

Environment

The screenshot displays the Eclipse IDE interface for configuring a Java application named 'werkkrant'. The 'Run/Debug Settings' dialog is open, showing the 'DataOphalen' launch configuration. The 'Edit launch configuration properties' section is visible, with the 'Main class' set to 'nl.java8cursus.rest.DataOphalen'. The 'Libraries' section lists the 'JRE System Library [jdk1.8.0_101]' and 'WildFly 10.x Runtime [WildFly 10.x Runtime]'. The 'Deployment Descriptor' section shows the 'JAX-RS Web Services' configuration, including the 'DataOphalen' service and its methods. The 'REST' annotation is placed near the 'DataOphalen' service, 'JAX' near the 'JAX-RS Web Services' section, 'json' near the 'DataOphalen' service, 'DAO' near the 'PersonDAO' class, and 'Wildfly' near the 'WildFly 10.x Runtime' library.

Run/Debug Settings

This page allows you to manage launch configurations for 'werkkrant'.

Launch configurations for 'werkkrant':

- DataOphalen

Edit launch configuration properties

Run a Java application

Name: DataOphalen

Project: werkkrant

Main class: nl.java8cursus.rest.DataOphalen

☐ Include system libraries when searching for a main class

☐ Include inherited mains when searching for a main class

☐ Stop in main

Deployment Descriptor:

- Context Parameters
- Error Pages
- Filter Mappings
- Filters
- Listeners
- References
- Servlet Mappings
 - /jaxrs/* -> JAX-RS Servlet
- Web Services
 - JAX-RS Web Services
 - DataOphalen
 - GET /rest/aanbod/person
 - consumes: */*
 - produces: application/json
 - DataOphalen.getPersons(...)
 - POST /rest/aanbod/person/1
 - consumes: application/json
 - produces: */*
 - DataOphalen.savePerson(...)

Java Resources

- src
 - nl.java8cursus.rest
 - DataOphalen.java
 - DataOphalen
 - personService
 - getOnePerson(int) : Person
 - getPersons() : List<Person>
 - savePerson(Person) : void
 - Person.java
 - Person
 - PersonDAO.java
 - PersonDAO
 - em
 - getPersonList() : List<Person>
 - savePerson(Person) : void
 - PersonService.java
 - PersonService
 - personDAO
 - getPersonList() : List<Person>
 - savePerson(Person) : void
 - RestApplication.java
 - RestApplication
 - META-INF
 - jobseekers.sql
 - persistence.xml

Libraries

- JRE System Library [jdk1.8.0_101]
- WildFly 10.x Runtime [WildFly 10.x Runtime]

The screenshot displays the Eclipse IDE interface for configuring a Java application named 'werkkrant'. The 'Run/Debug Settings' dialog is open, showing the 'Launch configurations for 'werkkrant'' with 'DataOphalen' selected. The 'Edit launch configuration properties' window is also visible, showing the 'Main' tab with the project 'werkkrant' and the main class 'nl.java8cursus.rest.DataOphalen'. The 'Properties for werkkrant' window shows the 'Run/Debug Settings' tab. The 'Deployment Descriptor' window is open, showing the 'Servlet Mappings' tab with a mapping for 'JAX-RS Servlet'. The 'Project Explorer' on the right shows the project structure, including 'Java Resources' and 'src' folders. The 'Libraries' window at the bottom right shows the 'Wildfly' runtime library.

REST

json

JAX

Wildfly

```

<html>
<head>
<link rel="stylesheet" href="template2.css">
</head>
<body onload="loadDoc()">

  <div class="header">
    <div class="header topmost"></div>
    <div class="header spacing_line_thin"></div>
    <div class="header Logo">

      
      
      
    </div>
    <div class="header search">
      <form method="get" action="/search" id="search"><input type="text" value="" />
    </div>
  </div>
  <div class="header spacing_line_thick"></div>
  <div id="main-content-text">
    <p><span>Ontmoet<a> Onze ICT Professionals</a></span></p>
    <p><span><b>Van Frontend Developer tot Senior Network Engineer</b></span></p>
  </div>
  <div id="main-content-candidates">
    <script src="mainContent.js"></script>
  </div>
  <div id="admin-bar">
    <button onclick="getPersonForm()">Persoon toevoegen</button>
  </div>
  <div id="footer">
    <span>Contact</span>
  </div>
</body>

```

call Javascript

present Form

CSS

```

<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>personForm</title>
</head>
<body>

  <form id="dataInputByForm">

    <fieldset>
      <legend>Information Candidate</legend>

      <label for="filename">Photo (max 2 MB)</label>
      <input id="upload" type="file" onC
      <label for="filename">CV (max 1MB)</label>
      <input id="upload" type="file" onC
      <!-- <input id="filename" type="text" value="" />
    </p></p>

    <input id="submit" onclick="formSubmit()" type="submit" value="Submit" />
  </form>

```

Form for new person

```

WebContent
├── CV
├── Images
├── META-INF
│   └── MANIFEST.MF
├── WEB-INF
│   ├── beans.xml
│   └── web.xml
├── index3.html
├── mainContent.js
│   ├── formSubmit()
│   ├── getPersonForm()
│   ├── loadDoc()
│   ├── loadPerson(id)
│   └── uploadOnChange(e, divName)
├── personForm.html
└── template2.css

```

```

JavaScript Resources
├── WebContent
│   ├── formSubmit()
│   ├── getPersonForm()
│   ├── loadDoc()
│   ├── loadPerson(id)
│   └── uploadOnChange(e, divName)
├── ECMAScript Built-In Library
└── ECMA 3 Browser Support Library

```

template2.css

Persistence

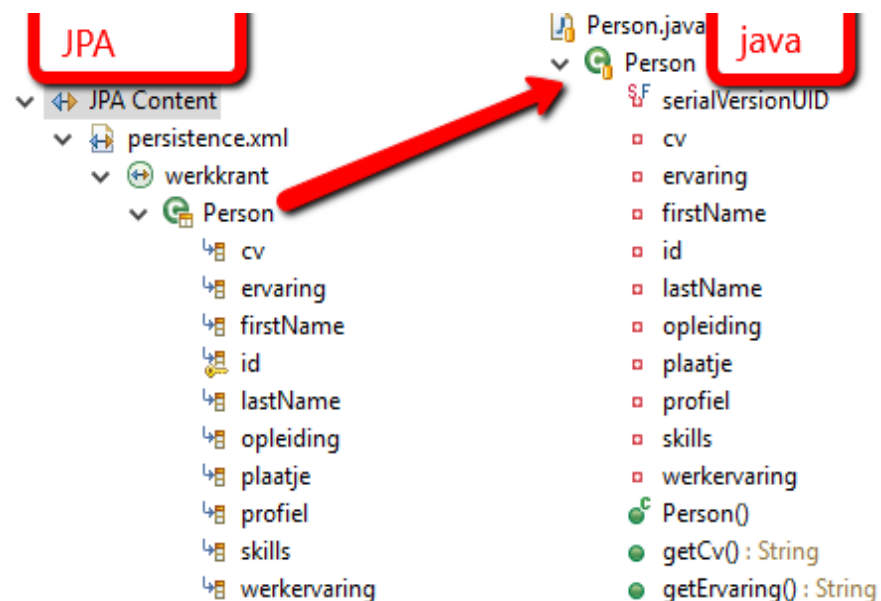
```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.1" xmlns="http://xmlns.jcp.org/xml/ns/persistence" xmlns:xsi="http://www.w3.org/2001/XMLSchema"
  <persistence-unit name="werkkrant">
    <provider>org.hibernate.ejb.HibernatePersistence</provider>
    <jta-data-source>java:jboss/datasources/java8databaseDS</jta-data-source>
    <class>nl.java8cursus.rest.Person</class>
    <properties>
      <property name="javax.persistence.schema-generation.database.action"
        value="drop-and-create" />
      <property name="javax.persistence.schema-generation.create-source"
        value="metadata" />
      <property name="javax.persistence.schema-generation.drop-source"
        value="metadata" />
      <property name="javax.persistence.sql-load-script-source" value="META-INF/jobseekers.sql"/>

      <property name="hibernate.show_sql" value="true" />
      <property name="hibernate.format_sql" value="true" />

      <property name="hibernate.dialect" value="org.hibernate.dialect.PostgreSQLDialect"/>
    </properties>
  </persistence-unit>
</persistence>
```

Hibernate

PostgreSQL



REST : Representational State Transfer

RESTful web services are one way of providing interoperability between computer systems on the Internet, REST-compliant Web services allow requesting systems to access and manipulate textual representations of [Web resources](#) using a uniform and predefined set of [stateless](#) operations. Using HTTP, as is most common, the kind of operations available include those predefined by the HTTP verbs GET, POST, PUT, DELETE and so on.

JAX-RS : Java API for RESTful Web Services is a [Java programming language API](#) spec that provides support in creating [web services](#) according to the [Representational State Transfer](#) (REST) architectural pattern.¹

JAX-WS : Java API for XML Web Services

URI : Uniform Resource Identifier

URL : Uniform Resource Locator

CSS : Cascading Style Sheets

HTML : HyperText Markup Language - creating [web pages](#) and [web applications](#)

AJAX : Asynchronous JavaScript and XML – a collection of related technologies based on JavaScript, XML and the DOM. Ajax is the ability to retrieve new data from the server without unloading the current page.

DOM : Document Object Model - The DOM enables Ajax because, once new data is retrieved, it can be used to insert the data into the page.

JSON : JavaScript Object Notation - JSON: The Fat-Free Alternative to XML

XML : Extensible Markup Language is a [markup language](#) that defines a set of rules for encoding [documents](#) in a [format](#) that is both [human-readable](#) and [machine-readable](#).

JavaScript : the programming language of HTML and the Web - is a high-level, dynamic, untyped, and interpreted programming language.

Java Platforms, Enterprise Edition or Java EE is a widely used [computing platform](#) for [enterprise software](#).

Java : an Object-Oriented programming language with its own runtime environment

DAO : Data Access Object is an [object](#) that provides an abstract [interface](#) to some type of [database](#) or other persistence mechanism.

Hibernate : provides a framework for mapping an object-oriented domain model to a relational database

JPA : Java Persistence API is a [Java application programming interface](#) specification that describes the management of [relational data](#) in applications using [Java Platform](#) SE / EE

PostgreSQL : an object-relational database (ORDBMS) – i.e. a RDBMS, with additional "object" features

Wildfly : (JBoss) Application Server authored by JBoss, now developed by Red Hat. WildFly is written in Java

The **Spring Framework** is an [application framework](#) and [inversion of controlcontainer](#) for the [Java platform](#). The framework's core features can be used by any Java application, but there are extensions for building web applications on top of the [Java EE](#) platform.

Java Platform, Enterprise Edition or **Java EE** is a widely used [computing platform](#) for development and deployment of [enterprise software](#) ([network](#) and [web services](#)).^[1] Java EE was formerly known as **Java 2 Platform, Enterprise Edition** or **J2EE**.