

Lépések

1. Hozzunk létre egy dinamikus web projektet RestExample néven
2. WEBCONTENT/web.xml tartalma legyen a következő:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd" id="WebApp_ID"
version="3.1">
  <display-name>RestExample</display-name>
  <servlet-mapping>
    <servlet-name>javax.ws.rs.core.Application</servlet-name>
    <url-pattern>/*</url-pattern>
  </servlet-mapping>
</web-app>
```

Ebben azt állítjuk be, hogy a beépített szervlet: javax.ws.rs.core.Application minden http kérést kapjon el.

3. Hozzuk létre a Book osztályt:

```
package com.unimiskolc;

import javax.xml.bind.annotation.XmlAccessType;
import javax.xml.bind.annotation.XmlAccessorType;
import javax.xml.bind.annotation.XmlAttribute;
import javax.xml.bind.annotation.XmlID;
import javax.xml.bind.annotation.XmlRootElement;

@XmlRootElement
@XmlAccessorType(XmlAccessType.NONE)
public class Book {
    @XmlID
    private String id;
    @XmlAttribute
    private String title;

    @XmlAttribute
    private String author;
    public Book() {
    }
    public Book(String id, String title, String author) {
        this.id = id;
        this.title = title;
        this.author = author;
    }
}
```

```
    public String getId() {  
        return id;  
    }  
  
    public void setId(String id) {  
        this.id = id;  
    }  
}
```

4. Hozzuk létre a BookStore osztályt:

```
package com.unimiskolc;  
  
import java.util.ArrayList;  
import java.util.Collection;  
import java.util.List;  
  
import javax.ws.rs.Consumes;  
import javax.ws.rs.DELETE;  
import javax.ws.rs.GET;  
import javax.ws.rs.POST;  
import javax.ws.rs.PUT;  
import javax.ws.rs.Path;  
import javax.ws.rs.PathParam;  
import javax.ws.rs.Produces;  
  
@Path("/")  
@Consumes({ "application/xml", "application/json" })  
@Produces({ "application/xml", "application/json" })  
public class BookStore {  
    List<Book> books;  
    public BookStore() {  
        books = new ArrayList<Book>();  
        books.add(new Book("1", "Gárdonyi Géza", "Egri Csillagok"));  
        books.add(new Book("2", "Kiss Elek", "Az én könyvem"));  
    }  
    @GET  
    @Path("books")  
    public Collection<Book> getBooks(){  
        return books;  
    }  
    @POST  
    @Path("books")  
    public void addBook(Book book){  
        books.add(book);  
    }  
  
    @GET  
    @Path("book/{id}")  
    public Book getBook(@PathParam("id") String id){
```

```
        for (Book book : books) {
            if(book.getId().equalsIgnoreCase(id)) {
                return book;
            }
        }
        return null;
    }

    @PUT
    @Path("book/{id}")
    public void updateBook(@PathParam("id") String id, Book book){
    }

    @DELETE
    @Path("book/{id}")
    public void deleteBook(@PathParam("id") String id){
    }

}
```

4. deploy után a <http://localhost:8080/RestExample/book/2> címen a kettessel jelölt könyvet kapjuk vissza.

5. <http://localhost:8080/RestExample/books> url az összes könyvet visszatérít

From:

<https://edu.iit.uni-miskolc.hu/> - Institute of Information Science - University of Miskolc

Permanent link:

https://edu.iit.uni-miskolc.hu/tanszek:oktatas:informatikai_rendszerek_epitese:jax-rs_webszolgaltatas

Last update: 2016/10/26 19:36

