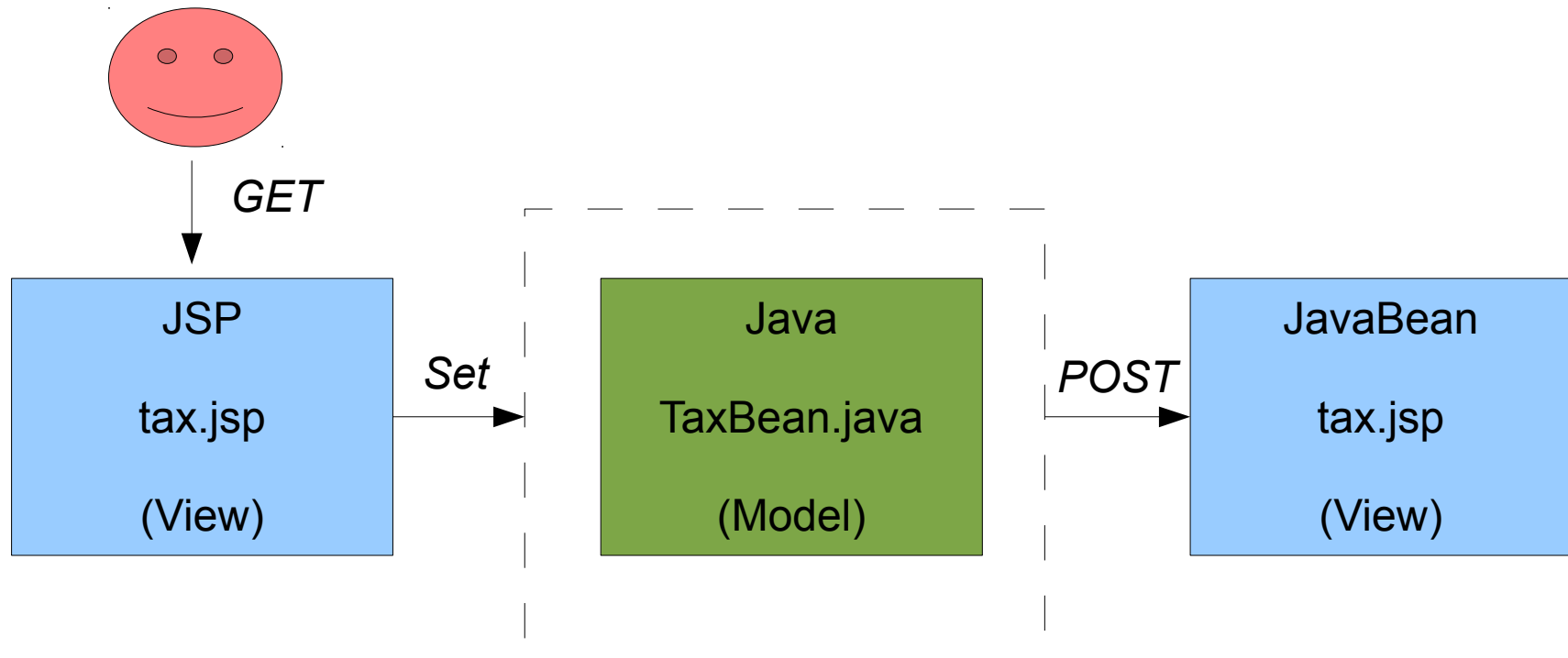


- What is a JavaBean?
 - Its a POJO (Plain Old Java Object)
 - ...Its just a very simple Java Class definition
 - 1. Must have a no-arguments default Constructor – e.g. `public MyBean() { }`
 - 2. Access to properties is through Accessor/Mutator methods – e.g. `getX()` / `setX(newX)` / `isX()`
Follow JavaBean naming conventions!
 - 3. “*Should*” be Serializable, allows for automatic store/restore of state by the container

```
public class NameBean {  
    private String firstName;  
    private String lastName;  
  
    public String getFirstName() {  
        return firstName;  
    }  
  
    public void setFirstName(String firstName) {  
        this.firstName = firstName;  
    }  
  
    ...  
}
```

- JSP and JavaBean Example



- JSP and JavaBean Example – TaxBean.java

```
package com.mycompany.javabeansexample1;
```

```
public class TaxBean implements java.io.Serializable {
```

```
    private double gross;
```

```
    private double taxPercentage;
```

```
    public double getGross() {  
        return gross;  
    }
```

```
    public void setGross(double gross) {  
        this.gross = gross;  
    }
```

```
    public double getTaxPercentage() {  
        return taxPercentage;  
    }
```

```
    public void setTaxPercentage(double taxPercentage) {  
        this.taxPercentage = taxPercentage;  
    }
```

```
    /* TODO Should we place logic in the model? */
```

```
    public double getNet() {  
        return (gross / (100 + getTaxPercentage())) * 100;  
    }  
}
```

Bean Name

Bean properties

Accessor/Mutator Methods

Accessor Method
(with calculation logic)

- JSP and JavaBean Example – tax.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
```

```
<jsp:useBean id="taxBean" class="com.mycompany.javabeansexample1.TaxBean" scope="session" />  
<jsp:setProperty name="taxBean" property="*" />
```

```
<html>  
  <head>  
    <title>Tax Calculator</title>  
  </head>  
  <body>  
    <form method="post">  
      <div>  
        <label for="gross">Gross Amount</label>:  
        <input name="gross" value="${taxBean.gross}" />  
      </div>  
      <div>  
        <label for="gross">VAT Percentage</label>:  
        <input name="taxPercentage" value="{taxBean.taxPercentage}" />  
        <span>(Excluding % symbol)</span>  
      </div>  
      <div><input type="submit" value="Calculate Net" /></div>  
      <div style="margin-top: 2em;">  
        <span>Net Value:</span>  
        <span><%=taxBean.getNet()%></span> <div>i.e. before VAT</div>  
      </div>  
    </form>  
  </body>  
</html>
```

- Problems with the JSP and JavaBeans approach
 - Its not MVC!
 - There is no Controller, its just MV
 - Separation of Concerns is desirable for creating something that is extensible, and maintainable in the long run, e.g. When Marketing ask for a new UI!
 - Where do you put your business logic? In the View? In the Model?
 - For simple/trivial examples, its maybe ok?
e.g. Computing a value in the Model, ok if there are no external dependencies.
 - For complex interactions, its NOT ok!
 - e.g. Database Persistence, Webservice calls, etc.
 - So... for MVC - Use JSP, JavaBeans *and* Servlets
 - Servlets can act as your Controllers

- JSP, JavaBean and Servlet Example – TaxBean.java

```
package com.mycompany.javabeansexample1;
```

```
public class TaxBean implements java.io.Serializable {
```

```
    private double gross;  
    private double taxPercentage;  
    private double net;
```

```
    public double getGross() {  
        return gross;  
    }
```

```
    public void setGross(double gross) {  
        this.gross = gross;  
    }
```

```
    public double getTaxPercentage() {  
        return taxPercentage;  
    }
```

```
    public void setTaxPercentage(double taxPercentage) {  
        this.taxPercentage = taxPercentage;  
    }
```

```
    public double getNet() {  
        return net;  
    }
```

```
    public void setNet(double net) {  
        this.net = net;  
    }
```

Bean Name

Bean properties

Accessor/Mutator Methods

NB. No logic in the Bean (Model)

JSP, JavaBean and Servlet Example – tax.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
```

```
<jsp:useBean id="taxBean" class="com.mycompany.jspandjavabeanandservletsexample.TaxBean"
scope="request"/>
```

```
<html>
  <head>
    <title>Tax Calculator</title>
  </head>
  <body>
    <form method="post" action="tax-proxy.jsp">
      <div>
        <label for="gross">Gross Amount</label>:
        <input name="gross" value="${taxBean.gross}"/>
      </div>
      <div>
        <label for="gross">VAT Percentage</label>:
        <input name="taxPercentage" value="${taxBean.taxPercentage}"/>
        <span>(Excluding % symbol)</span>
      </div>
      <div><input type="submit" value="Calculate Net"/></div>
      <div style="margin-top: 2em;">
        <span>Net Value:</span> <span><%=taxBean.getNet()%></span>
        <div>i.e. before VAT</div>
      </div>
    </form>
  </body>
</html>
```

NB. No property setter
We have to use a proxy
for the Servlet!

JSP, JavaBean and Servlet Example – tax-proxy.jsp

NB. You cant set Bean properties during a form submission to a Servlet, you have to use a Proxy, to set and forward the request.

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
```

```
<jsp:useBean id="taxBean" class="com.mycompany.jspandjavabeanandservletexample.TaxBean"
scope="request"/>
```

```
<jsp:setProperty name="taxBean" property="*/>
```



Our property setter

```
<jsp:forward page="TaxServlet" />
```


JSP, JavaBean and Servlet Example – TaxServlet.java

```
public class TaxServlet extends HttpServlet {

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException,
    IOException {

        //get the tax bean from the request
        TaxBean taxBean = (TaxBean)req.getAttribute("taxBean");

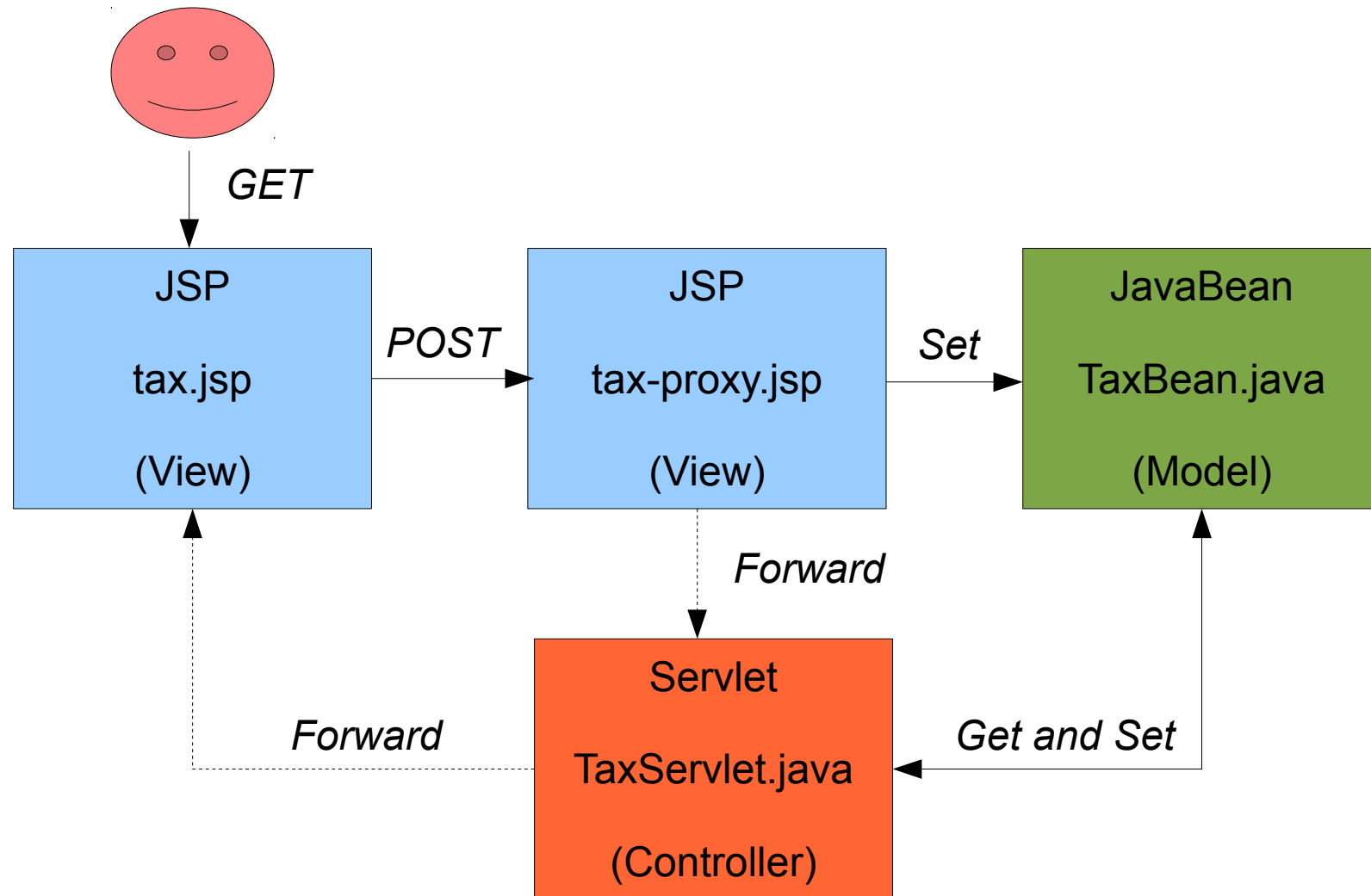
        //perform the calculation
        double net = calculateNet(taxBean.getGross(), taxBean.getTaxPercentage());
        taxBean.setNet(net);

        //set the updated tax bean in the request
        req.setAttribute("taxBean", taxBean);

        //sent to the result page
        RequestDispatcher rd = req.getRequestDispatcher("tax.jsp");
        rd.forward(req, resp);
    }

    /* Business logic for calculating the net value */
    private double calculateNet(double gross, double taxPercentage) {
        return (gross / (100 + taxPercentage)) * 100;
    }
}
```

- JSP, JavaBean and Servlet Example



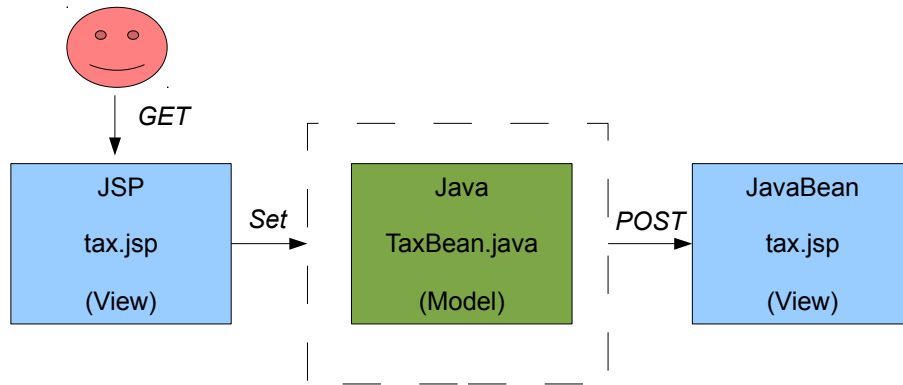
- What is a JavaBean?

- Its a POJO (Plain Old Java Object)
 - ...Its just a very simple Java Class definition
- 1. Must have a no-arguments default Constructor – e.g. `public MyBean() { }`
- 2. Access to properties is through Accessor/Mutator methods – e.g. `getX()` / `setX(newX)` / `isX()`
Follow JavaBean naming conventions!
- 3. “*Should*” be Serializable, allows for automatic store/restore of state by the container

```
public class NameBean {  
    private String firstName;  
    private String lastName;  
  
    public String getFirstName() {  
        return firstName;  
    }  
  
    public void setFirstName(String firstName) {  
        this.firstName = firstName;  
    }  
  
    ...  
}
```

- Good details on JSP and JavaBeans -
http://java.sun.com/j2ee/tutorial/1_3-fcs/doc/JSPBeans.html

- JSP and JavaBean Example



- JSP and JavaBean Example – TaxBean.java

```
package com.mycompany.javabeansexample1;
```

```
public class TaxBean implements java.io.Serializable {
```

```
    private double gross;
```

```
    private double taxPercentage;
```

```
    public double getGross() {  
        return gross;  
    }
```

```
    public void setGross(double gross) {  
        this.gross = gross;  
    }
```

```
    public double getTaxPercentage() {  
        return taxPercentage;  
    }
```

```
    public void setTaxPercentage(double taxPercentage) {  
        this.taxPercentage = taxPercentage;  
    }
```

```
    /* TODO Should we place logic in the model? */
```

```
    public double getNet() {  
        return (gross / (100 + getTaxPercentage())) * 100;  
    }
```

```
}
```

Bean Name

Bean properties

Accessor/Mutator Methods

Accessor Method
(with calculation logic)

- JSP and JavaBean Example – tax.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>

<jsp:useBean id="taxBean" class="com.mycompany.javabeansexample1.TaxBean" scope="session"/>
<jsp:setProperty name="taxBean" property="*" />

<html>
  <head>
    <title>Tax Calculator</title>
  </head>
  <body>
    <form method="post">
      <div>
        <label for="gross">Gross Amount</label>:
        <input name="gross" value="{taxBean.gross}"/>
      </div>
      <div>
        <label for="gross">VAT Percentage</label>:
        <input name="taxPercentage" value="{taxBean.taxPercentage}"/>
        <span>(Excluding % symbol)</span>
      </div>
      <div><input type="submit" value="Calculate Net"/></div>
      <div style="margin-top: 2em;">
        <span>Net Value:</span>
        <span><%=taxBean.getNet()%></span> <div>i.e. before VAT</div>
      </div>
    </form>
  </body>
</html>
```

- Problems with the JSP and JavaBeans approach
 - Its not MVC!
 - There is no Controller, its just MV
 - Separation of Concerns is desirable for creating something that is extensible, and maintainable in the long run, e.g. When Marketing ask for a new UI!
 - Where do you put your business logic? In the View? In the Model?
 - For simple/trivial examples, its maybe ok?
e.g. Computing a value in the Model, ok if there are no external dependencies.
 - For complex interactions, its NOT ok!
 - e.g. Database Persistence, Webservice calls, etc.
 - So... for MVC - Use JSP, JavaBeans *and* Servlets
 - Servlets can act as your Controllers

- JSP, JavaBean and Servlet Example – TaxBean.java

```
package com.mycompany.javabeansexample1;
```

```
public class TaxBean implements java.io.Serializable {
```

```
    private double gross;  
    private double taxPercentage;  
    private double net;
```

```
    public double getGross() {  
        return gross;  
    }
```

```
    public void setGross(double gross) {  
        this.gross = gross;  
    }
```

```
    public double getTaxPercentage() {  
        return taxPercentage;  
    }
```

```
    public void setTaxPercentage(double taxPercentage) {  
        this.taxPercentage = taxPercentage;  
    }
```

```
    public double getNet() {  
        return net;  
    }
```

```
    public void setNet(double net) {  
        this.net = net;  
    }  
}
```

Bean Name

Bean properties

Accessor/Mutator Methods

NB. No logic in the Bean (Model)

JSP, JavaBean and Servlet Example – tax.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
```

```
<jsp:useBean id="taxBean" class="com.mycompany.jspandjavaBeanandServletExample.TaxBean"
scope="request"/>
```

```
<html>
<head>
<title>Tax Calculator</title>
</head>
<body>
<form method="post" action="tax-proxy.jsp">
<div>
<label for="gross">Gross Amount</label>:
<input name="gross" value="${taxBean.gross}"/>
</div>
<div>
<label for="gross">VAT Percentage</label>:
<input name="taxPercentage" value="${taxBean.taxPercentage}"/>
<span>(Excluding % symbol)</span>
</div>
<div><input type="submit" value="Calculate Net"/></div>
<div style="margin-top: 2em;">
<span>Net Value:</span> <span><%=taxBean.getNet()%></span>
<div>i.e. before VAT</div>
</div>
</form>
</body>
</html>
```

NB. No property setter
We have to use a proxy
for the Servlet!

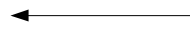
JSP, JavaBean and Servlet Example – tax-proxy.jsp

NB. You cant set Bean properties during a form submission to a Servlet, you have to use a Proxy, to set and forward the request.

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
```

```
<jsp:useBean id="taxBean" class="com.mycompany.jspandjavaBeanandServletexample.TaxBean" scope="request"/>
```

```
<jsp:setProperty name="taxBean" property="*" />
```



Our property setter

```
<jsp:forward page="TaxServlet" />
```

JSP, JavaBean and Servlet Example – TaxServlet.java

```
public class TaxServlet extends HttpServlet {

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException,
        IOException {

        //get the tax bean from the request
        TaxBean taxBean = (TaxBean)req.getAttribute("taxBean");

        //perform the calculation
        double net = calculateNet(taxBean.getGross(), taxBean.getTaxPercentage());
        taxBean.setNet(net);

        //set the updated tax bean in the request
        req.setAttribute("taxBean", taxBean);

        //sent to the result page
        RequestDispatcher rd = req.getRequestDispatcher("tax.jsp");
        rd.forward(req, resp);
    }

    /* Business logic for calculating the net value */
    private double calculateNet(double gross, double taxPercentage) {
        return (gross / (100 + taxPercentage)) * 100;
    }
}
```

- JSP, JavaBean and Servlet Example

