

Java

jaxb, restlet (-jaxb), jsp, taglibs

JAXB

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="urn:Test.Namespace" xmlns="urn:Test.Namespace">

  <xs:element name="employees">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="employee" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="employee">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="name"/>
      </xs:sequence>
      <xs:attribute name="id" type="xs:int" use="required"/>
    </xs:complexType>
  </xs:element>

  <xs:element name="name" type="xs:string"/>
</xs:schema>
```

xjc -d src -p packagename schema.xsd

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<employees xmlns="urn:Test.Namespace" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="schema.xsd">
  <employee id="1">
    <name>Name</name>
  </employee>
</employees>
```

Restlet

Model (Singleton)

```
Unmarshaller unmarshaller = JAXBContext.newInstance(Employees.class).createUnmarshaller();
Employees employees = (Employees) unmarshaller.unmarshal(new File( pathname: "employees.xml"));

Marshaller marshaller = JAXBContext.newInstance(Employees.class).createMarshaller();
marshaller.setProperty(Marshaller.JAXB_FORMATTED_OUTPUT, true);
marshaller.setProperty(Marshaller.JAXB_NO_NAMESPACE_SCHEMA_LOCATION, "schema.xsd");
marshaller.marshal(employees, new File( pathname: "employees.xml"));
```

Interfaces for every provided object

```
public interface IEmployeesResource {
    @Get("xml")
    Employees read();

    @Put()
    boolean store(Employee employee);

    @Delete
    boolean delete(Employee employee);
}
```

Resources for every provided object [getAttribute("id")]

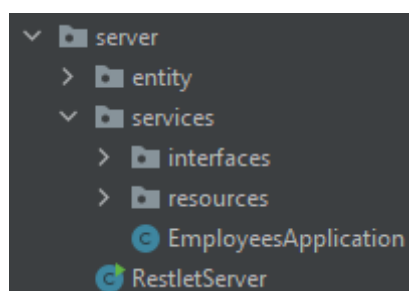
```
public class EmployeesResource extends ServerResource implements IEmployeesResource {  
    @Override  
    public Employees read() {  
        return Model.getEmployees();  
    }  
  
    @Override  
    public boolean store(Employee employee) {  
        return Model.addEmployee(employee);  
    }  
  
    @Override  
    public boolean delete(Employee employee) {  
        return Model.remove(employee);  
    }  
}
```

Application

```
public class EmployeesApplication extends Application {  
    @Override  
    public synchronized Restlet createInboundRoot() {  
        Router router = new Router(getContext());  
  
        router.attach( pathTemplate: "", EmployeesResource.class);  
        router.attach( pathTemplate:("/{id}", EmployeeResource.class);  
        router.attach( pathTemplate:("/{id}/logbook", LogbookEntriesResource.class);  
        router.attach( pathTemplate:("/{id}/logbook/{lid}", LogbookEntriesResource.class);  
        router.attach( pathTemplate:("/{id}/id", NewIDResource.class);  
  
        return router;  
    }  
}
```

Main (Create Restlet Component)

```
Component component = new Component();  
component.getServers().add(Protocol.HTTP, port: 8182);  
component.getDefaultHost().attach( uriPattern: "/api", new EmployeesApplication());  
component.start();
```



JSP

- Add Framework Support (Web Application)
- Project Structure (Tomcat Server Local)
- fix Artifact
- falls nicht geht -> Project Structure/Module/Dependencies (hacker! libs)

Model muss in package sein!

index.jsp form to /employees -> employees Servlet (if no Parameter) -> employees.jsp
(model in req)-> on view click -> employee Servlet -> Parameter id -> Model get Employee by id (model in req) -> on Update -> employees Servlet -> Parameter -> model Update -> model in req

```
@WebServlet(urlPatterns = "/employees")
public class EmployeesServlet extends HttpServlet {
    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        Model model = new Model();

        if (req.getParameter("eid") != null) {
            Employee employee = new Employee();
            employee.setId(Integer.parseInt(req.getParameter("eid")));
            employee.setLastname(req.getParameter("lastname"));
            employee.setFirstname(req.getParameter("firstname"));

            try {
                LocalDate date = LocalDate.parse(req.getParameter("dateOfBirth"));
                XMLGregorianCalendar xmlCal = DatatypeFactory.newInstance().newXMLGregorianCalendar();
                xmlCal.setYear(date.getYear());
                xmlCal.setMonth(date.getMonthValue());
                xmlCal.setDay(date.getDayOfMonth());
                employee.setDateOfBirth(xmlCal);
            } catch (DatatypeConfigurationException ignored) {}

            model.updateEmployee(employee);
        }

        req.setAttribute("model", model);

        RequestDispatcher requestDispatcher = req.getRequestDispatcher("employees.jsp");
        requestDispatcher.forward(req, resp);
    }
}

@WebServlet(urlPatterns = "/employee")
public class EmployeeServlet extends HttpServlet {
    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        Model model = new Model();
        model.setId(Integer.parseInt(req.getParameter("id")));

        req.setAttribute("model", model);

        RequestDispatcher requestDispatcher = req.getRequestDispatcher("detail.jsp");
        requestDispatcher.forward(req, resp);
    }
}
```

```

<%@ taglib prefix = "c" uri = "http://java.sun.com/jsp/jstl/core" %>
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<html>
<head>
<title>Employees</title>
<link rel="stylesheet" href="css/index.css">
</head>
<body>
<jsp:useBean id="model" class="jsp.model.Model" scope="request"/>

<div>
<h1>Employees</h1>
<form action="employee" method="post">
<table>
<c:forEach items="${model.employees.employee}" var="employee">
<tr>
<td>${employee.id}</td>
<td>${employee.lastname}</td>
<td>${employee.firstname}</td>
<td>${employee.dateOfBirth}</td>
<td><button type="submit" name="id" value="${employee.id}">View</button></td>
</tr>
</c:forEach>
<tr><td colspan="5" style="text-align: right;"><button type="submit" name="id" value="-1">Add New</button></td></tr>
</table>
</form>
</div>
</body>
</html>

```

```

<%@ taglib prefix = "c" uri = "http://java.sun.com/jsp/jstl/core" %>
<%@ page contentType="text/html; charset=UTF-8" %>
<html>
<head>
<title>Employee</title>
<link rel="stylesheet" href="css/index.css">
</head>
<body>
<jsp:useBean id="model" class="jsp.model.Model" scope="request"/>
<div>
<form action="/employees" method="post">
<label for="eid">Employee ID</label><br>
<input type="number" id="eid" name="eid" value="${model.employee.id}"><br>
<label for="lastname">Lastname</label><br>
<input type="text" id="lastname" name="lastname" value="${model.employee.lastname}"><br>
<label for="firstname">Firstname</label><br>
<input type="text" id="firstname" name="firstname" value="${model.employee.firstname}"><br>
<label for="dateOfBirth">Date of Birth</label><br>
<input type="date" id="dateOfBirth" name="dateOfBirth" value="${model.employee.dateOfBirth}"><br><br>
<button type="submit">Update</button>
</form>
<table>
<c:forEach items="${model.employee.logbookEntries.logbookEntry}" var="logbookEntry">
<tr>
<td>${logbookEntry.id}</td>
<td>${logbookEntry.activity}</td>
<td>${logbookEntry.startTime}</td>
<td>${logbookEntry.stopTime}</td>
</tr>
</c:forEach>
<tr><td colspan="5" style="text-align: right;"><button type="submit" name="id" value="-1">Add New</button></td></tr>
</table>
</div>
</body>
</html>

```

```

ClientResource resource = new ClientResource( uri: "http://localhost:8182/api");
IEmployeesResource employeesResource = resource.wrap(IEmployeesResource.class);
return employeesResource.read();

```