

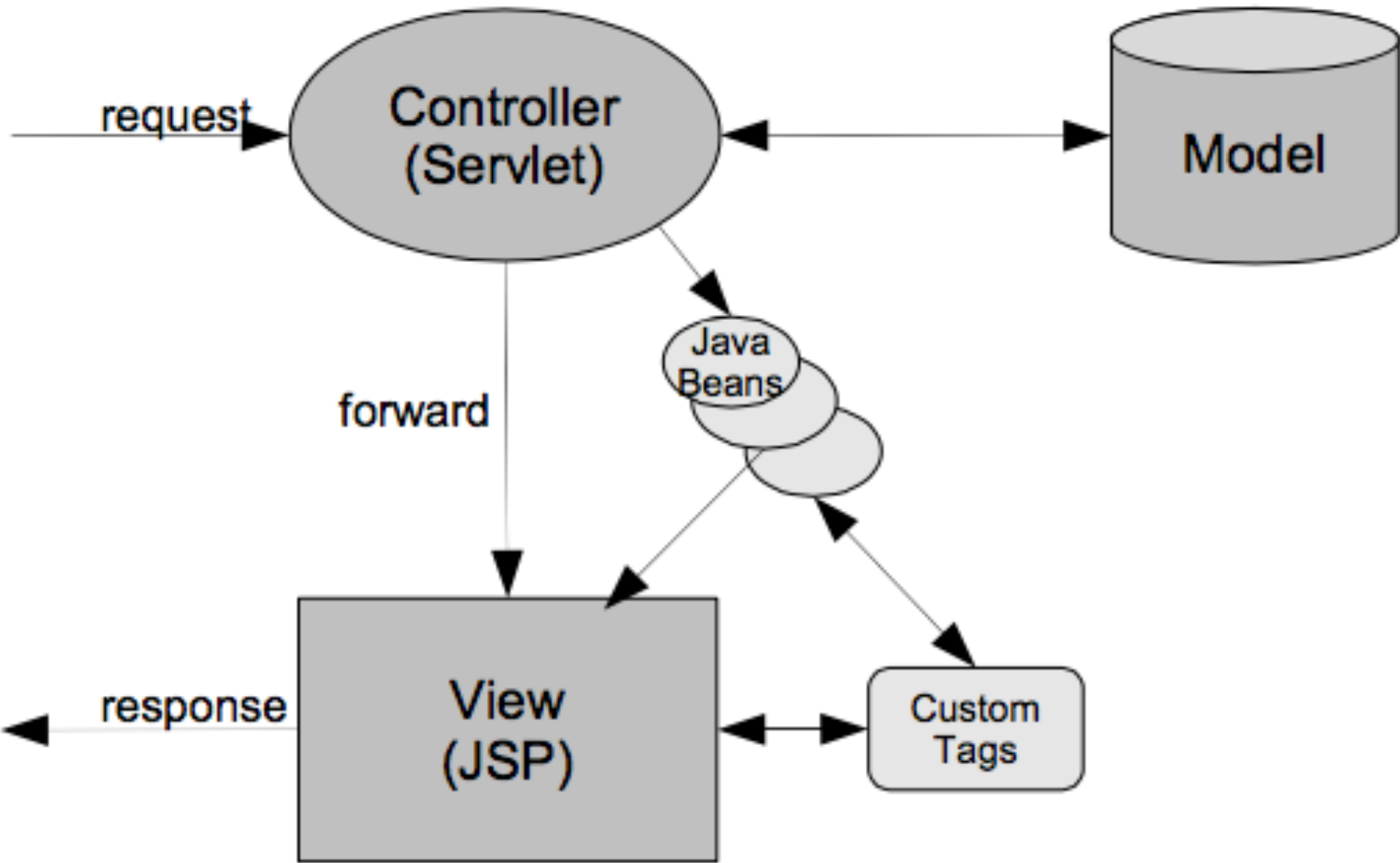
WEB-COMPONENTEN

MODEL-VIEW-CONTROLLER

- ▶ Scriptlets = layout(html) en programmatie(java) verweven
 - ▶ minder gestructureerd
 - ▶ moeilijker te onderhouden
- ▶ MVC(design pattern)
 - ▶ *Separate the user interface into three interconnected components: the model, the view and the controller. Let the model manage the data, the view display the data and the controller mediate updating the data and redrawing the display.*

<https://github.com/iluwatar/java-design-patterns/tree/master/model-view-controller>

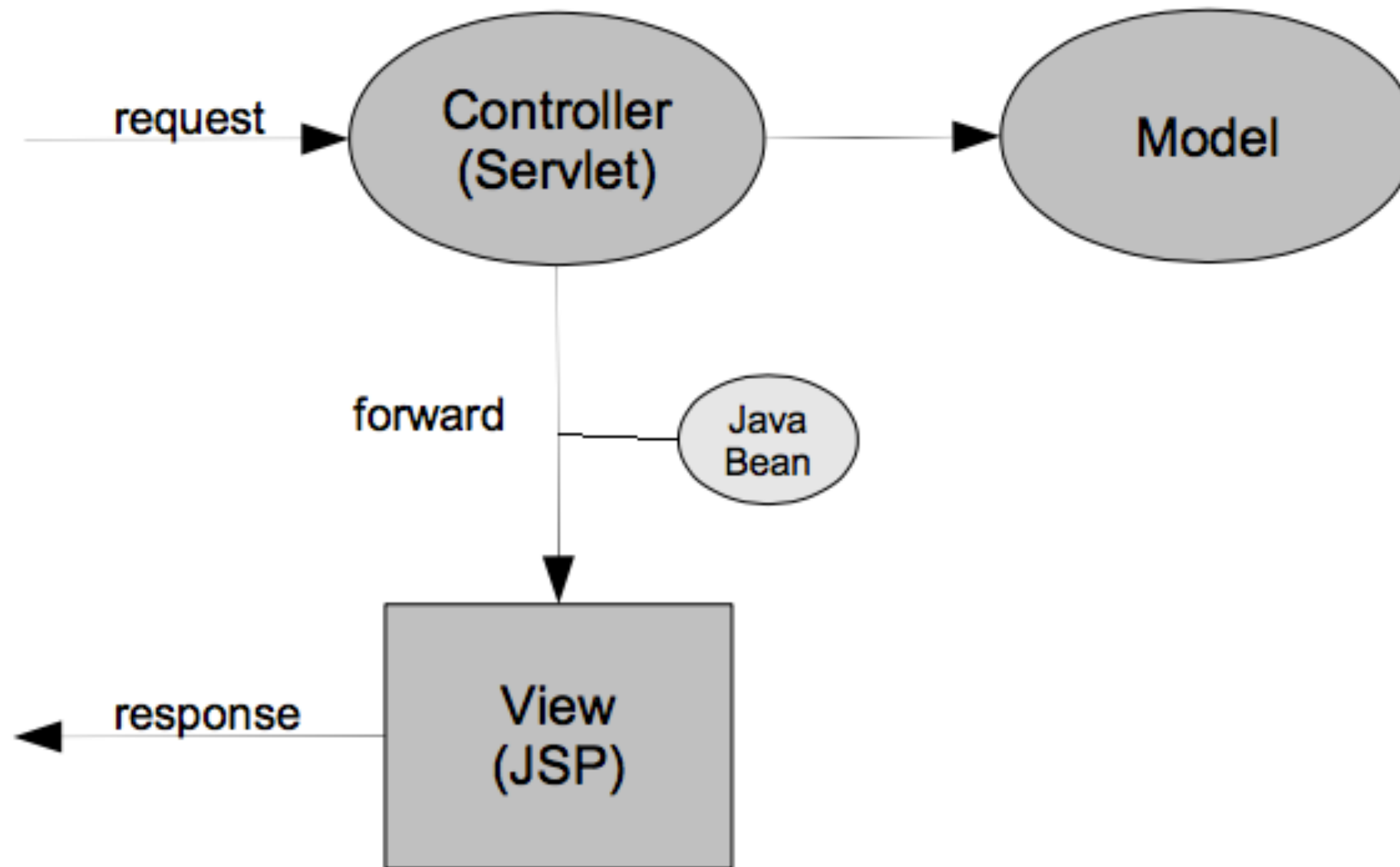
MVC



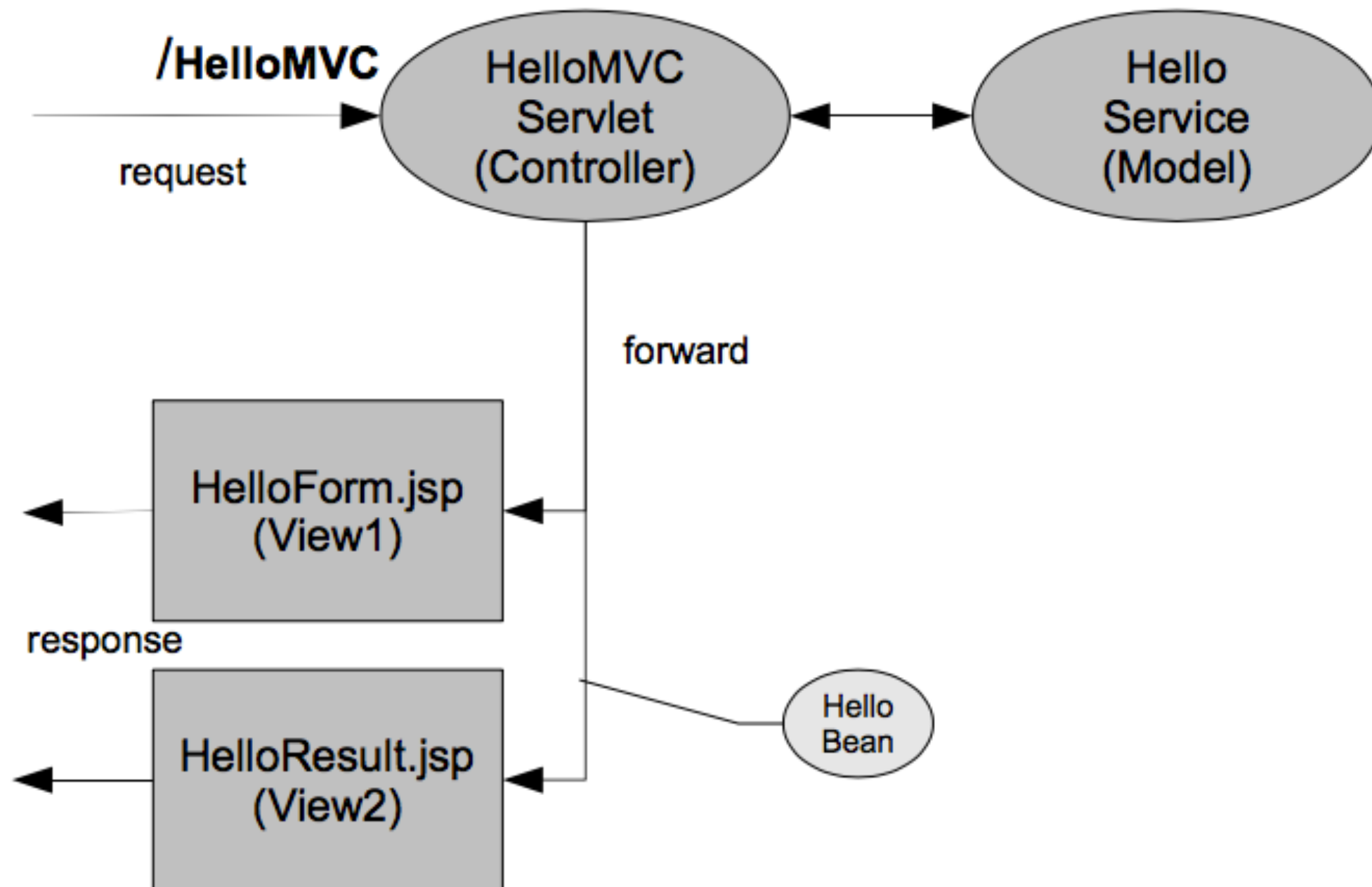
- ▶ EL (expression language)
- ▶ custom tags
- ▶ Java Beans
 - ▶ default no-args constructor
 - ▶ getters/setters
 - ▶ member variables
 - ▶ synthetic variables
 - ▶ implements Serializable

- ▶ default no-args constructor
- ▶ getters/setters
 - ▶ member variables
 - ▶ synthetic variables
- ▶ implements Serializable

MVC - JAVA BEANS



MVC - JAVA BEANS (VOORBEELD)



MVC - JAVA BEANS (VOORBEELD)

```
package eu.noelvaes.web;
import java.io.*;

public class HelloBean implements Serializable{
    private String name;

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}
```

De service:

```
package eu.noelvaes.web;

public class HelloService {
    public void handleHello(HelloBean hello) {
        System.out.println("Handling " + hello.getName());
    }
}
```


MVC - JAVA BEANS (VOORBEELD)

De servlet:

```
package eu.noelvaes.web;
import java.io.*;
import javax.servlet.*;
import javax.servlet.annotation.*;
import javax.servlet.http.*;

@WebServlet("/HelloMVC")
public class HelloMVCServlet extends HttpServlet {
    private HelloService service = new HelloService();

    protected void doGet(HttpServletRequest request,
                          HttpServletResponse response)
        throws IOException, ServletException {
        request.getRequestDispatcher("HelloForm.jsp")
            .forward(request, response);
    }

    protected void doPost(HttpServletRequest request,
                          HttpServletResponse response)
        throws IOException, ServletException {
```

```
        HelloBean hb = new HelloBean();
        hb.setName(request.getParameter("name"));
        service.handleHello(hb);
        request.setAttribute("helloBean", hb);
        request.getRequestDispatcher("HelloResult.jsp")
            .forward(request, response);
    }
}
```

HelloForm.jsp

```
<%@ page contentType="text/html; charset=UTF-8"%>
<html>
<head>
<title>Hello Form</title>
</head>
<body>
    <h1>Hello Form</h1>
    <form method="POST">
        Enter your name:
        <input type="text" name="name" />
        <input type="submit" value="Submit" />
    </form>
</body>
</html>
```

HelloResult.jsp

```
<%@ page contentType="text/html; charset=UTF-8"%>
<html>
<head>
<title>Hello Result</title>
</head>
<body>
    <h1>Hello Result</h1>
    Hello ${helloBean.name}
</body>
</html>
```

- ▶ op een eenvoudige manier expressies opnemen in een JSP
- ▶ syntax: `${..}`
- ▶ Types
 - ▶ Literals, bvb `${'Hello, world'}` of `${12.07}`
 - ▶ Operatoren, bvb `${ 4 + 5}` of `${ 3 < 5 }`
 - ▶ scope-objecten, bvb `${helloBean.name}`
 - ▶ bean, via `#setAttribute(..)` op scope van request of session gezet
 - ▶ page scope, request scope, session scope, application scope

MVC - EL - OPERATOREN

Operator	Omschrijving
<code>==</code> of <code>eq</code>	Test of twee waarden gelijk zijn. Resultaat <code>true</code> of <code>false</code> .
<code>!=</code> of <code>ne</code>	Test of twee waarden verschillend zijn. Resultaat <code>true</code> of <code>false</code> .
<code><</code> of <code>lt</code>	Test of de eerste waarde kleiner is dan de tweede waarde. Resultaat <code>true</code> of <code>false</code> .
<code>></code> of <code>gt</code>	Test of de eerste waarde groter is dan de tweede waarde. Resultaat <code>true</code> of <code>false</code> .
<code><=</code> of <code>le</code>	Test of de eerste waarde kleiner is dan of gelijk is aan de tweede waarde. Resultaat <code>true</code> of <code>false</code> .
<code>>=</code> of <code>ge</code>	Test of de eerste waarde groter is dan of gelijk is aan de tweede waarde. Resultaat <code>true</code> of <code>false</code> .
<code>+</code>	De som van twee waarden.
<code>-</code>	Het verschil van twee waarden.
<code>*</code>	Het produkt van twee waarden.
<code>/</code> of <code>div</code>	Het quotient van twee waarden.
<code>%</code> of <code>mod</code>	De modulo van twee waarden (rest van de deling).
<code>&&</code> of <code>and</code>	EN combinatie tussen twee logische waarden (<code>true</code> of <code>false</code>).
<code> </code> of <code>or</code>	OF combinatie tussen twee logische waarden (<code>true</code> of <code>false</code>).
<code>!</code> of <code>not</code>	Het tegengestelde (inverse) van een logische waarde (<code>true</code> of <code>false</code>).
<code>empty</code>	Test of een bepaalde waarde al dan niet bestaat of ingevuld is.

MVC - EL - PREDEFINED OBJECTS

Naam	Omschrijving
pageContext	Dit is een object van de klasse <code>PageContext</code> dat via zijn <i>getters</i> tevens toegang geeft tot o.a. <code>servletContext</code> , <code>session</code> , <code>request</code> en <code>response</code> .

Naam	Omschrijving
param	Een <i>map</i> van <i>request</i> parameters.
paramValues	Een <i>map</i> van reeksen van <i>request</i> parameters.
header	Een <i>map</i> van <i>header values</i> .
headerValues	Een <i>map</i> van een reeks van <i>header values</i> .
cookie	Een <i>map</i> van <i>cookies</i> .
initParam	Een <i>map</i> van initialisatieparameters.
pageScope	De <i>map</i> van <i>pageContext</i> attributen.
requestScope	De <i>map</i> van <i>request</i> attributen.
sessionScope	De <i>map</i> van <i>session</i> attributen.
applicationScope	De <i>map</i> van <i>servlet context</i> attributen.