolan&lt;/director&gt; &lt;/movie&gt;</pre>
<b>Response HTTP body:</b>	<pre>&lt;result&gt;   &lt;id&gt;61&lt;/id&gt; &lt;/result&gt;</pre>

Példa egy film frissítésére:

<b>PUT</b>	<code>http://localhost:8080/Rest_NEPTUN/resources/MovieDatabase/movies/61</code>
<b>Request HTTP body:</b>	<pre>{   "title": "Batman Begins",   "year": 2005,   "director": "Cristopher Nolan",   "actor": [ "Christian Bale", "Michael Caine" ] }</pre>
<b>Response HTTP body:</b>	

Példa egy film frissítésére:

<b>PUT</b>	<code>http://localhost:8080/Rest_NEPTUN/resources/MovieDatabase/movies/61</code>
<b>Request HTTP body:</b>	<pre>&lt;movie&gt;   &lt;title&gt;Batman Begins&lt;/title&gt;   &lt;year&gt;2005&lt;/year&gt;   &lt;director&gt;Cristopher Nolan&lt;/director&gt;   &lt;actor&gt;Christian Bale&lt;/actor&gt;   &lt;actor&gt;Michael Caine&lt;/actor&gt; &lt;/movie&gt;</pre>
<b>Response HTTP body:</b>	

Példa egy film törlésére:

<b>DELETE</b>	<code>http://localhost:8080/Rest_NEPTUN/resources/MovieDatabase/movies/61</code>
<b>Request HTTP body:</b>	
<b>Response HTTP body:</b>	

Példa egy film lekérésére:

<b>GET</b>	<code>http://localhost:8080/Rest_NEPTUN/resources/MovieDatabase/movies/61</code>
<b>Request HTTP body:</b>	
<b>Response HTTP body:</b>	<pre>{   "title": "Batman Begins",   "year": 2005,   "director": "Cristopher Nolan",   "actor": [ "Christian Bale", "Michael Caine" ] }</pre>

Példa egy film lekérésére:

<b>GET</b>	<b>http://localhost:8080/Rest_NEPTUN/resources/MovieDatabase/movies/61</b>
<b>Request HTTP body:</b>	
<b>Response HTTP body:</b>	<pre> &lt;movie&gt;   &lt;title&gt;Batman Begins&lt;/title&gt;   &lt;year&gt;2005&lt;/year&gt;   &lt;director&gt;Cristopher Nolan&lt;/director&gt;   &lt;actor&gt;Christian Bale&lt;/actor&gt;   &lt;actor&gt;Michael Caine&lt;/actor&gt; &lt;/movie&gt; </pre>

Példa az összes film lekérésére:

<b>GET</b>	<b>http://localhost:8080/Rest_NEPTUN/resources/MovieDatabase/movies</b>
<b>Request HTTP body:</b>	
<b>Response HTTP body:</b>	<pre> {   "movie": [     {       "title": "Batman Begins",       "year": 2005,       "director": "Cristopher Nolan",       "actor": [ "Christian Bale", "Michael Caine" ]     },     {       "title": "Pirates of the Caribbean: The Curse of the Black Pearl",       "year": 2003,       "director": "Gore Verbinski",       "actor": [ "Johnny Depp", "Geoffrey Rush" ]     }   ] } </pre>

Példa az összes film lekérésére:

<b>GET</b>	<b>http://localhost:8080/Rest_NEPTUN/resources/MovieDatabase/movies</b>
<b>Request HTTP body:</b>	
<b>Response HTTP body:</b>	<pre> &lt;movies&gt;   &lt;movie&gt;     &lt;title&gt;Batman Begins&lt;/title&gt;     &lt;year&gt;2005&lt;/year&gt;     &lt;director&gt;Cristopher Nolan&lt;/director&gt;     &lt;actor&gt;Christian Bale&lt;/actor&gt;     &lt;actor&gt;Michael Caine&lt;/actor&gt;   &lt;/movie&gt;   &lt;movie&gt;     &lt;title&gt;Pirates of the Caribbean: The Curse of the Black Pearl&lt;/title&gt;     &lt;year&gt;2003&lt;/year&gt;     &lt;director&gt;Gore Verbinski&lt;/director&gt;     &lt;actor&gt;Johnny Depp&lt;/actor&gt;     &lt;actor&gt;Geoffrey Rush&lt;/actor&gt;   &lt;/movie&gt; &lt;/movies&gt; </pre>

Példa év szerinti lekérdezésre:

GET	http://localhost:8080/Rest_NEPTUN/resources/MovieDatabase/movies/find?year=2007&orderby=Director
Request HTTP body:	
Response HTTP body:	{"id": [56, 41]}

Példa év szerinti lekérdezésre:

GET	http://localhost:8080/Rest_NEPTUN/resources/MovieDatabase/movies/find?year=2003&orderby=Title
Request HTTP body:	
Response HTTP body:	<movies> <id>23</id> <id>37</id> </movies>

A szervernek nem kell XML-re XML-t illetve JSON-ra JSON-t visszaadnia. A szervernek azt a formátumot kell visszaadnia, amit a kliens a HTTP Accept fejlécben kér.

### 3 A kliens

Beadandó klienst nem kell készíteni. Tesztelési célra erősen ajánlott saját klienst készíteni, de azt nem kell beadni.

### 4 Segítség a megoldáshoz

A pontos XML tagek illetve JSON attribútumok megfelelő elnevezéséhez tanult annotációkat/attribútumokat érdemes finomhangolni.

Szükség esetén célszerű csomagolóosztályt készíteni a megfelelő XML illetve JSON struktúra eléréséhez. A feladat egyik kihívása a megfelelő csomagolóosztályok létrehozása.

### 5 Beadandók

Beadandó egyetlen ZIP fájl. Más tömörítési eljárás használata tilos!

A ZIP fájl gyökerében egyetlen könyvtárnak kell lennie, a szolgáltatás alkalmazása:

- **Rest\_NEPTUN:** a szolgáltatás Maven vagy .NET Core projektje teljes egészében

A szolgáltatásnak a fenti példákban szereplő XML és JSON formátumokat kell támogatnia. Más XML vagy JSON formátum/elnevezés használata tilos!

A megoldásnak a telepítési leírásban meghatározott környezetben kell fordulnia és futnia, a **pom.xml**-ek illetve a **csproj** fájlok tartalmát a Neptun-kód módosításán felül megváltoztatni tilos!

Fontos: a dokumentumban szereplő elnevezéseket és kikötéseket pontosan be kell tartani! Még egyszer kiemelve: a NEPTUN szó helyére mindig a saját Neptun-kódot kell behelyettesíteni, csupa nagybetűvel!

## 5.1 Maven választása esetén

Mielőtt a projektet becsomagolnánk a ZIP-be, a projekten belül adjuk ki az **mvn clean** parancsot! Ez a parancs törli a **target** alkönyvtárat, amely elég nagy is lehet és egyébként sem lesz figyelembe véve a kiértékelés során.

A ZIP kibontása után a szolgáltatásnak parancssorból fordulnia kell a következő parancs kiadásával:  
`...\Rest_NEPTUN>mvn package`

A szolgáltatásnak parancssorból települnie kell a szerverre következő parancs kiadásával:  
`...\Rest_NEPTUN>mvn wildfly:deploy`

## 5.2 .NET Core választása esetén

Mielőtt a projektet becsomagolnánk a ZIP-be, a projekten belül adjuk ki a **dotnet clean** parancsot! Ez a parancs törli a **bin** és **obj** alkönyvtárakat, amelyek elég nagyok is lehetnek és egyébként sem lesznek figyelembe véve a kiértékelés során.

A ZIP kibontása után a szolgáltatásnak parancssorból fordulnia kell a következő parancs kiadásával:  
`...\Rest_NEPTUN>dotnet build`

A szolgáltatásnak parancssorból futnia kell a következő parancs kiadásával:  
`...\Rest_NEPTUN>dotnet run`