

2. REST házi feladat

Dr. Simon Balázs (sbalazs@iit.bme.hu), BME IIT, 2019.

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 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 webalkalmazás neve a következő: **Rest_NEPTUN**

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!

A szolgáltatás báziscíme a következő URL:

http://localhost:8080/Rest_NEPTUN/resources/MovieDatabase

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 Title vagy Director	visszaadja az összes {year} évhez tartozó Movie objektum azonosítóját egy listában a listát rendezni kell a {field} paraméterben megadott kritérium alapján (Title esetén cím szerint, Director 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:

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

Példa egy film frissítésére:

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

Példa egy film frissítésére:

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

Példa egy film törlésére:

DELETE	http://localhost:8080/Rest_NEPTUN/resources/MovieDatabase/movies/61
Request HTTP body:	
Response HTTP body:	

Példa egy film lekérésére:

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

Példa egy film lekérésére:

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

Példa az összes film lekérésére:

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

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

Példa év szerinti lekérdezésre:

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

Példa év szerinti lekérdezésre:

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

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. Saját célra készíthető kliens, de azt nem kell beadni.

4 Segítség a megoldáshoz

A pontos XML tagek illetve JSON attribútumok megfelelő elnevezéséhez az **@XmlElement** és **@XmlRootElement** annotációk **name** attribútumát 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 projektje teljes egészében

A lefordított class fájlokat (target alkönyvtár) nem kötelező beadni.

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
```

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 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 nagybetűkkel!