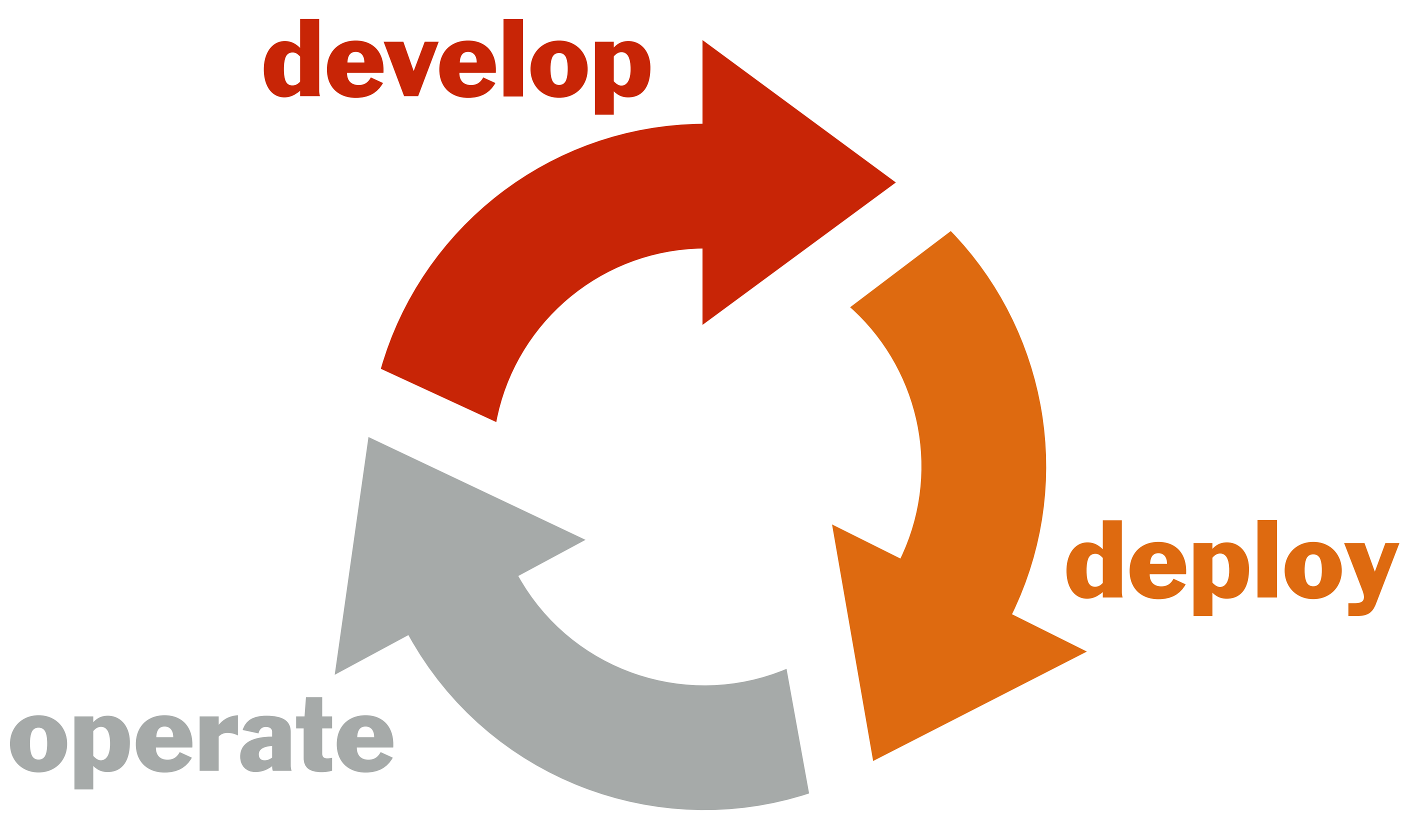
web applications

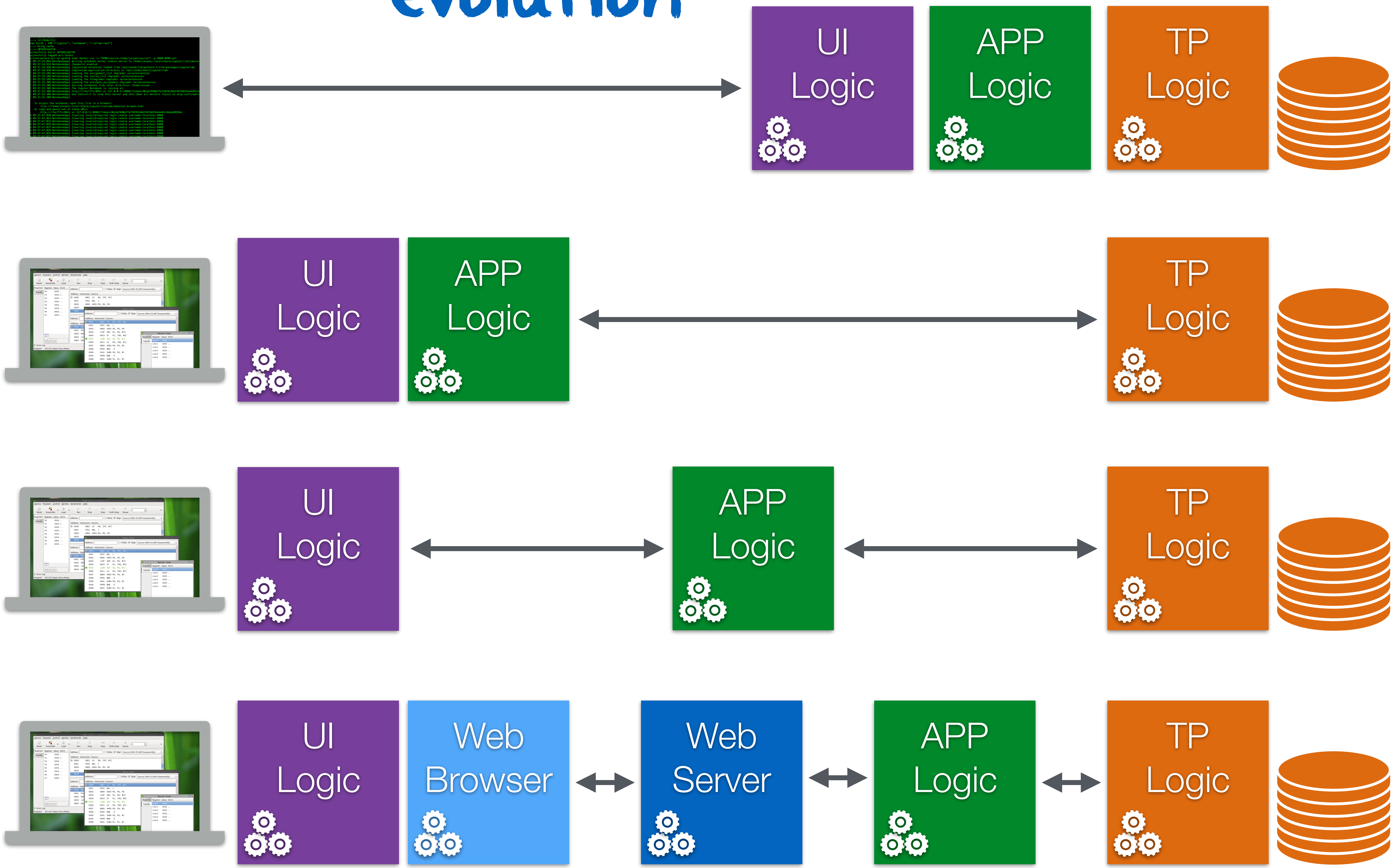
learning objectives



- learn the basics of web applications
- learn the model-view-controller pattern
- learn about the java ee web tier technologies

distributed architecture

evolution



mainframe
70's

client/server
80's

three-tier
early 90's

web-centric
late 90's

UI = User Interface

APP = Application

TP = Transaction Processing

why the web?

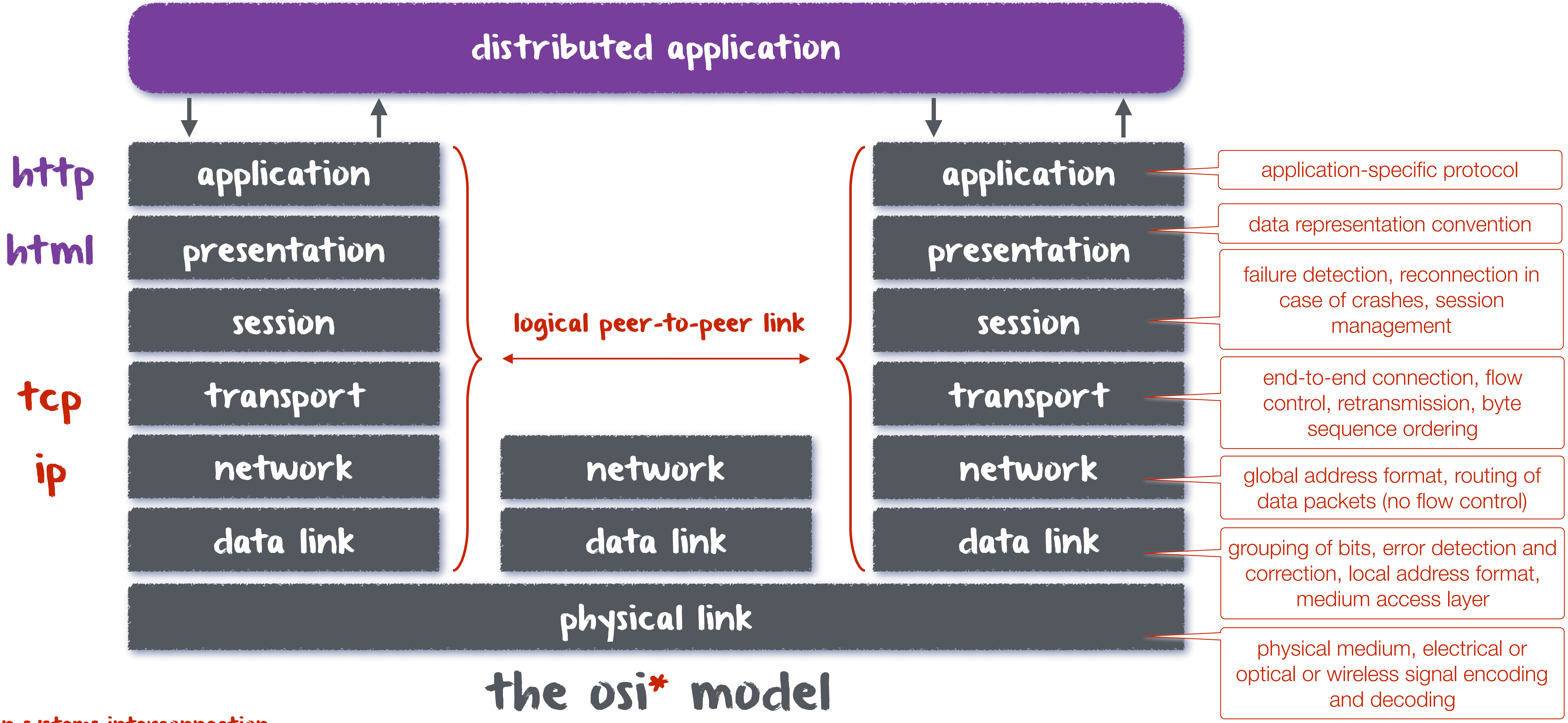
the internet became popular thanks to the world wide web

a web browser is a graphical terminal

a web client displays graphical orders sent by a web server

so there is no need to install dedicated rich clients for different applications

what is the web?



distributed application
http
html
tcp
ip
application
presentation
session
transport
network
data link
logical peer-to-peer link
network
data link
application
presentation
session
transport
network
data link
physical link
application-specific protocol
data representation convention
failure detection, reconnection in case of crashes, session management
end-to-end connection, flow control, retransmission, byte sequence ordering
global address format, routing of data packets (no flow control)
grouping of bits, error detection and correction, local address format, medium access layer
physical medium, electrical or optical or wireless signal encoding and decoding

the osi* model

*open systems interconnection

what is the web?

html = hypertext markup language

standard markup language for describing documents to be displayed in a web browser

http = hypertext transfer protocol

standard a request-response protocol used by web browsers and web servers to communicate

the web

tcp = transmission control protocol

standard internet protocol for transporting byte streams in a reliable and fifo manner

ip = internet protocol

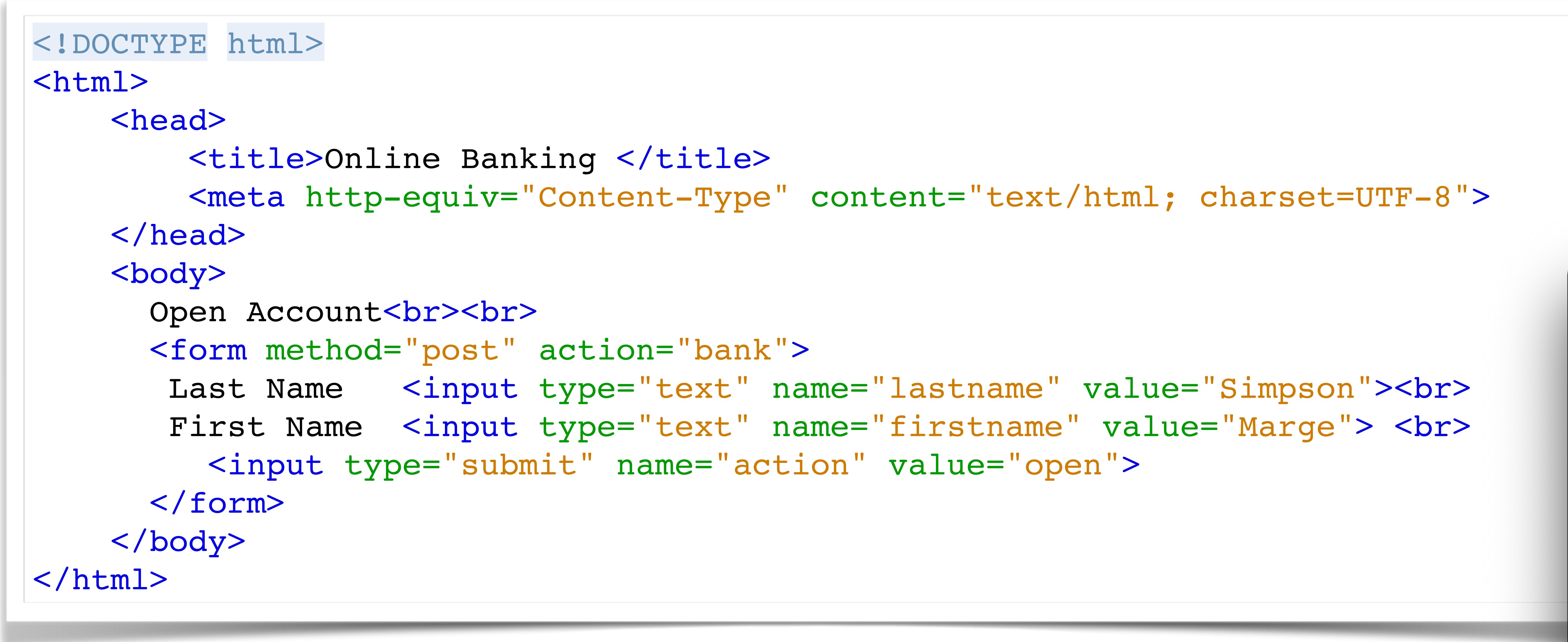
standard internet protocol for routing byte packets across subnets in a best-effort manner

the internet

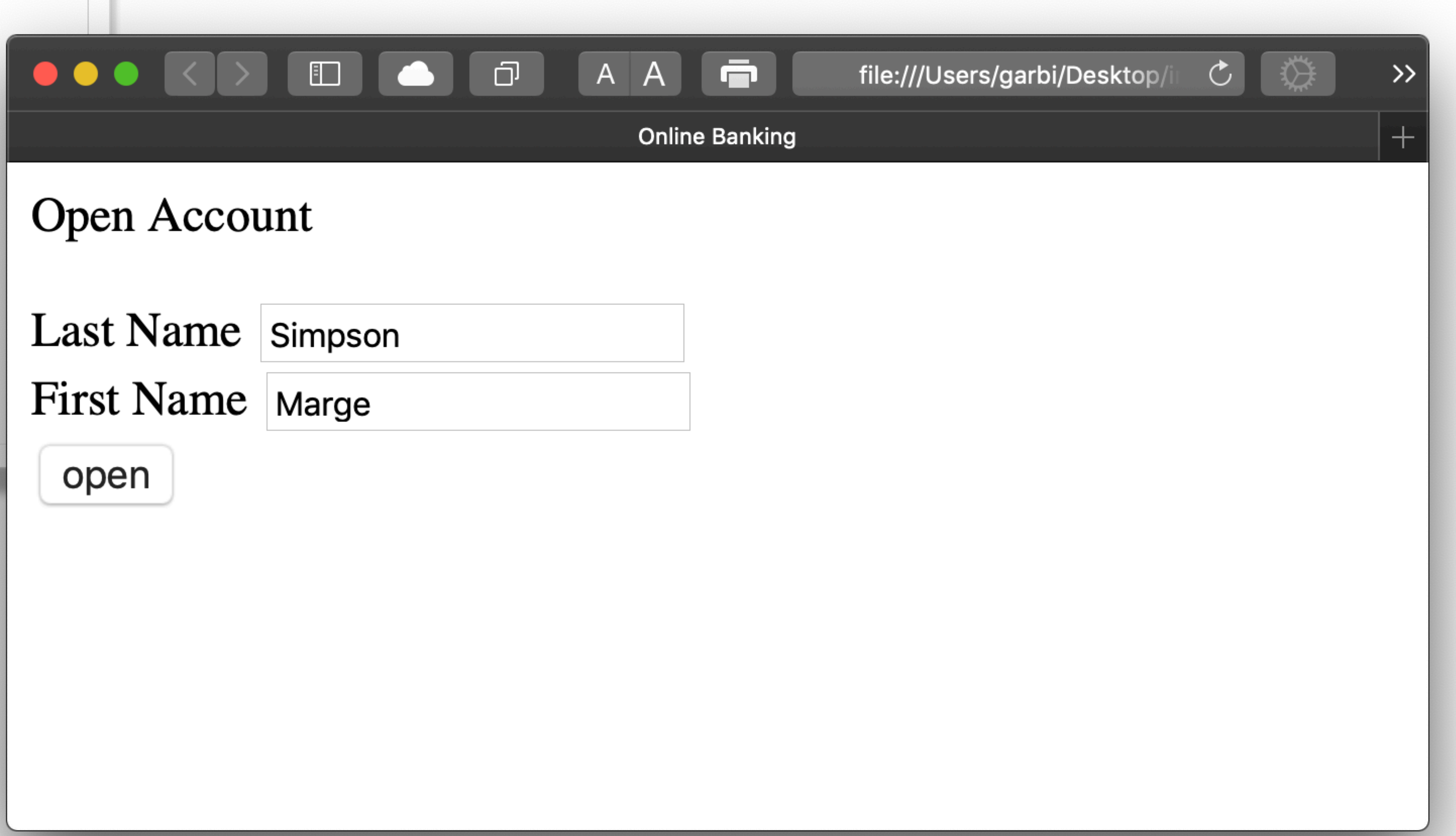
what is the web?

html = hypertext markup language

standard markup language for describing documents to be displayed in a web browser



```
<!DOCTYPE html>
<html>
    <head>
        <title>Online Banking </title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    </head>
    <body>
      Open Account<br><br>
      <form method="post" action="bank">
       Last Name   <input type="text" name="lastname" value="Simpson"><br>
       First Name  <input type="text" name="firstname" value="Marge"> <br>
         <input type="submit" name="action" value="open">
      </form>
    </body>
</html>
```



Online Banking

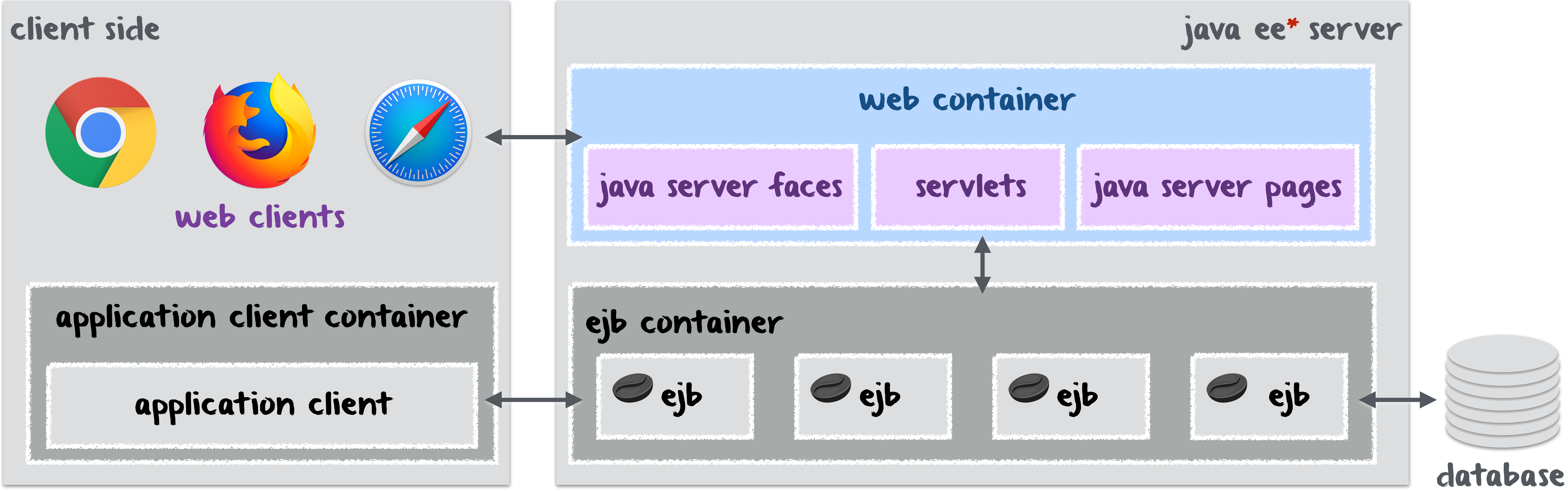
Open Account

Last Name Simpson

First Name Marge

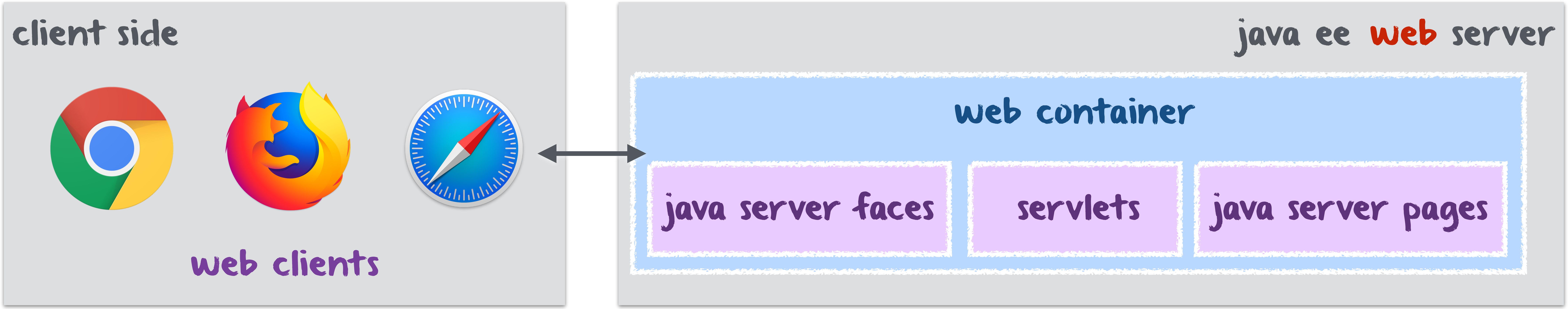
open

web applications in java



*java enterprise edition

what are servlets?

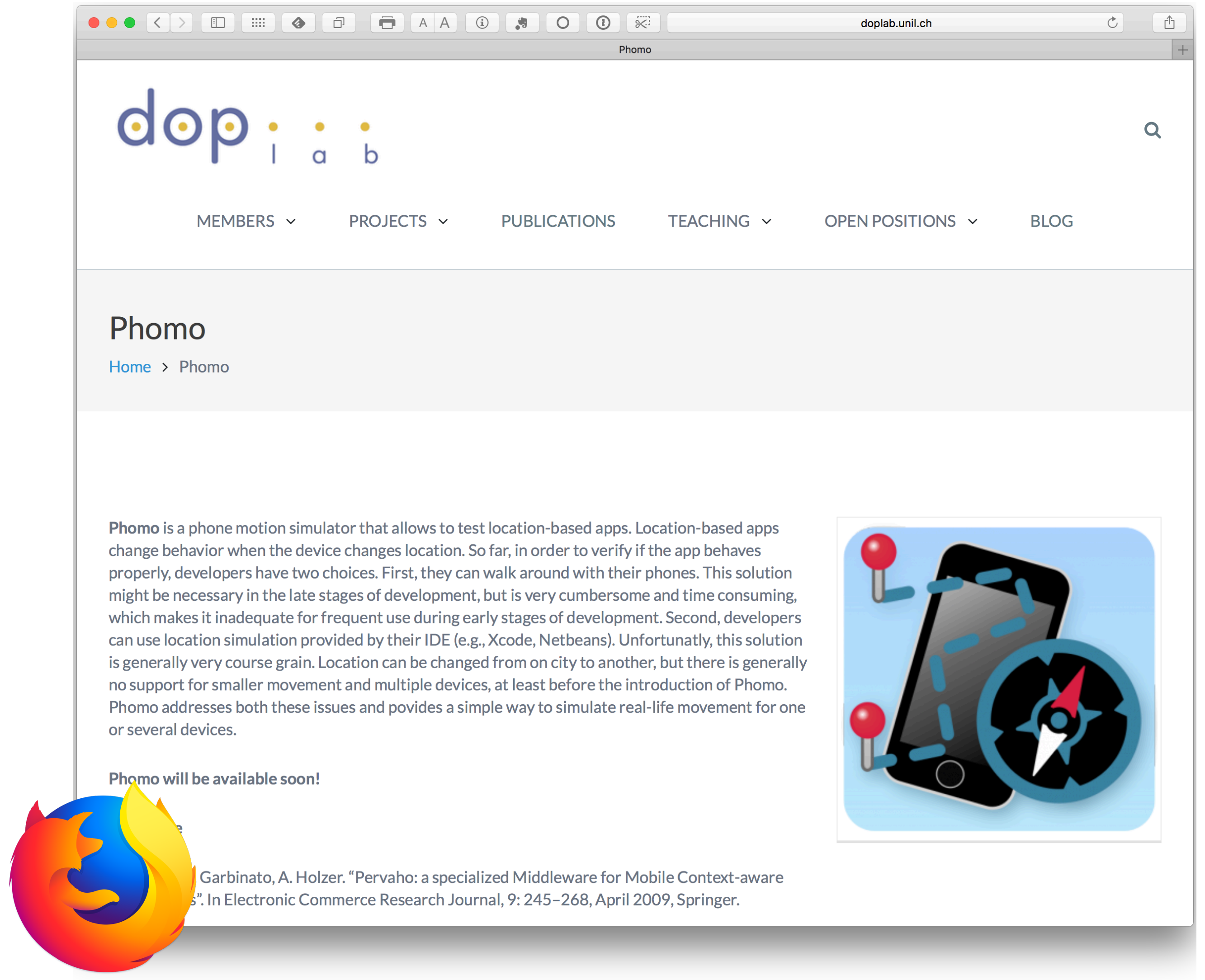


they dynamically generate responses to requests sent by web clients

they are container-managed web server components

servlet lifecycle

upon first request...



doplab.unil.ch

Phomo

MEMBERS PROJECTS PUBLICATIONS TEACHING OPEN POSITIONS BLOG

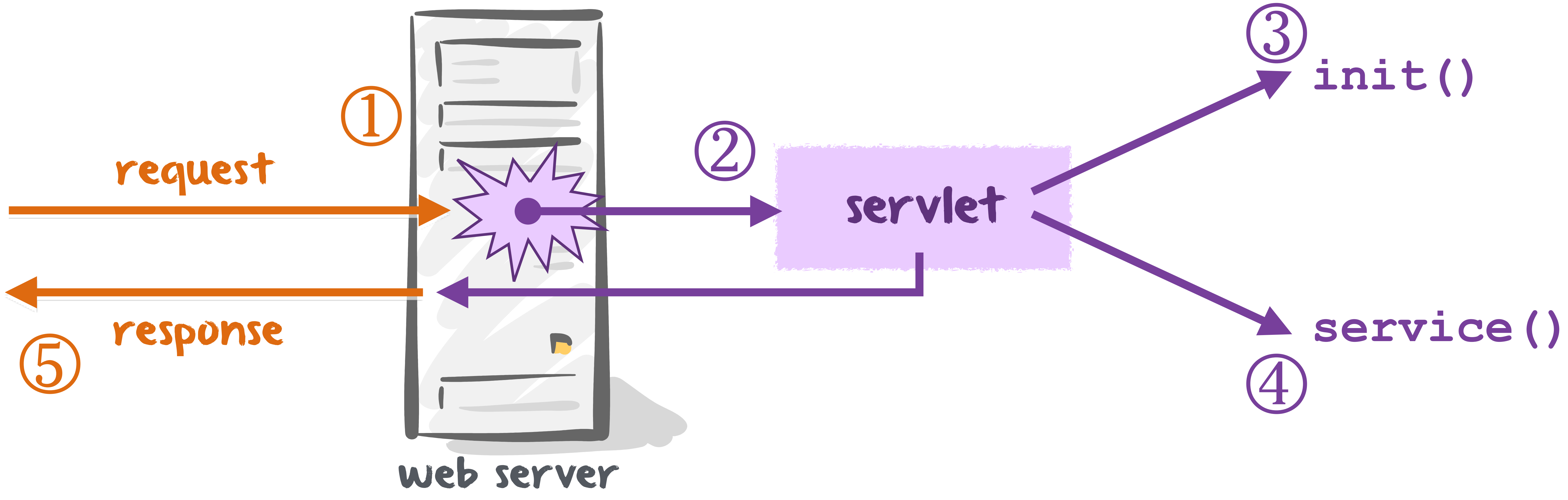
Phomo

Home › Phomo

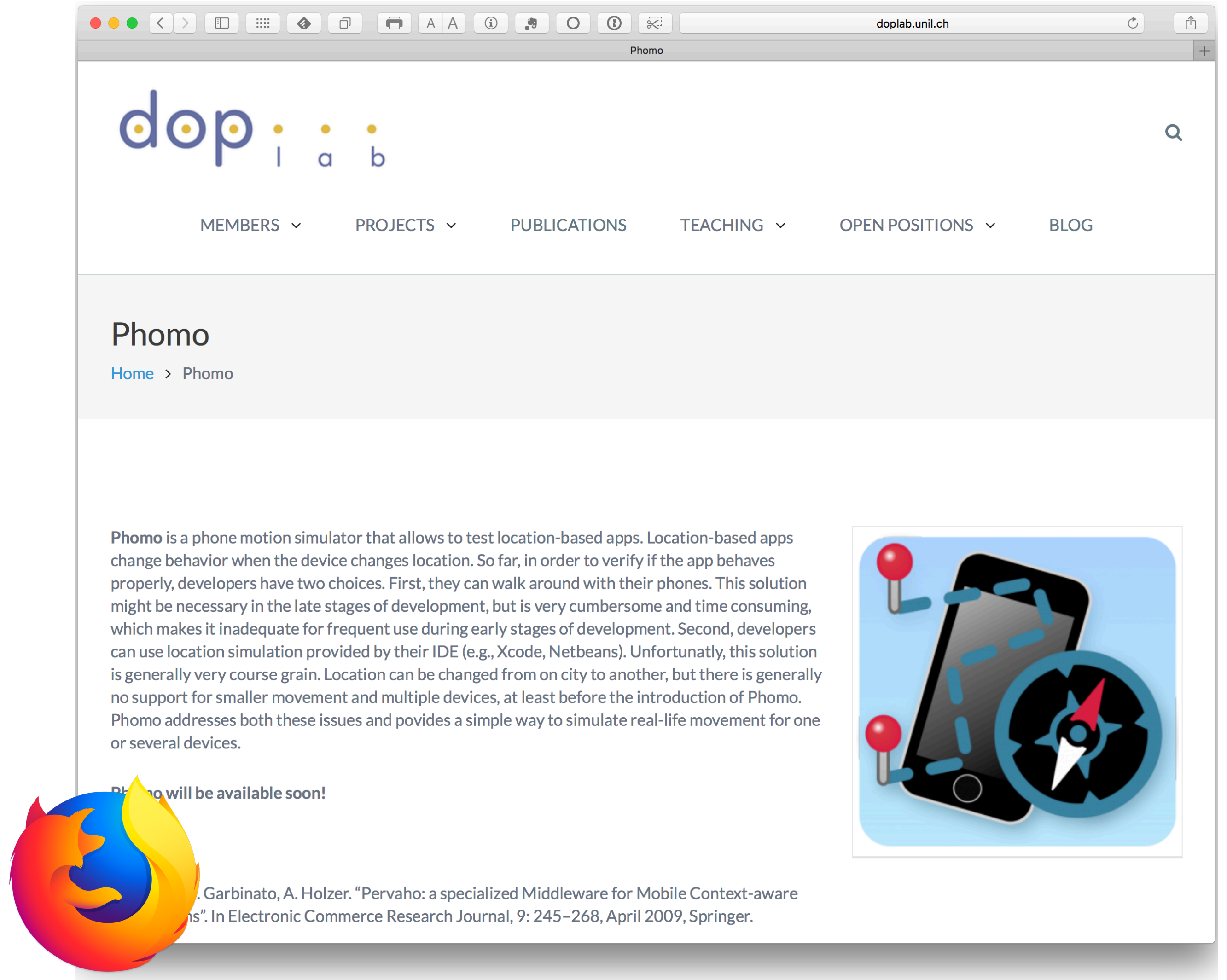
Phomo is a phone motion simulator that allows to test location-based apps. Location-based apps change behavior when the device changes location. So far, in order to verify if the app behaves properly, developers have two choices. First, they can walk around with their phones. This solution might be necessary in the late stages of development, but is very cumbersome and time consuming, which makes it inadequate for frequent use during early stages of development. Second, developers can use location simulation provided by their IDE (e.g., Xcode, Netbeans). Unfortunatly, this solution is generally very course grain. Location can be changed from on city to another, but there is generally no support for smaller movement and multiple devices, at least before the introduction of Phomo. Phomo addresses both these issues and povides a simple way to simulate real-life movement for one or several devices.

Phomo will be available soon!

Garbinato, A. Holzer. “Pervaho: a specialized Middleware for Mobile Context-aware s”. In Electronic Commerce Research Journal, 9: 245–268, April 2009, Springer.



upon subsequent requests...



doplab.unil.ch

Phomo

MEMBERS PROJECTS PUBLICATIONS TEACHING OPEN POSITIONS BLOG

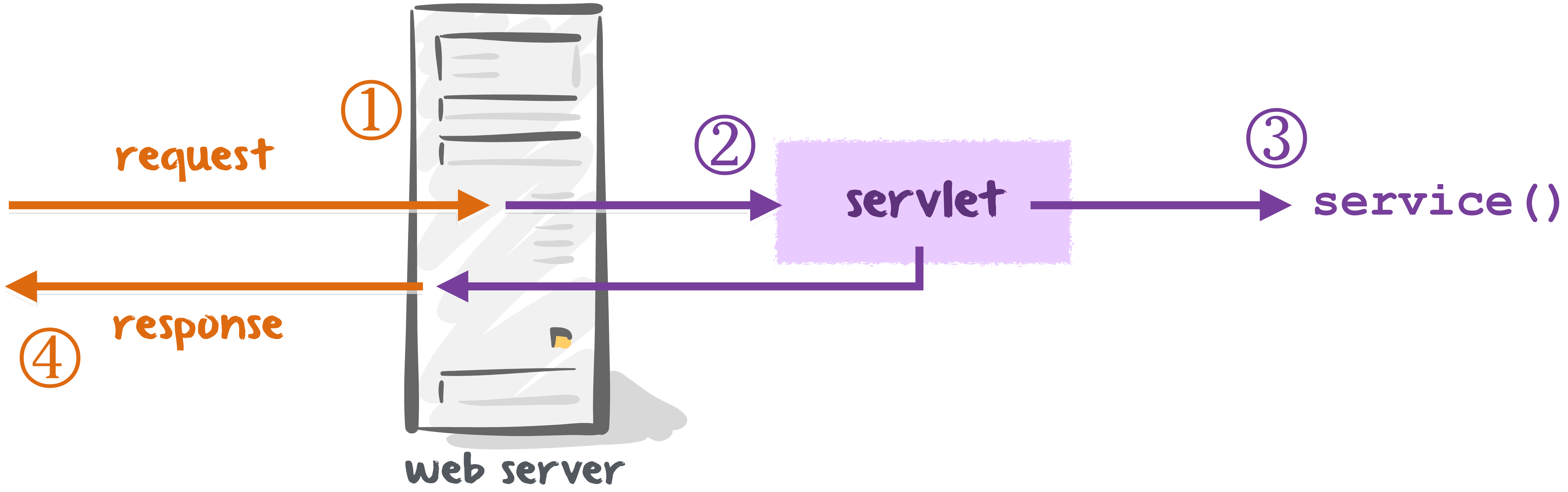
Phomo

Home › Phomo

Phomo is a phone motion simulator that allows to test location-based apps. Location-based apps change behavior when the device changes location. So far, in order to verify if the app behaves properly, developers have two choices. First, they can walk around with their phones. This solution might be necessary in the late stages of development, but is very cumbersome and time consuming, which makes it inadequate for frequent use during early stages of development. Second, developers can use location simulation provided by their IDE (e.g., Xcode, Netbeans). Unfortunatly, this solution is generally very course grain. Location can be changed from on city to another, but there is generally no support for smaller movement and multiple devices, at least before the introduction of Phomo. Phomo addresses both these issues and povides a simple way to simulate real-life movement for one or several devices.

Ph o will be available soon!

Garbinato, A. Holzer. “Pervaho: a specialized Middleware for Mobile Context-aware s”. In Electronic Commerce Research Journal, 9: 245–268, April 2009, Springer.



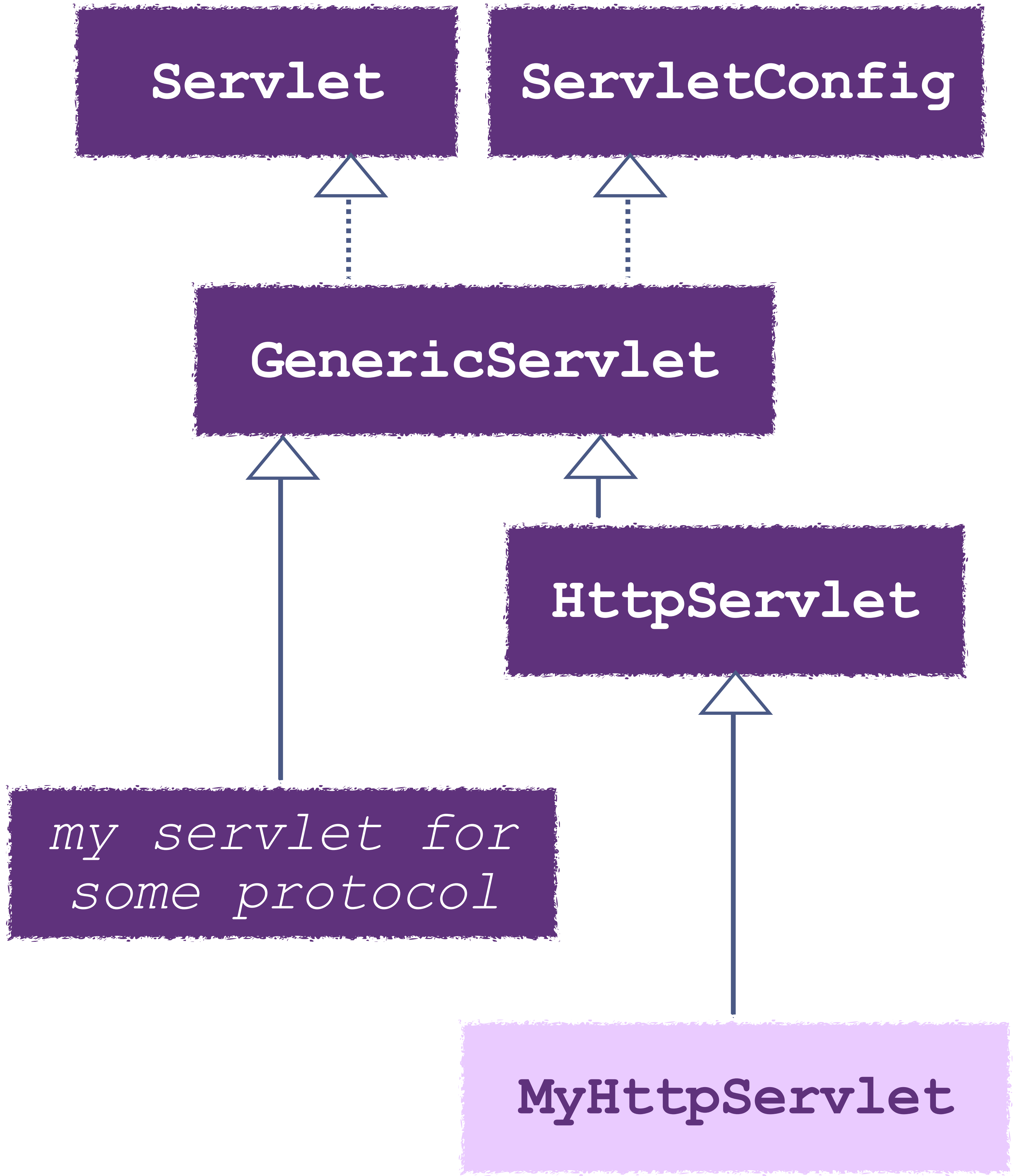
servlet lifecycle & hierarchy

lifecycle management

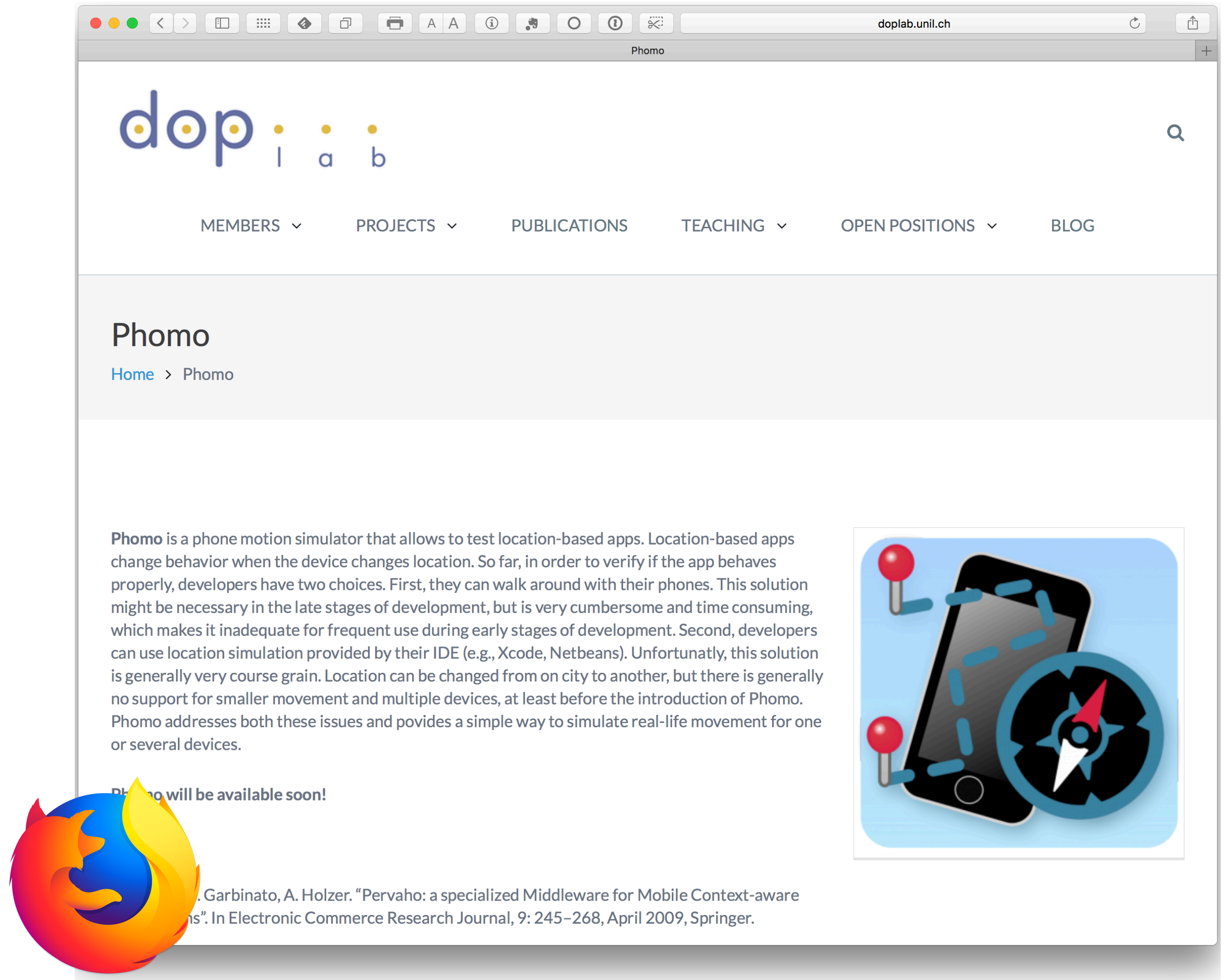
`init()`	performs one-time setup and configuration
`service()`	carries out a single request from the client
`destroy()`	destroys a servlet and its resources

typical callbacks

`doGet()`	request data from the server
`doPost()`	modify data on the server



what do you write?



doplab.unil.ch

Phomo

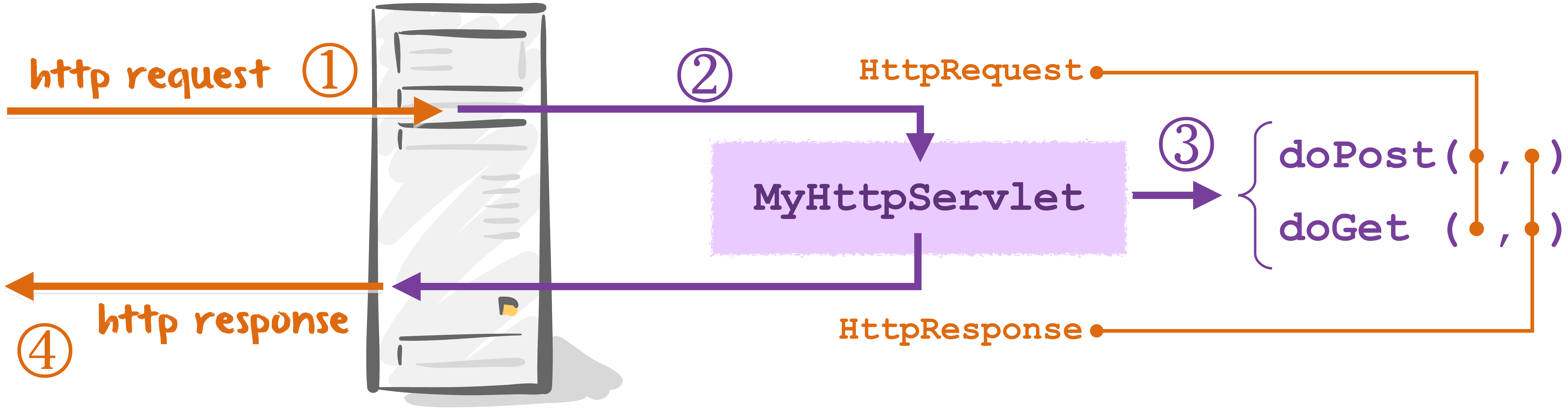
Phomo

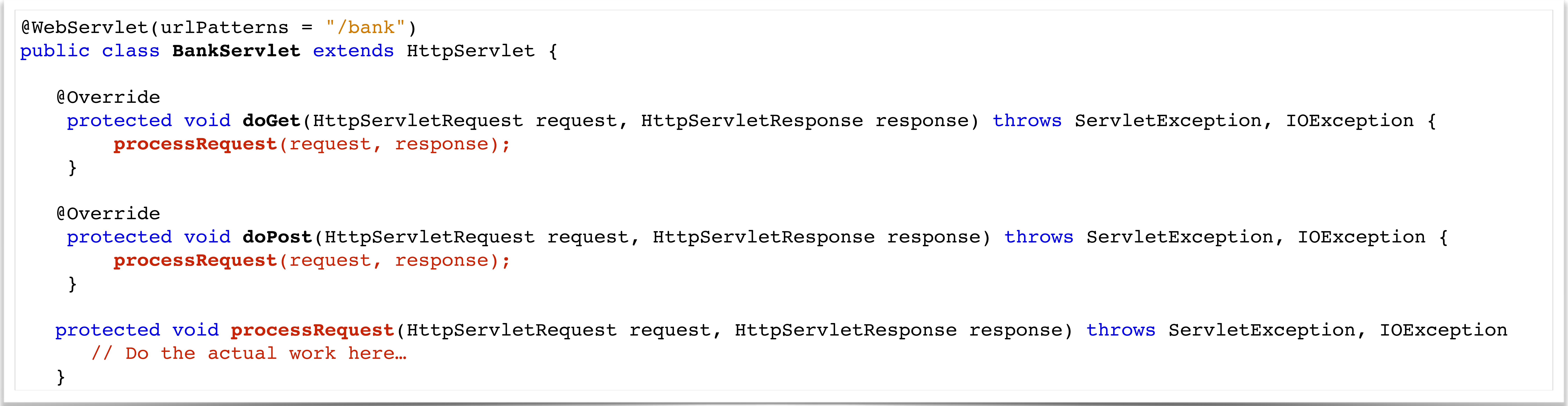
Home › Phomo

Phomo is a phone motion simulator that allows to test location-based apps. Location-based apps change behavior when the device changes location. So far, in order to verify if the app behaves properly, developers have two choices. First, they can walk around with their phones. This solution might be necessary in the late stages of development, but is very cumbersome and time consuming, which makes it inadequate for frequent use during early stages of development. Second, developers can use location simulation provided by their IDE (e.g., Xcode, Netbeans). Unfortunatly, this solution is generally very course grain. Location can be changed from on city to another, but there is generally no support for smaller movement and multiple devices, at least before the introduction of Phomo. Phomo addresses both these issues and povides a simple way to simulate real-life movement for one or several devices.

Phomo will be available soon!

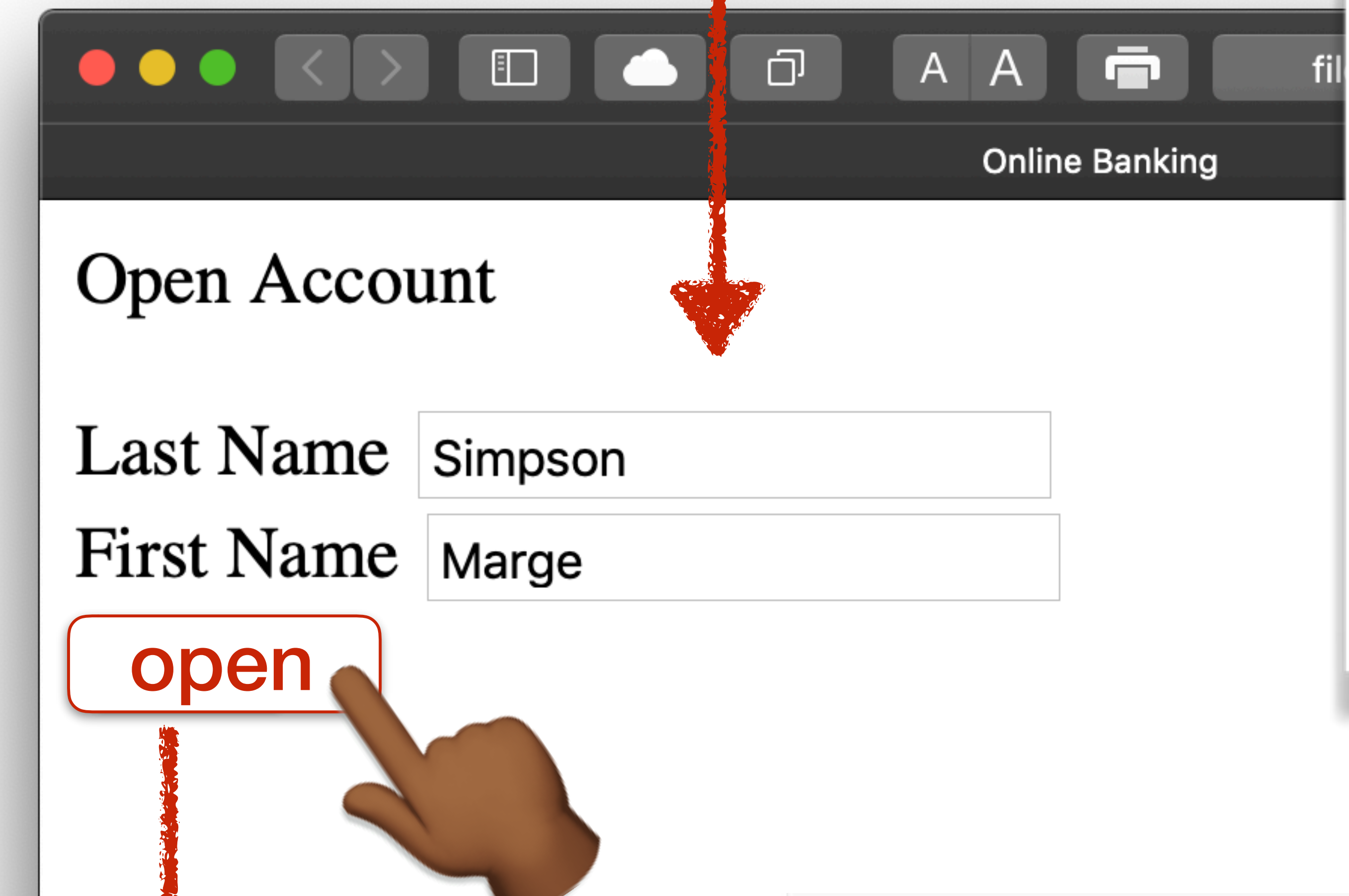
. Garbinato, A. Holzer. “Pervaho: a specialized Middleware for Mobile Context-aware ns”. In Electronic Commerce Research Journal, 9: 245–268, April 2009, Springer.





```
@WebServlet(urlPatterns = "/bank")
public class BankServlet extends HttpServlet {

   @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        processRequest(request, response);
    }

   @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        processRequest(request, response);
    }

   protected void processRequest(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException
      // Do the actual work here…
   }
```

how to retrieve parameters?

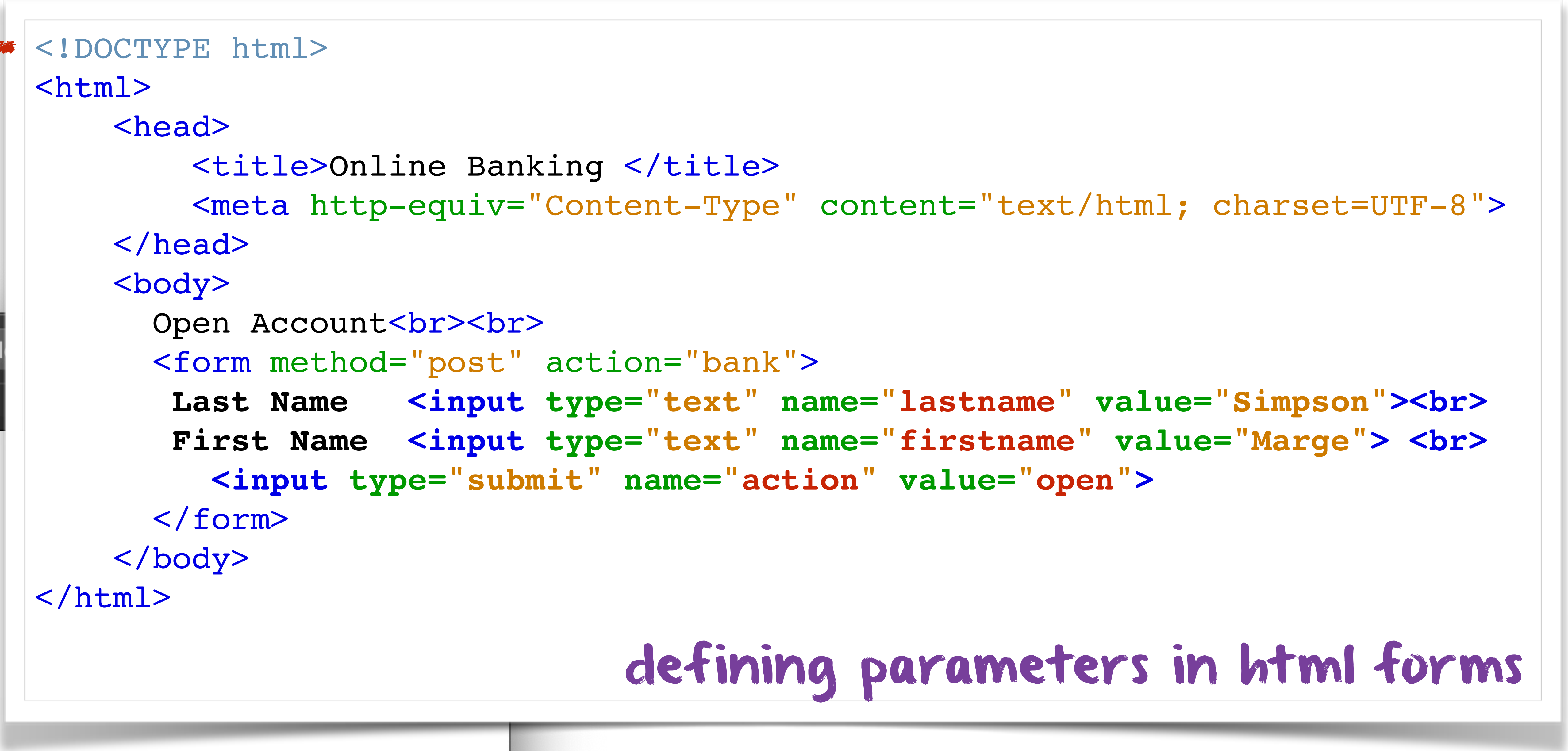


Online Banking

Open Account

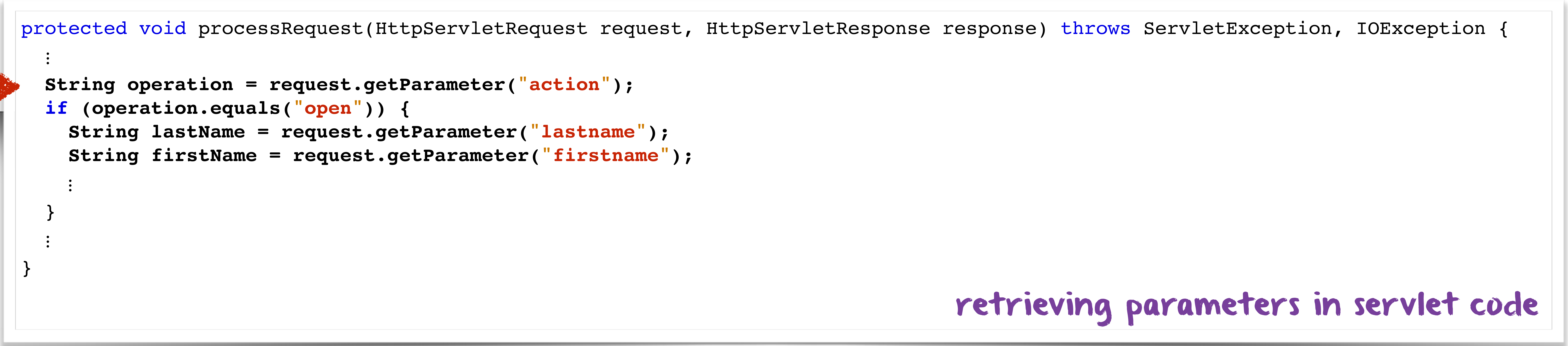
Last Name Simpson

First Name Marge



```
<!DOCTYPE html>
<html>
    <head>
        <title>Online Banking </title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    </head>
    <body>
      Open Account<br><br>
      <form method="post" action="bank">
       Last Name   <input type="text" name="lastname" value="Simpson"><br>
       First Name  <input type="text" name="firstname" value="Marge"> <br>
         <input type="submit" name="action" value="open">
      </form>
    </body>
</html>
```

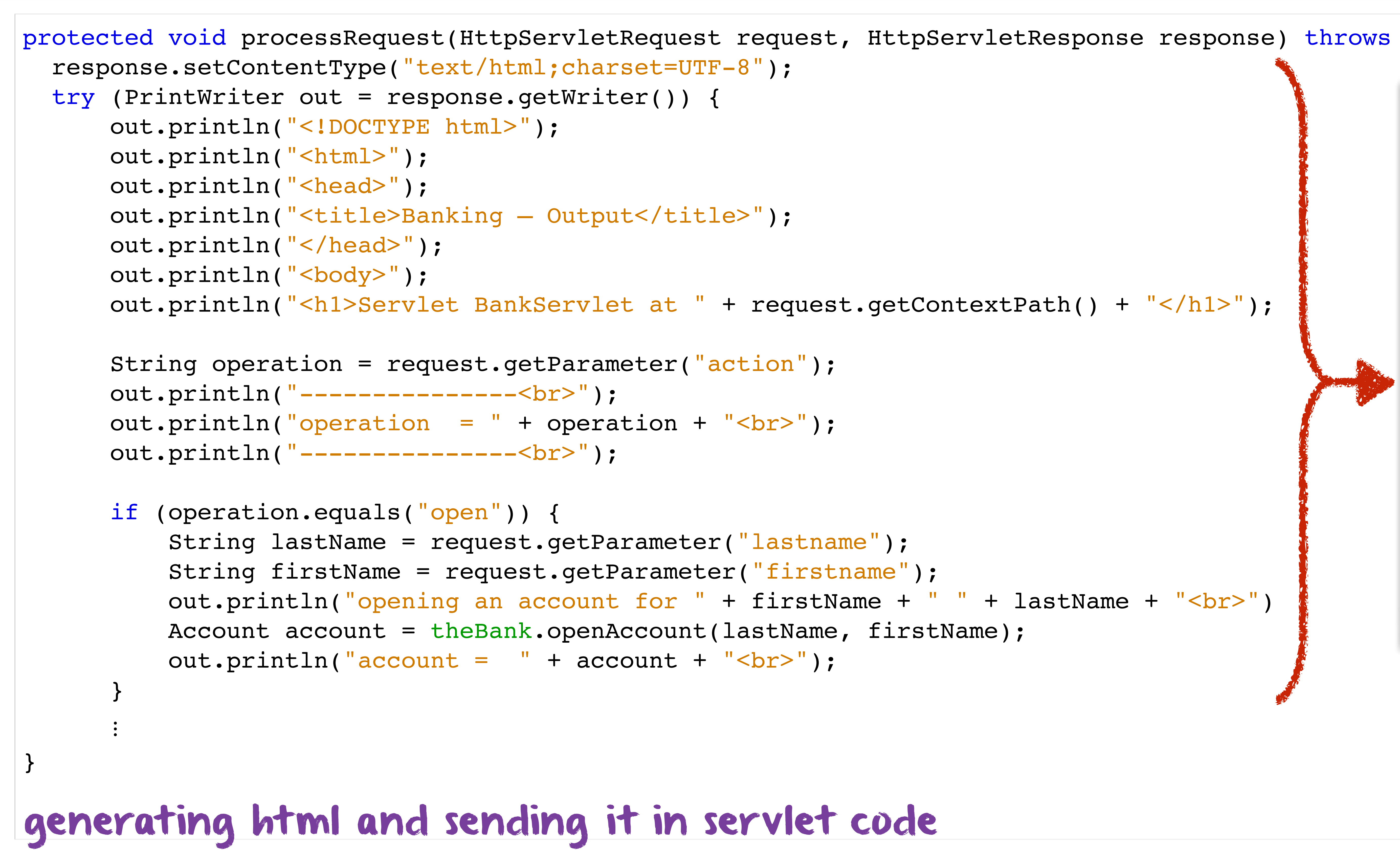
defining parameters in html forms



```
protected void processRequest(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
  ⋮
  String operation = request.getParameter("action");
  if (operation.equals("open")) {
    String lastName = request.getParameter("lastname");
    String firstName = request.getParameter("firstname");
    ⋮
  }
  ⋮
}
```

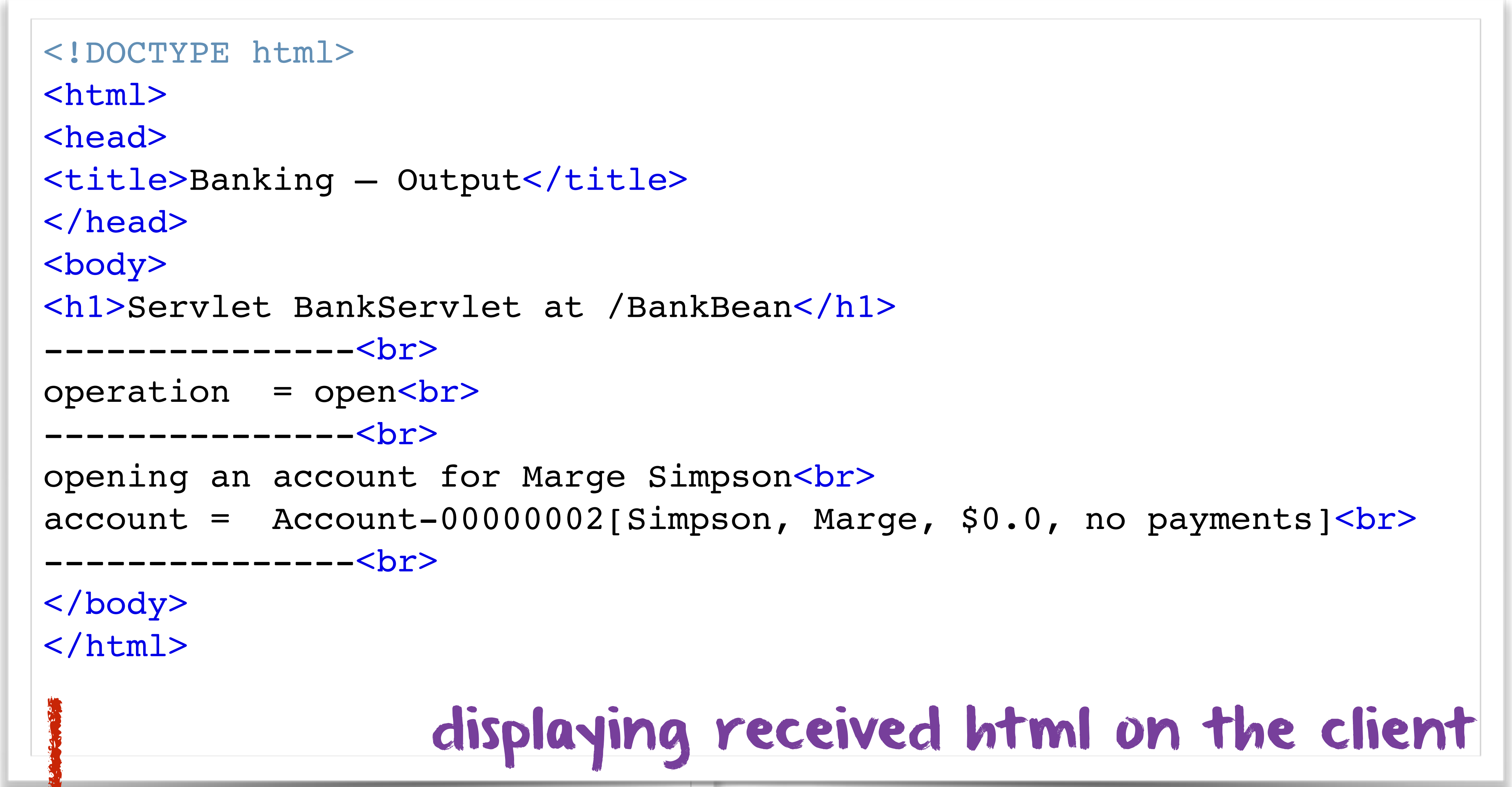
retrieving parameters in servlet code

how to generate a response?



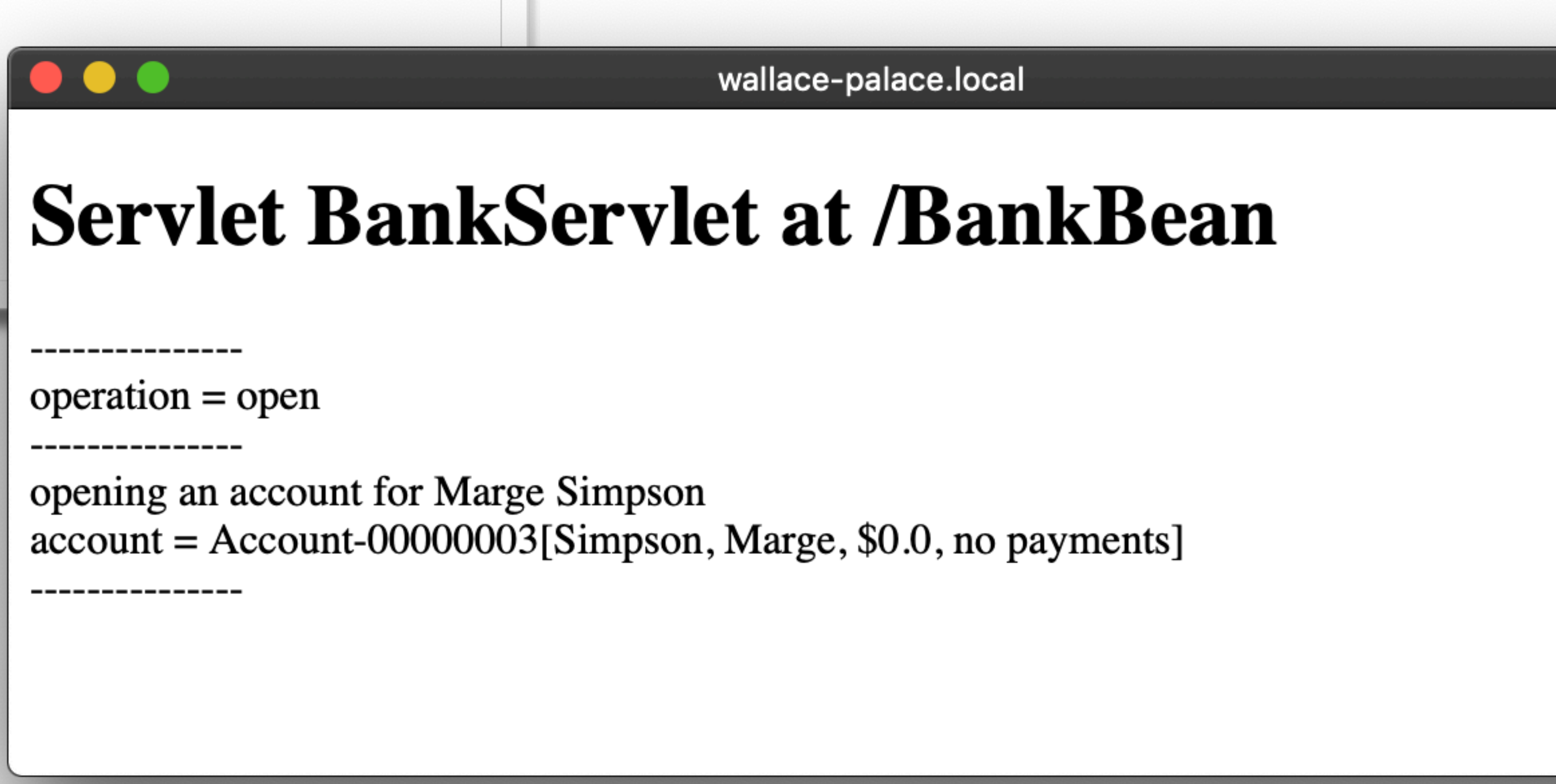
```
protected void processRequest(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
  response.setContentType("text/html;charset=UTF-8");
  try (PrintWriter out = response.getWriter()) {
      out.println("<!DOCTYPE html>");
      out.println("<html>");
      out.println("<head>");
      out.println("<title>Banking – Output</title>");
      out.println("</head>");
      out.println("<body>");
      out.println("<h1>Servlet BankServlet at " + request.getContextPath() + "</h1>");

      String operation = request.getParameter("action");
      out.println("---------------<br>");
      out.println("operation  = " + operation + "<br>");
      out.println("---------------<br>");

      if (operation.equals("open")) {
          String lastName = request.getParameter("lastname");
          String firstName = request.getParameter("firstname");
          out.println("opening an account for " + firstName + " " + lastName + "<br>")
          Account account = theBank.openAccount(lastName, firstName);
          out.println("account =  " + account + "<br>");
      }
      ⋮
}
```

generating html and sending it in servlet code



```
<!DOCTYPE html>
<html>
<head>
<title>Banking – Output</title>
</head>
<body>
<h1>Servlet BankServlet at /BankBean</h1>
---------------<br>
operation  = open<br>
---------------<br>
opening an account for Marge Simpson<br>
account =  Account-00000002[Simpson, Marge, $0.0, no payments]<br>
---------------<br>
</body>
</html>
```

displaying received html on the client



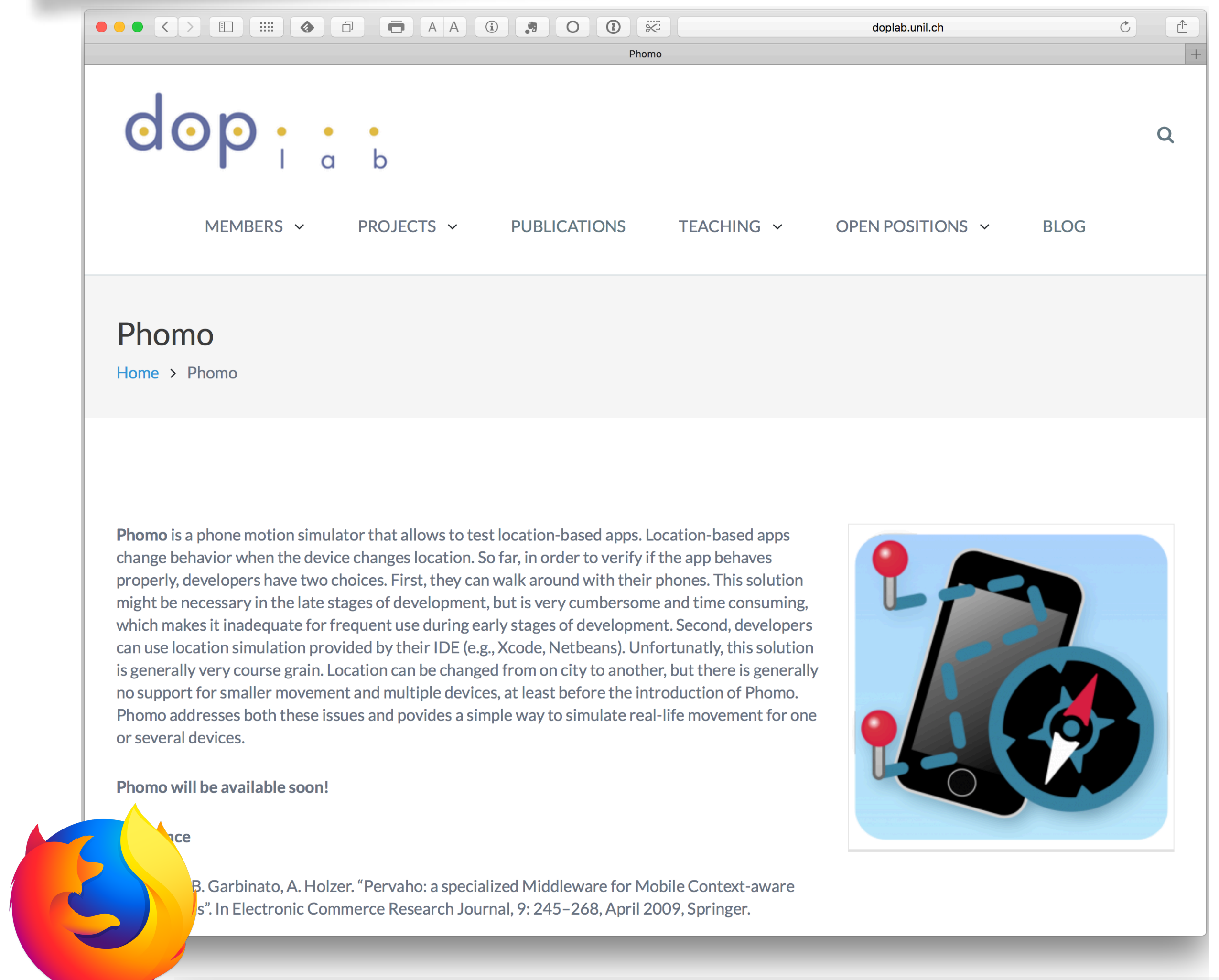
Servlet BankServlet at /BankBean

----
operation = open
----
opening an account for Marge Simpson
account = Account-00000003[Simpson, Marge, $0.0, no payments]
----

session management

http is a stateless protocol but...

...the web container manages sessions transparently



doplab.unil.ch

Phomo

dop l a b

MEMBERS PROJECTS PUBLICATIONS TEACHING OPEN POSITIONS BLOG

Phomo

Home > Phomo

Phomo is a phone motion simulator that allows to test location-based apps. Location-based apps change behavior when the device changes location. So far, in order to verify if the app behaves properly, developers have two choices. First, they can walk around with their phones. This solution might be necessary in the late stages of development, but is very cumbersome and time consuming, which makes it inadequate for frequent use during early stages of development. Second, developers can use location simulation provided by their IDE (e.g., Xcode, Netbeans). Unfortunatly, this solution is generally very course grain. Location can be changed from on city to another, but there is generally no support for smaller movement and multiple devices, at least before the introduction of Phomo. Phomo addresses both these issues and povides a simple way to simulate real-life movement for one or several devices.

Phomo will be available soon!

nce

B. Garbinato, A. Holzer. “Pervaho: a specialized Middleware for Mobile Context-aware s”. In Electronic Commerce Research Journal, 9: 245–268, April 2009, Springer.

doplab.unil.ch

Phomo

dop l a b

MEMBERS PROJECTS PUBLICATIONS TEACHING OPEN POSITIONS BLOG

Phomo

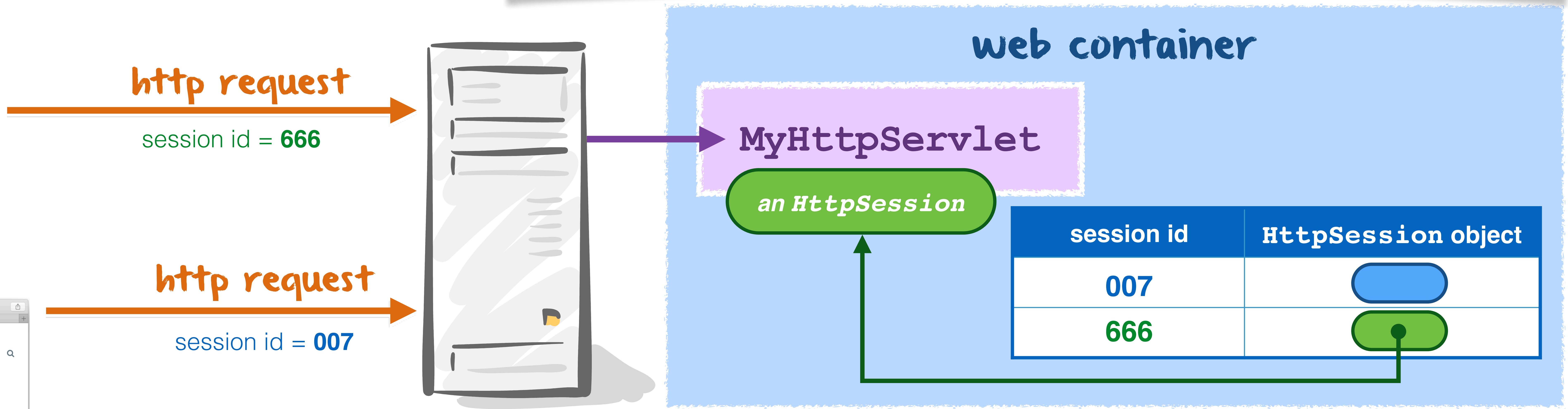
Home > Phomo

Phomo is a phone motion simulator that allows to test location-based apps. Location-based apps change behavior when the device changes location. So far, in order to verify if the app behaves properly, developers have two choices. First, they can walk around with their phones. This solution might be necessary in the late stages of development, but is very cumbersome and time consuming, which makes it inadequate for frequent use during early stages of development. Second, developers can use location simulation provided by their IDE (e.g., Xcode, Netbeans). Unfortunatly, this solution is generally very course grain. Location can be changed from on city to another, but there is generally no support for smaller movement and multiple devices, at least before the introduction of Phomo. Phomo addresses both these issues and povides a simple way to simulate real-life movement for one or several devices.

Phomo will be available soon!

nce

B. Garbinato, A. Holzer. “Pervaho: a specialized Middleware for Mobile Context-aware s”. In Electronic Commerce Research Journal, 9: 245–268, April 2009, Springer.

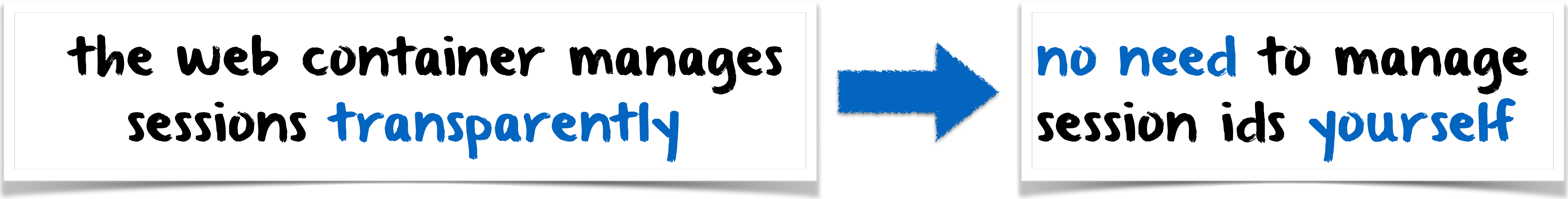


session id	HttpSession object
007	🟦
666	🟩

```
protected void processRequest(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
  :
  HttpSession mySession = request.getSession();
  if (mySession.isNew()) {
    mySession.setAttribute("lastName", lastName);
    mySession.setAttribute("firstName", firstName);
  } else {
    lastName = (String) mySession.getAttribute("lastName");
    firstName = (String) mySession.getAttribute("firstName");
  }
  :
}
```

using the session object in servlet code

session management



the web container manages sessions transparently

no need to manage session ids yourself

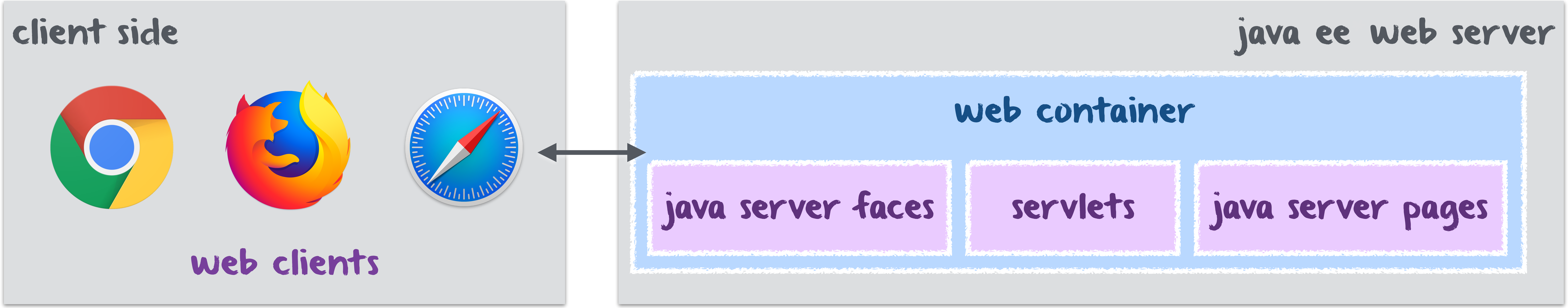
to achieve this, the web container uses various strategies

implicit cookies

url rewriting

hidden fields

what are java server pages?



they dynamically generate responses to requests sent by web clients

they are container-managed web server components

so how do they differ from servlets?

what are java server pages?

so how do they differ from servlets?

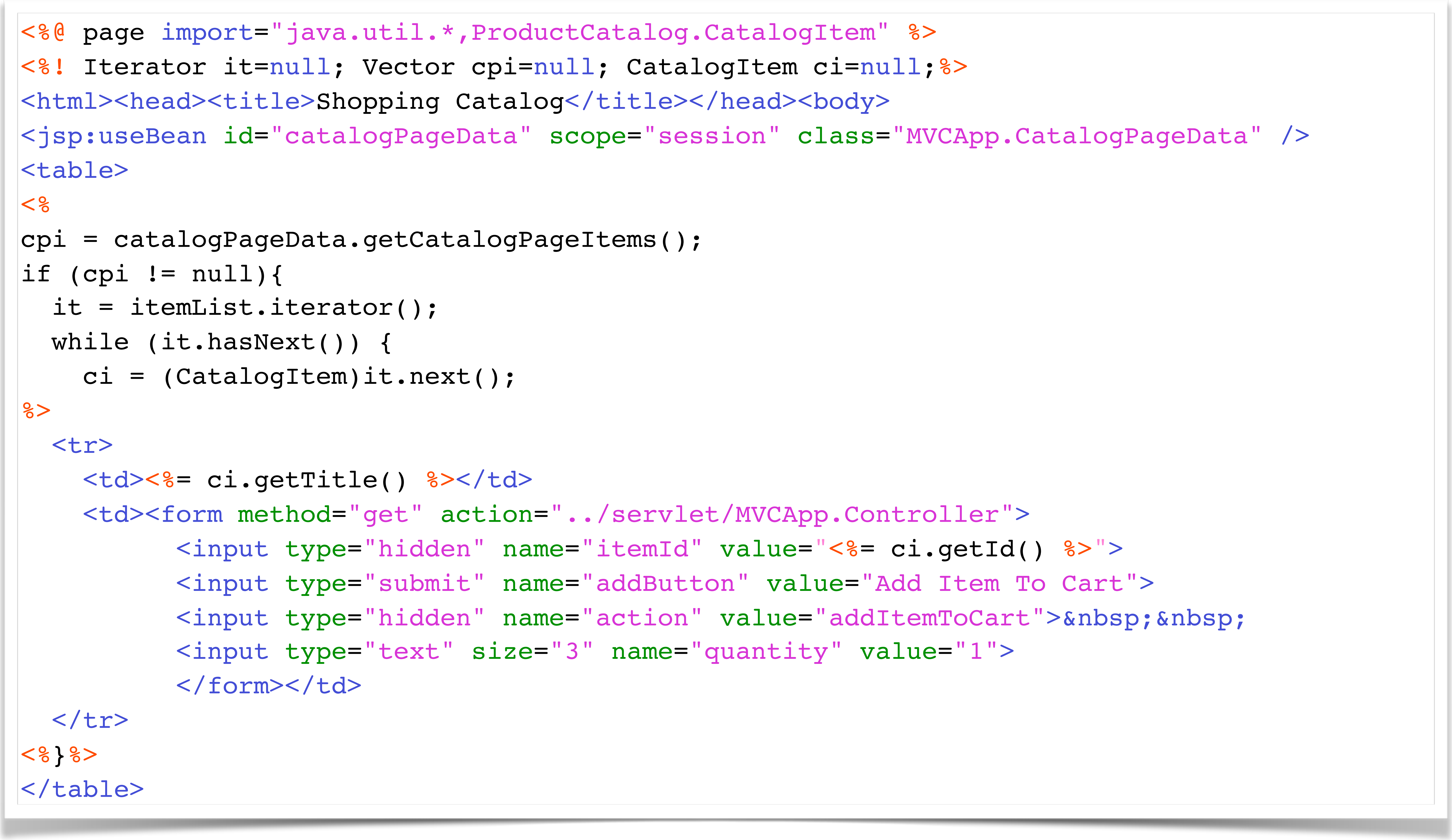
they "reverse" the java-html relationship

they are built on top of servlets

they help web authors & web developers to work jointly on the same pages

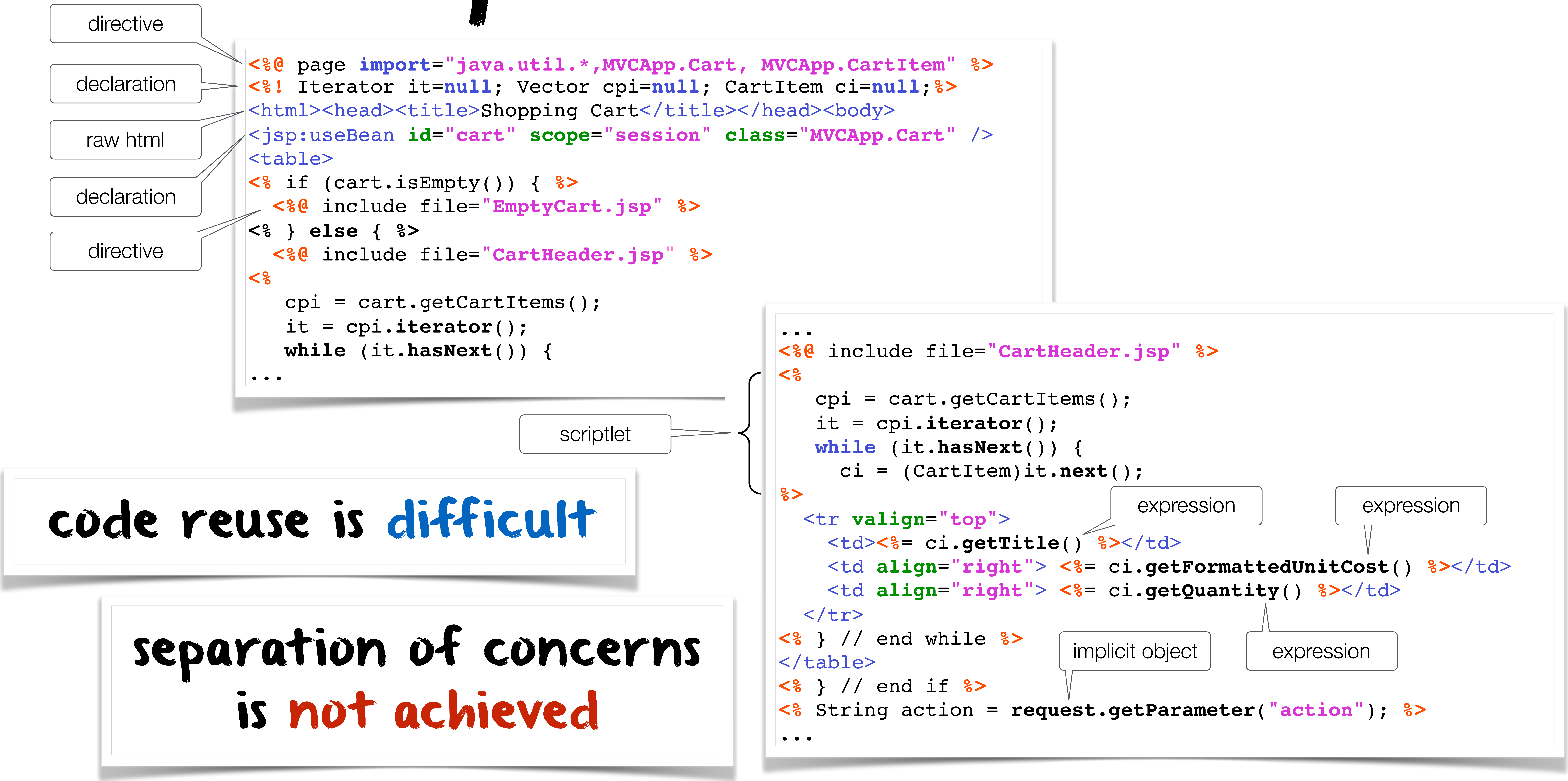
what are java server pages?

syntax overview



```
<%@ page import="java.util.*,ProductCatalog.CatalogItem" %>
<%! Iterator it=null; Vector cpi=null; CatalogItem ci=null;%>
<html><head><title>Shopping Catalog</title></head><body>
<jsp:useBean id="catalogPageData" scope="session" class="MVCApp.CatalogPageData" />
<table>
<%
cpi = catalogPageData.getCatalogPageItems();
if (cpi != null){
  it = itemList.iterator();
  while (it.hasNext()) {
    ci = (CatalogItem)it.next();
%>
  <tr>
    <td><%= ci.getTitle() %></td>
    <td><form method="get" action="../servlet/MVCApp.Controller">
          <input type="hidden" name="itemId" value="<%= ci.getId() %>">
          <input type="submit" name="addButton" value="Add Item To Cart">
          <input type="hidden" name="action" value="addItemToCart">&nbsp;&nbsp;
          <input type="text" size="3" name="quantity" value="1">
          </form></td>
  </tr>
<%}%>
</table>
```

syntax overview



directive

declaration

raw html

declaration

directive

```
<%@ page import="java.util.*,MVCApp.Cart, MVCApp.CartItem" %>
<%! Iterator it=null; Vector cpi=null; CartItem ci=null;%>
<html><head><title>Shopping Cart</title></head><body>
<jsp:useBean id="cart" scope="session" class="MVCApp.Cart" />
<table>
<% if (cart.isEmpty()) { %>
  <%@ include file="EmptyCart.jsp" %>
<% } else { %>
  <%@ include file="CartHeader.jsp" %>
<%
   cpi = cart.getCartItems();
   it = cpi.iterator();
   while (it.hasNext()) {
...
```

scriptlet

```
...
<%@ include file="CartHeader.jsp" %>
<%
   cpi = cart.getCartItems();
   it = cpi.iterator();
   while (it.hasNext()) {
     ci = (CartItem)it.next();
%>
  <tr valign="top">
    <td><%= ci.getTitle() %></td>
    <td align="right"> <%= ci.getFormattedUnitCost() %></td>
    <td align="right"> <%= ci.getQuantity() %></td>
  </tr>
<% } // end while %>
</table>
<% } // end if %>
<% String action = request.getParameter("action"); %>
...
```

expression

expression

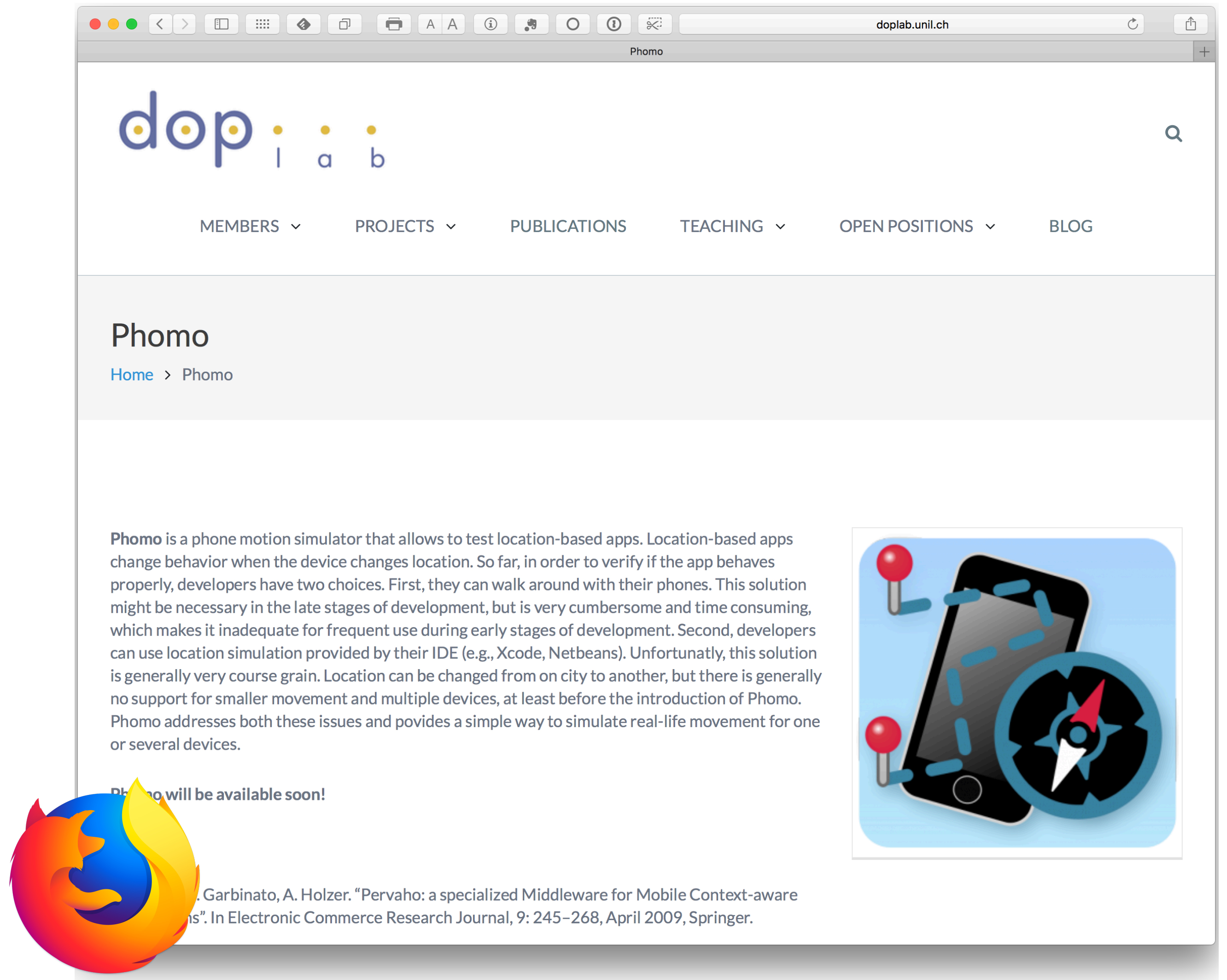
implicit object

expression

code reuse is difficult

separation of concerns is not achieved

java server pages lifecycle



doplab.unil.ch

Phomo

MEMBERS PROJECTS PUBLICATIONS TEACHING OPEN POSITIONS BLOG

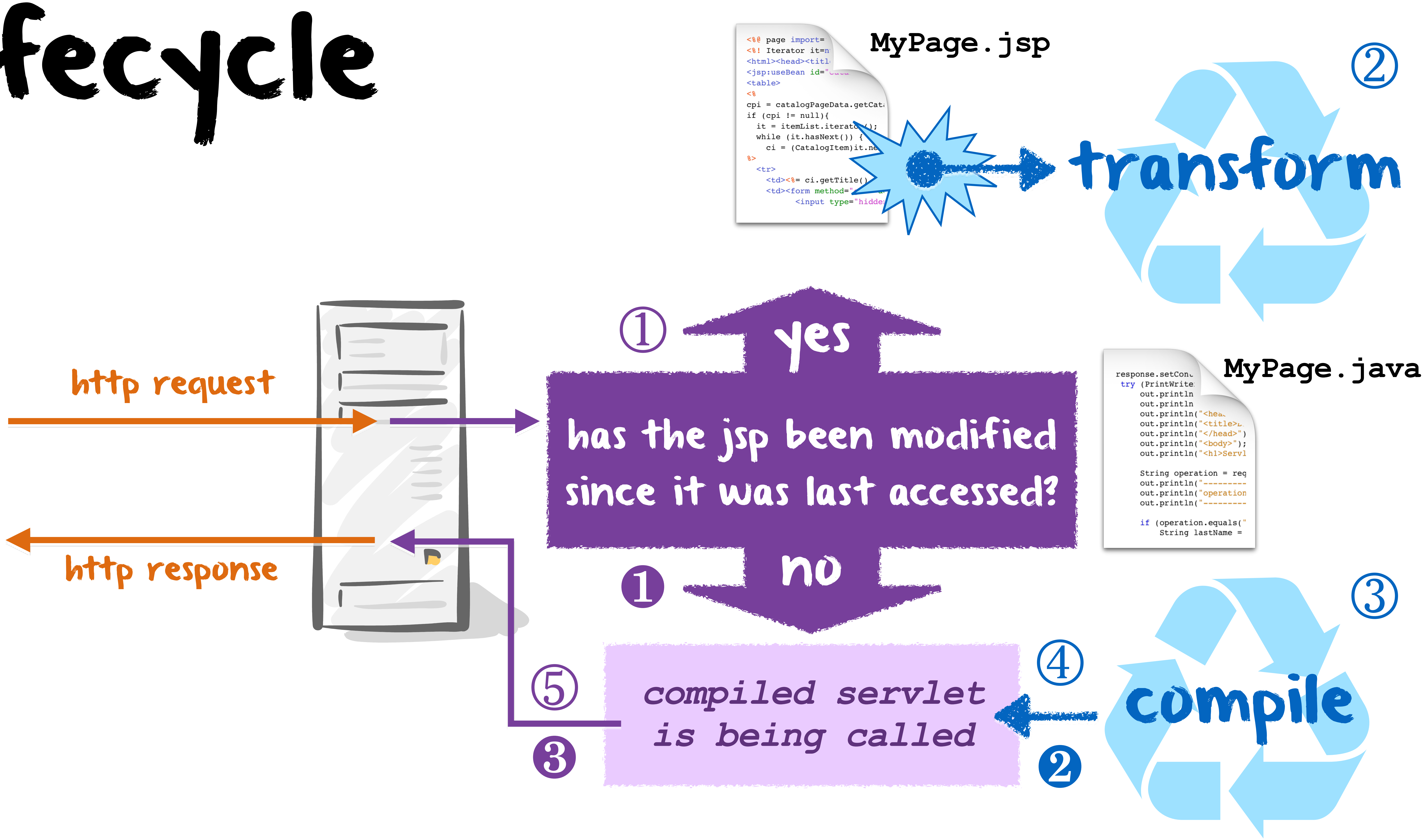
Phomo

Home > Phomo

Phomo is a phone motion simulator that allows to test location-based apps. Location-based apps change behavior when the device changes location. So far, in order to verify if the app behaves properly, developers have two choices. First, they can walk around with their phones. This solution might be necessary in the late stages of development, but is very cumbersome and time consuming, which makes it inadequate for frequent use during early stages of development. Second, developers can use location simulation provided by their IDE (e.g., Xcode, Netbeans). Unfortunatly, this solution is generally very course grain. Location can be changed from on city to another, but there is generally no support for smaller movement and multiple devices, at least before the introduction of Phomo. Phomo addresses both these issues and povides a simple way to simulate real-life movement for one or several devices.

Ph[illegible]o will be available soon!

[illegible]. Garbinato, A. Holzer. “Pervaho: a specialized Middleware for Mobile Context-aware [illegible]ns”. In Electronic Commerce Research Journal, 9: 245–268, April 2009, Springer.


MyPage.jsp
②
transform
http request
①
yes
has the jsp been modified since it was last accessed?
❶
no
MyPage.java
③
④
compile
❷
⑤
compiled servlet is being called
❸
http response

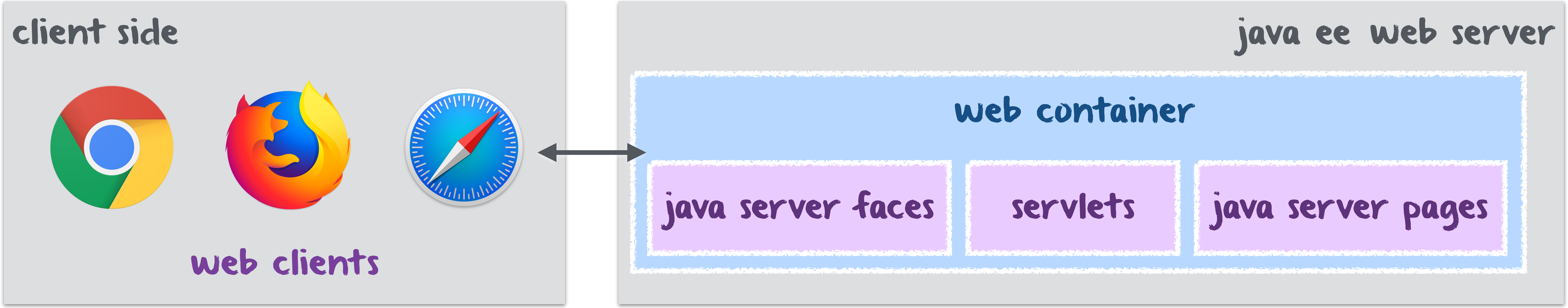
```
<%@ page import=
<%! Iterator it=n
<html><head><titl
<jsp:useBean id="
<table>
<%
cpi = catalogPageData.getCat
if (cpi != null){
  it = itemList.iterato
  while (it.hasNext()) {
    ci = (CatalogItem)it.ne
%>
  <tr>
    <td><%= ci.getTitle()
    <td><form method="
          <input type="hidde
```

```
response.setCon
 try (PrintWrite
     out.println
     out.println
     out.println("<hea
     out.println("<title>
     out.println("</head>")
     out.println("<body>");
     out.println("<h1>Servl

     String operation = req
     out.println("---------
     out.println("operation
     out.println("---------

     if (operation.equals("
         String lastName =
```

① : long cycle

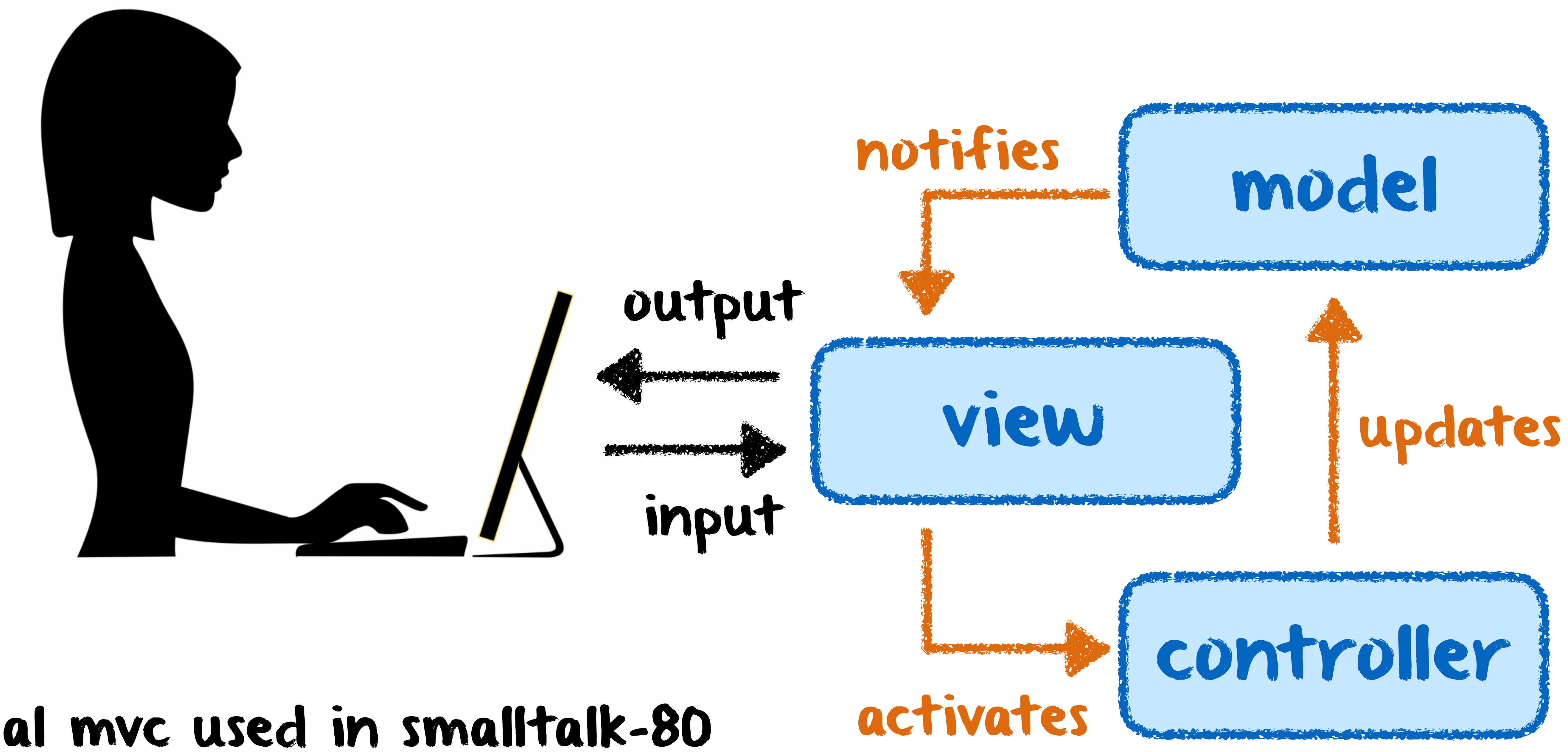
❶ : short cycle

what are java server faces?



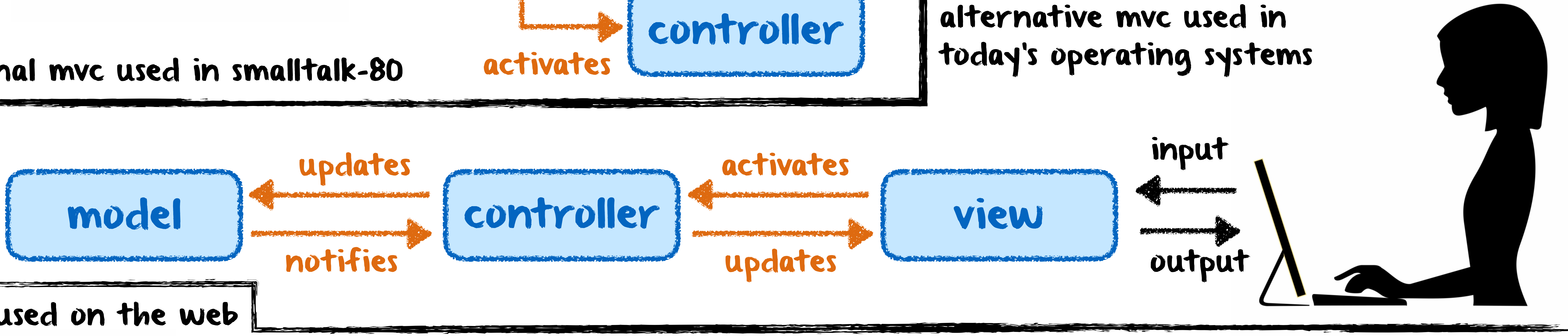
to understand java server faces, we must introduce the model-view-controller design patterns first

model view controller



original mvc used in smalltalk-80

alternative mvc used in today's operating systems



mvc used on the web

wallace-palace.local
output
input
view
response
request
web server
web container
web component
controller
remote call
business logic
ejb
model
transactions
database

Hi, my name is Duke!

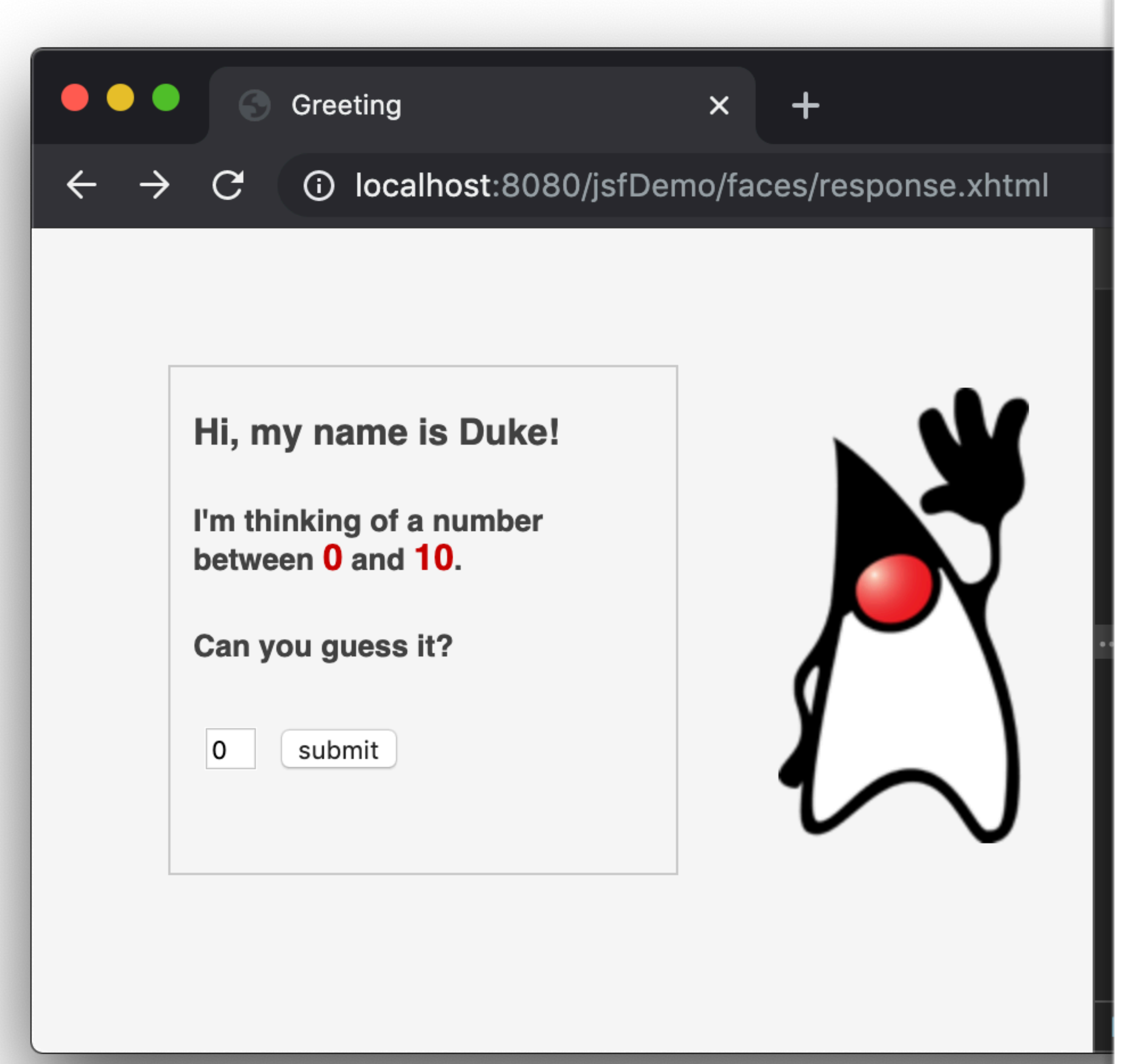
I'm thinking of a number between 0 and 10.

Can you guess it?

0 submit

model-view-controller

output
input
wallace-palace.local
Hi, my name is Duke!
I'm thinking of a number between 0 and 10.
Can you guess it?
0
submit
view
response
request
web server
web container
web component
controller
remote call
business logic
ejb
model
transactions
database



Hi, my name is Duke!

I'm thinking of a number between 0 and 10.

Can you guess it?

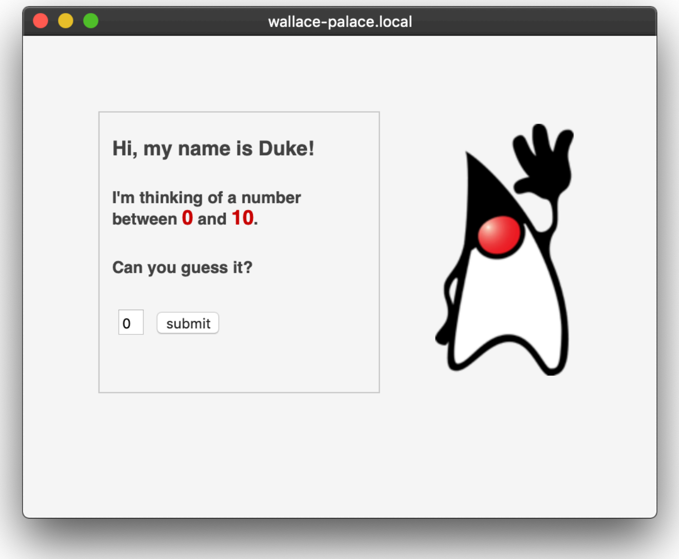
0 submit

html received on the client side

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html">
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
        <link href="css/stylesheet.css" rel="stylesheet" type="text/css" />
        <title>Greeting</title>
    </head>
    <body>
        <div id="mainContainer">
            <div id="left" class="subContainer greyBox">

                <h4>Hi, my name is Duke!</h4>
                <h5>I'm thinking of a number
                    <br/>
                    between
                    <span class="highlight">0</span> and
                    <span class="highlight">10</span>.</h5>
                <h5>Can you guess it?</h5>
                <h:form>
                    <h:inputText id="userNumber" size="2" maxlength="2" value="#{userNumberBean.userNumber}"/>
                    <h:commandButton id="submit" value="submit" action="response"/>
                </h:form>
            </div>
```

model-view-controller

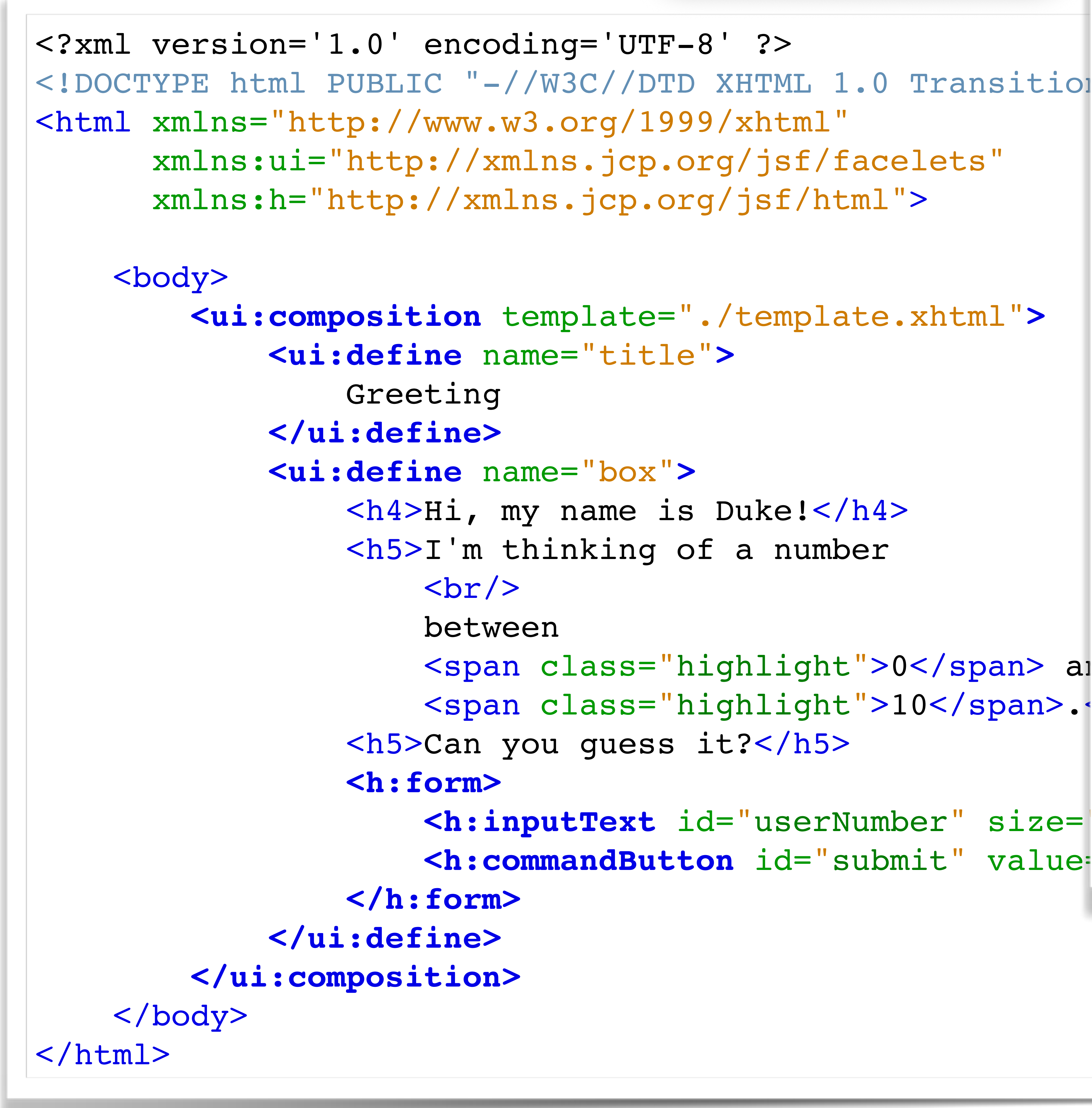
output
input



Hi, my name is Duke!

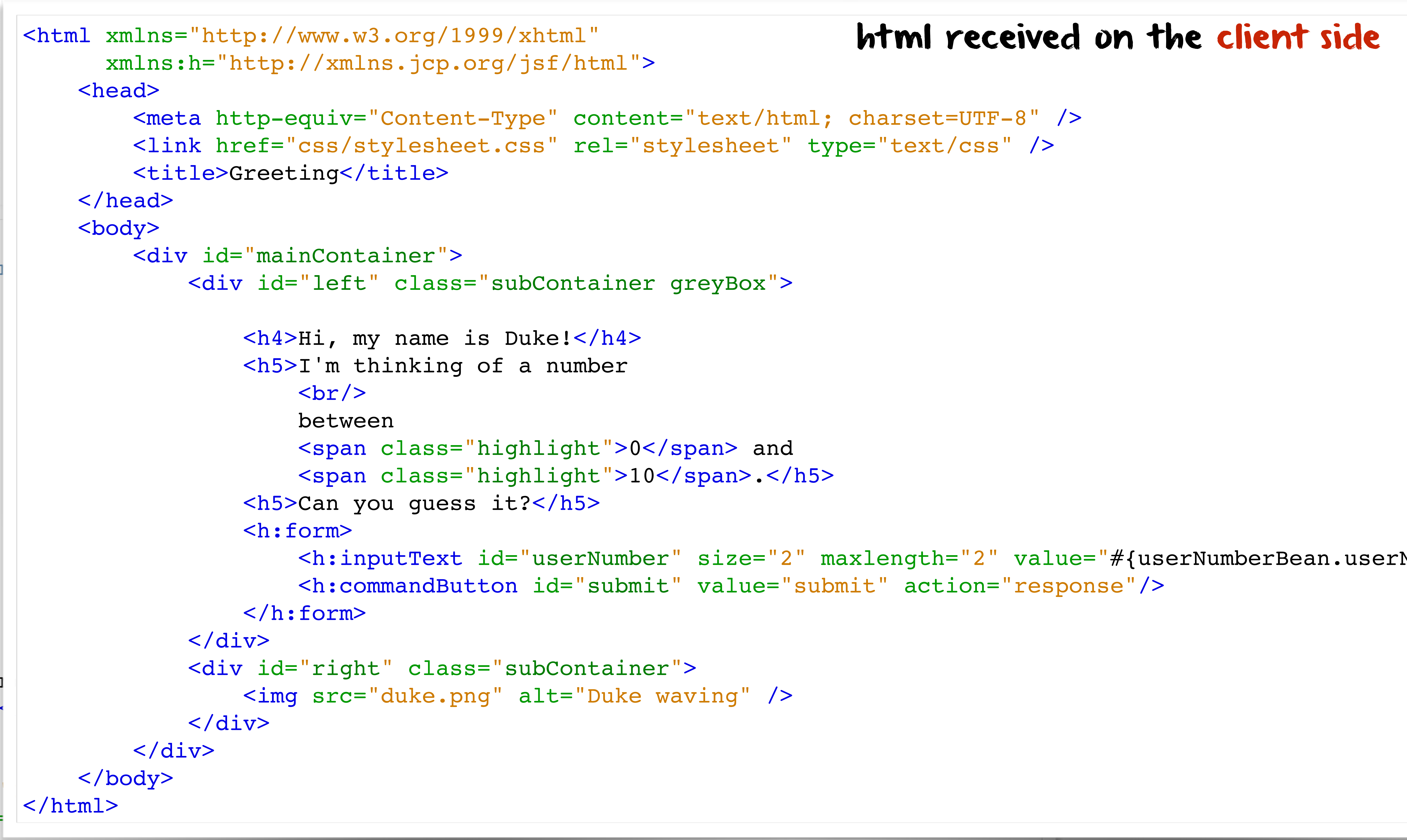
I'm thinking of a number between 0 and 10.

Can you guess it?



```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitio
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:ui="http://xmlns.jcp.org/jsf/facelets"
      xmlns:h="http://xmlns.jcp.org/jsf/html">

    <body>
        <ui:composition template="./template.xhtml">
            <ui:define name="title">
                Greeting
            </ui:define>
            <ui:define name="box">
                <h4>Hi, my name is Duke!</h4>
                <h5>I'm thinking of a number
                    <br/>
                    between
                    <span class="highlight">0</span> a
                    <span class="highlight">10</span>.
                <h5>Can you guess it?</h5>
                <h:form>
                    <h:inputText id="userNumber" size=
                    <h:commandButton id="submit" value=
                </h:form>
            </ui:define>
        </ui:composition>
    </body>
</html>
```

xhtml interpreted on the server side

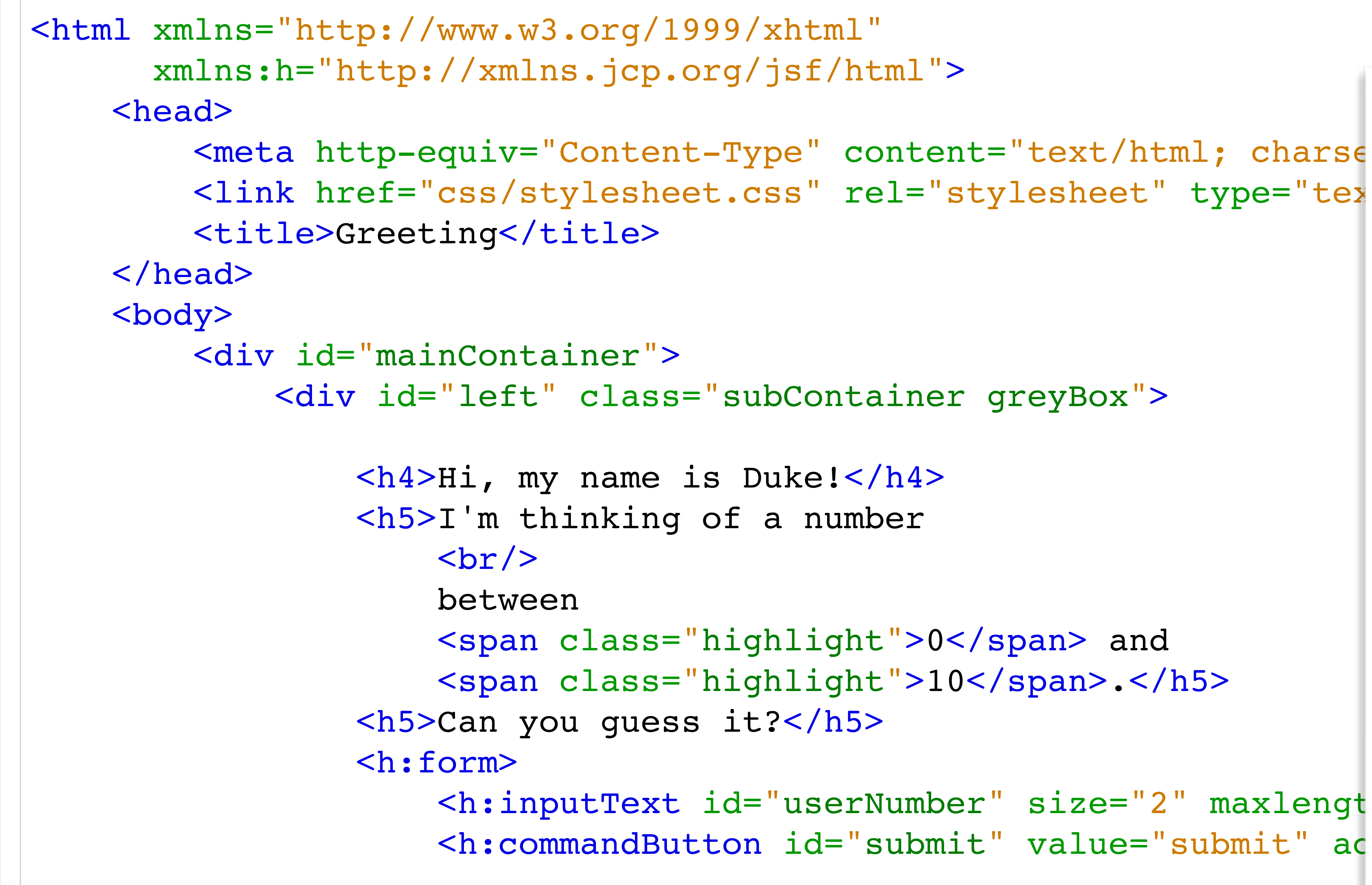


html received on the client side

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html">
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
        <link href="css/stylesheet.css" rel="stylesheet" type="text/css" />
        <title>Greeting</title>
    </head>
    <body>
        <div id="mainContainer">
            <div id="left" class="subContainer greyBox">

                <h4>Hi, my name is Duke!</h4>
                <h5>I'm thinking of a number
                    <br/>
                    between
                    <span class="highlight">0</span> and
                    <span class="highlight">10</span>.</h5>
                <h5>Can you guess it?</h5>
                <h:form>
                    <h:inputText id="userNumber" size="2" maxlength="2" value="#{userNumberBean.userN
                    <h:commandButton id="submit" value="submit" action="response"/>
                </h:form>
            </div>
            <div id="right" class="subContainer">
                <img src="duke.png" alt="Duke waving" />
            </div>
        </div>
    </body>
</html>
```

model-view-controller

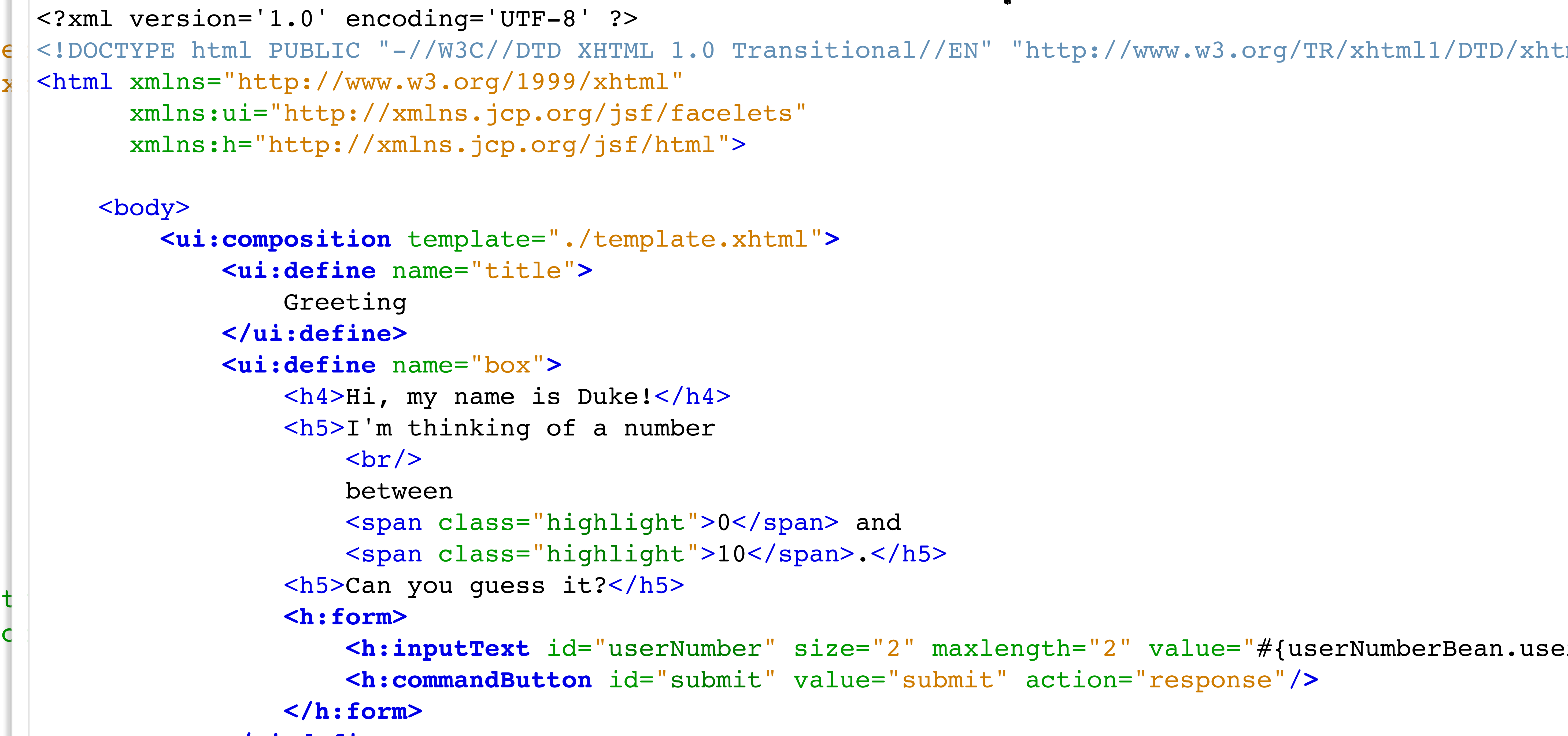
output
input
wallace-palace.local
Hi, my name is Duke!
I'm thinking of a number between 0 and 10.
Can you guess it?
0
submit
view
response
request
web server
web container
greeting.xhtml
controller
remote call
business logic
ejb
model
transactions
database



```
<html xmlns="http://www.w3.org/1999/xhtml"
   xmlns:h="http://xmlns.jcp.org/jsf/html">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charse
    <link href="css/stylesheet.css" rel="stylesheet" type="tex
    <title>Greeting</title>
  </head>
  <body>
    <div id="mainContainer">
      <div id="left" class="subContainer greyBox">

        <h4>Hi, my name is Duke!</h4>
        <h5>I'm thinking of a number
          <br/>
          between
          <span class="highlight">0</span> and
          <span class="highlight">10</span>.</h5>
        <h5>Can you guess it?</h5>
        <h:form>
          <h:inputText id="userNumber" size="2" maxlengt
          <h:commandButton id="submit" value="submit" ac
```

html received on the client side

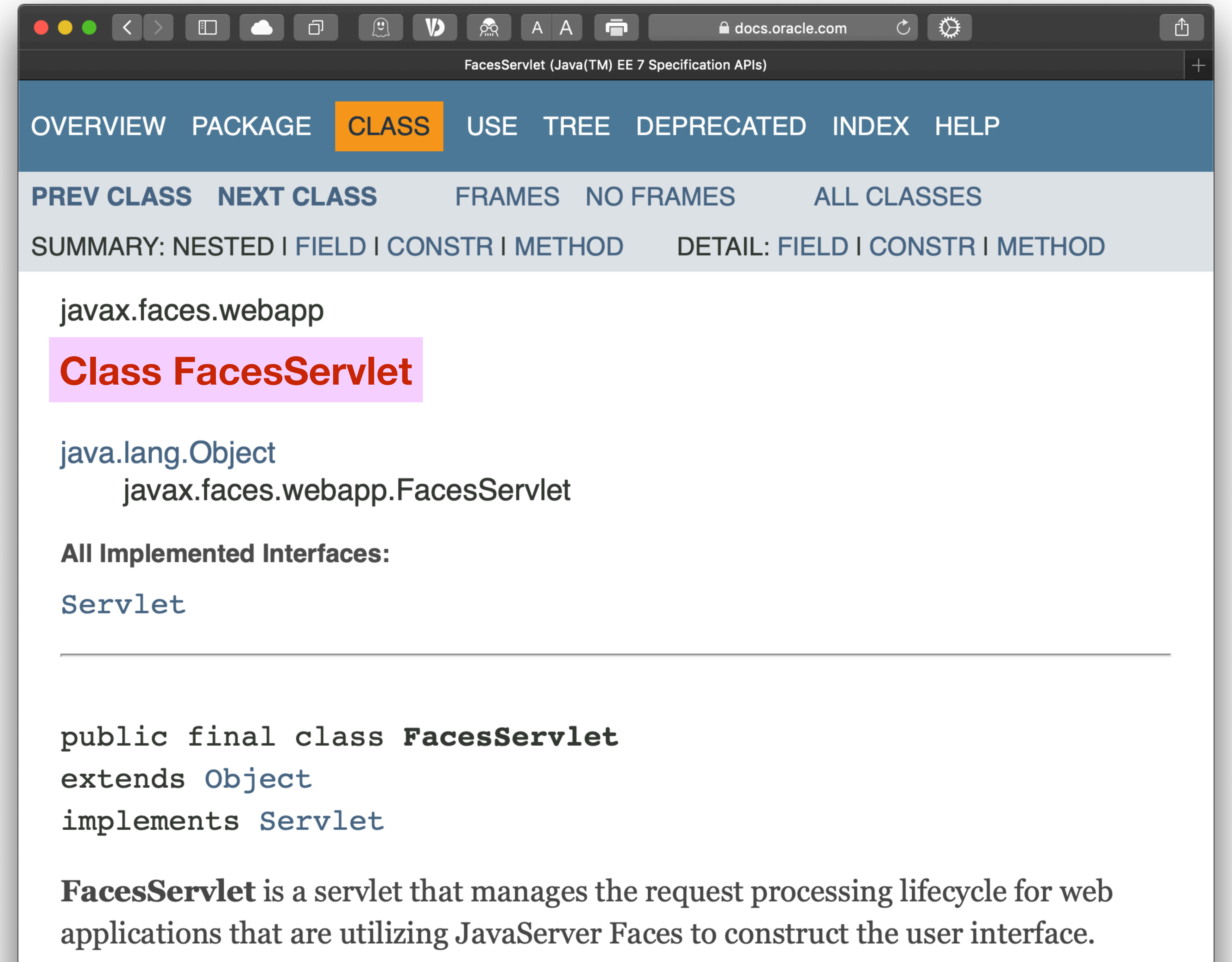
xhtml interpreted on the server side



```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xht
<html xmlns="http://www.w3.org/1999/xhtml"
   xmlns:ui="http://xmlns.jcp.org/jsf/facelets"
   xmlns:h="http://xmlns.jcp.org/jsf/html">

  <body>
    <ui:composition template="./template.xhtml">
      <ui:define name="title">
        Greeting
      </ui:define>
      <ui:define name="box">
        <h4>Hi, my name is Duke!</h4>
        <h5>I'm thinking of a number
          <br/>
          between
          <span class="highlight">0</span> and
          <span class="highlight">10</span>.</h5>
        <h5>Can you guess it?</h5>
        <h:form>
          <h:inputText id="userNumber" size="2" maxlength="2" value="#{userNumberBean.use
          <h:commandButton id="submit" value="submit" action="response"/>
        </h:form>
```

model-view-controller

output
input
wallace-palace.local
Hi, my name is Duke!
I'm thinking of a number between 0 and 10.
Can you guess it?
0
submit
view
response
request
web server
web container
web component
controller
remote call
business logic
ejb
model
transactions
database



FacesServlet (Java(TM) EE 7 Specification APIs)

OVERVIEW PACKAGE CLASS USE TREE DEPRECATED INDEX HELP

PREV CLASS NEXT CLASS FRAMES NO FRAMES ALL CLASSES

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

javax.faces.webapp

Class FacesServlet

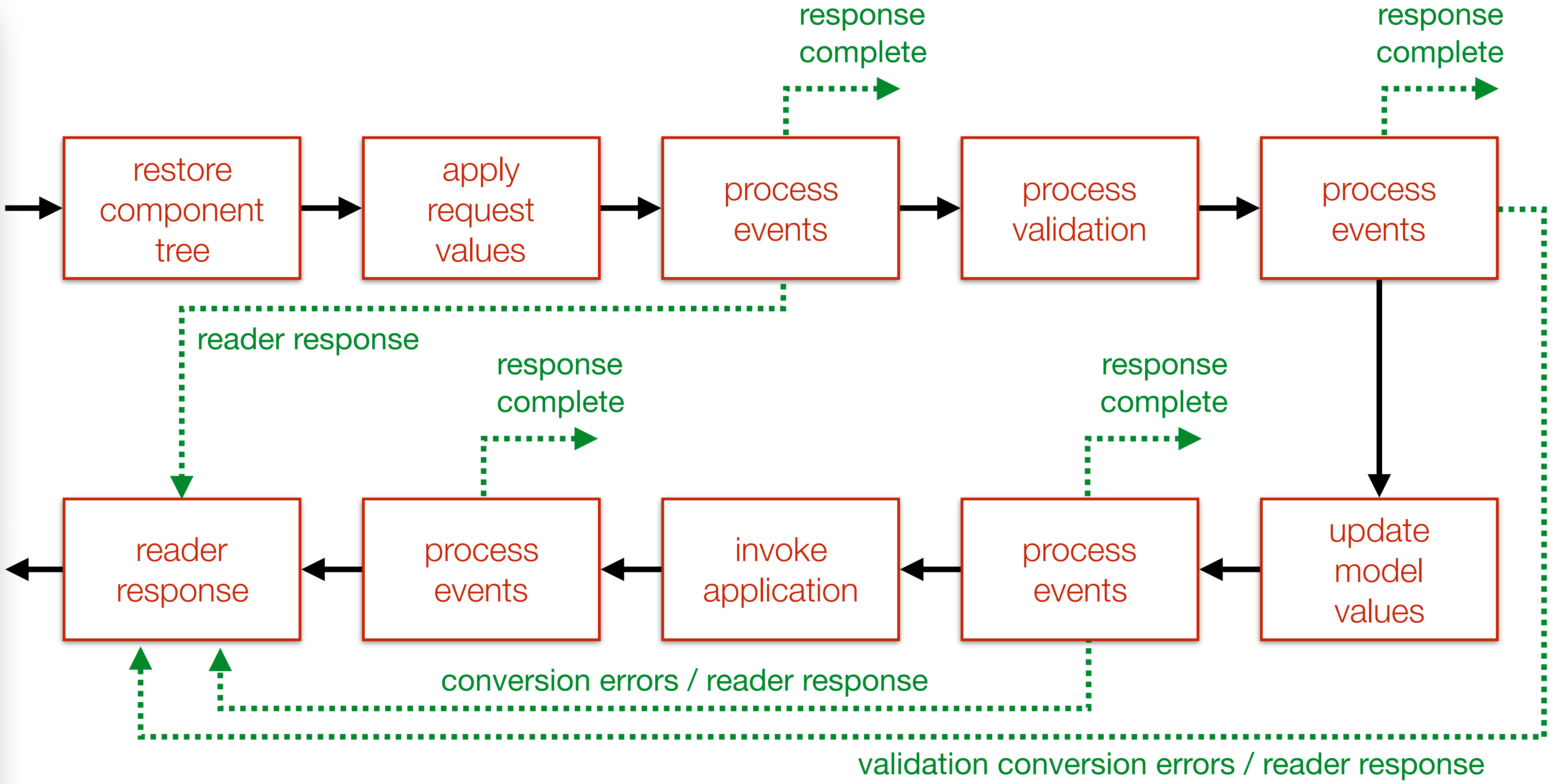
java.lang.Object
    javax.faces.webapp.FacesServlet

All Implemented Interfaces:

`Servlet`

```
public final class FacesServlet
extends Object
implements Servlet
```

FacesServlet is a servlet that manages the request processing lifecycle for web applications that are utilizing JavaServer Faces to construct the user interface.



model-view-controller

output

input

wallace-palace.local

Hi, my name is Duke!

I'm thinking of a number between 0 and 10.

Can you guess it?

0 submit

view

response

request

web server

web container

UserNumberBean

controller

remote call

business logic

ejb

model

transactions

database

```
import javax.inject.Named;
import javax.enterprise.context.SessionScoped;
import javax.faces.context.FacesContext;
import javax.servlet.http.HttpSession;
import java.io.Serializable;
import java.util.Random;

@Named
@SessionScoped
public class UserNumberBean implements Serializable {
   private Integer randomInt = 0;
   private Integer userNumber = 0;
   ⋮
   public Integer getUserNumber() {
        return userNumber;
   }
   public void setUserNumber(Integer userNumber) {
        this.userNumber = userNumber;
   }
   ⋮
}
```

a cdi* java bean

xhtml interpreted on the server side

```
⋮
  <h:form>
    <h:inputText id="userNumber" size="2" maxlength="2" value="#{userNumberBean.userNumber}"/>
    <h:commandButton id="submit" value="submit" action="response"/>
  </h:form>
⋮
```

the FacesServlet (controller) updates the UserNumberBean cdi bean (model)

*context and dependency injection

model-view-controller

output
input
wallace-palace.local
view
response
request
web server
web container
Banking
controller
remote call
business logic
BankBean
model
transactions
database

Hi, my name is Duke!

I'm thinking of a number between 0 and 10.

Can you guess it?

0 submit

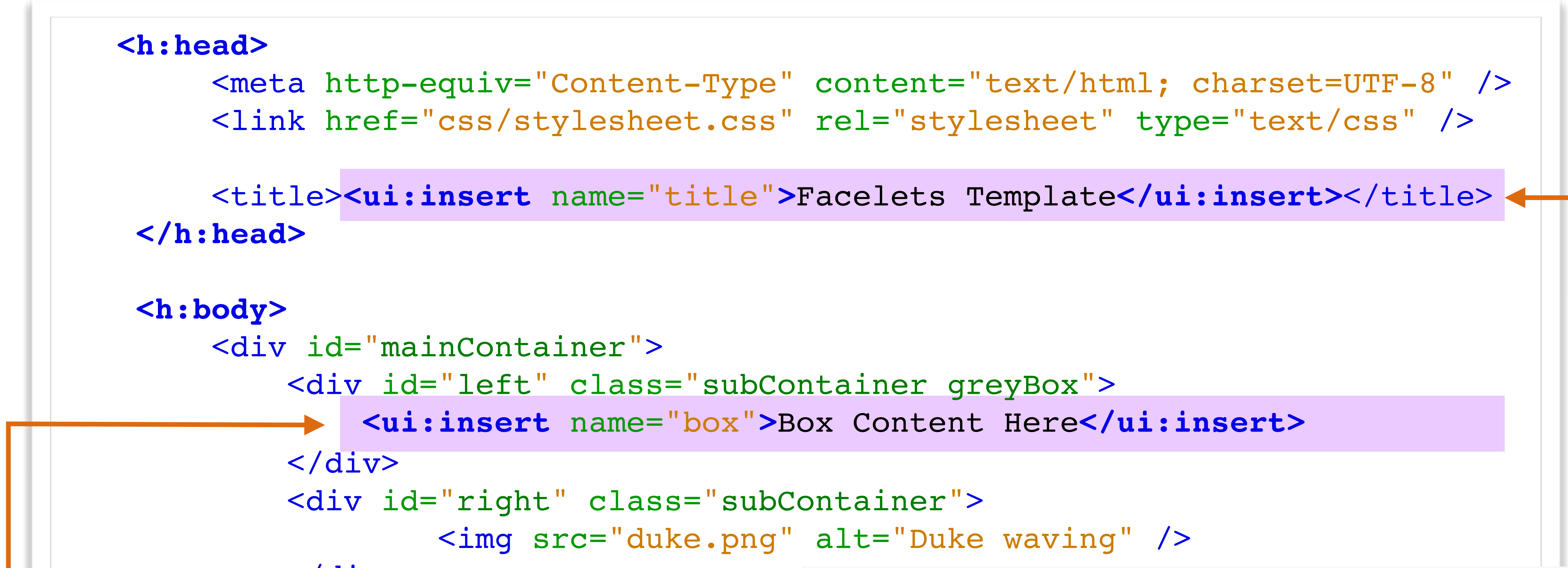
```java
import ch.unil.doplab.bank.Account;
import ch.unil.doplab.bank.ejb.BankBeanLocal;
import javax.inject.Named;
import javax.enterprise.context.SessionScoped;
import java.io.Serializable;
import javax.ejb.EJB;

@Named(value = "banking")
@SessionScoped
public class Banking implements Serializable {
    @EJB
    BankBeanLocal theBank;

   private Account account = null;
   public String getAccountID() {
        if (account == null) {
            account = theBank.openAccount(lastName, firstName);
            return account.getId().toString();
        }
        return account.getId().toString();
    }
    ⋮
}
```

a cdi java bean

```java
import javax.ejb.SessionContext;
import javax.ejb.Stateless;
import javax.ejb.TransactionManagement;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;

@Stateless
@TransactionManagement(javax.ejb.TransactionManagementType.CONTAINER)
public class BankBean implements BankBeanLocal {
    @Resource
    SessionContext context;

    @PersistenceContext
    private EntityManager manager;

    @Override
    public Account openAccount(String lastName, String firstName) {
       Account account = new Account(lastName, firstName);
       manager.persist(account);
       return account;
    }
    ⋮
}
```

an enterprise java bean

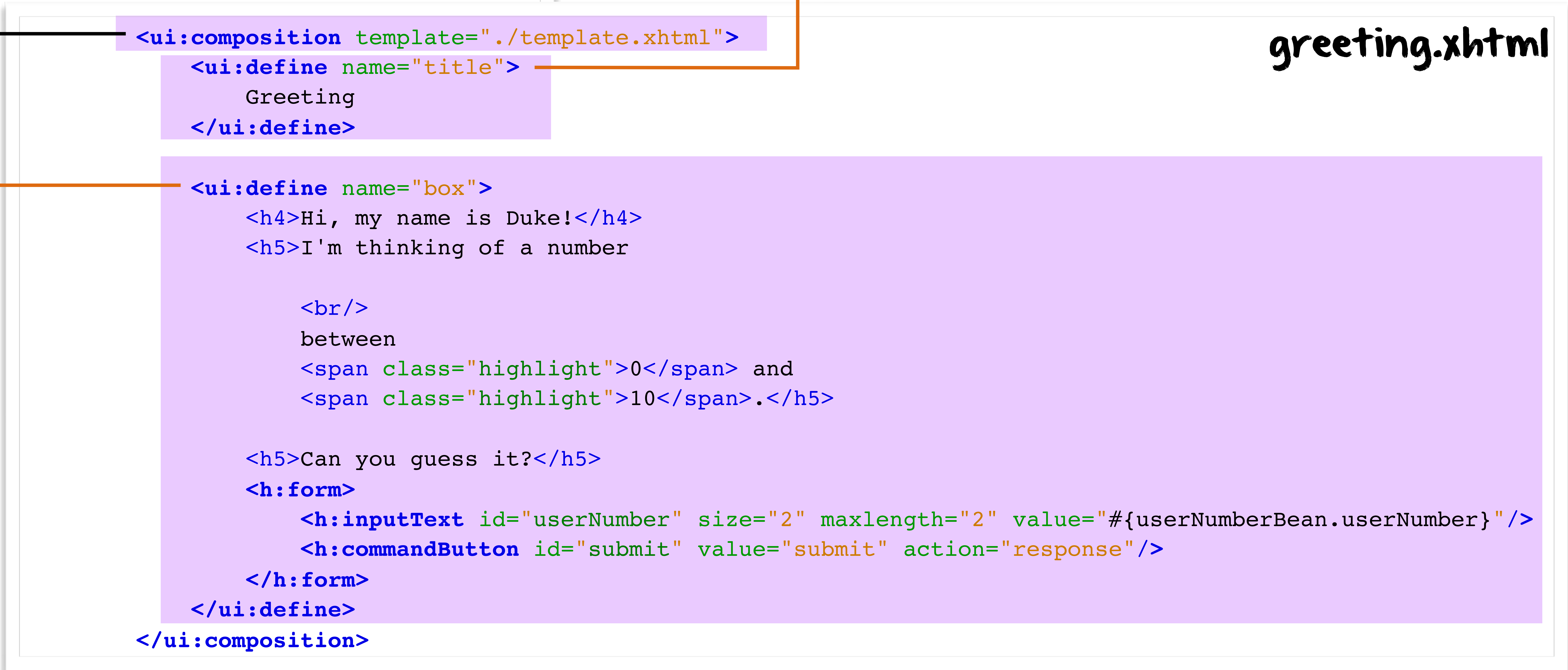
facelet templating



```
  <h:head>
       <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
       <link href="css/stylesheet.css" rel="stylesheet" type="text/css" />

       <title><ui:insert name="title">Facelets Template</ui:insert></title>
   </h:head>

   <h:body>
       <div id="mainContainer">
           <div id="left" class="subContainer greyBox">
             <ui:insert name="box">Box Content Here</ui:insert>
           </div>
           <div id="right" class="subContainer">
                   <img src="duke.png" alt="Duke waving" />
           </div>
       </div>
   </h:body>
```

template.xhtml

facelet templating **factors out** recurring web **page design**



greeting.xhtml

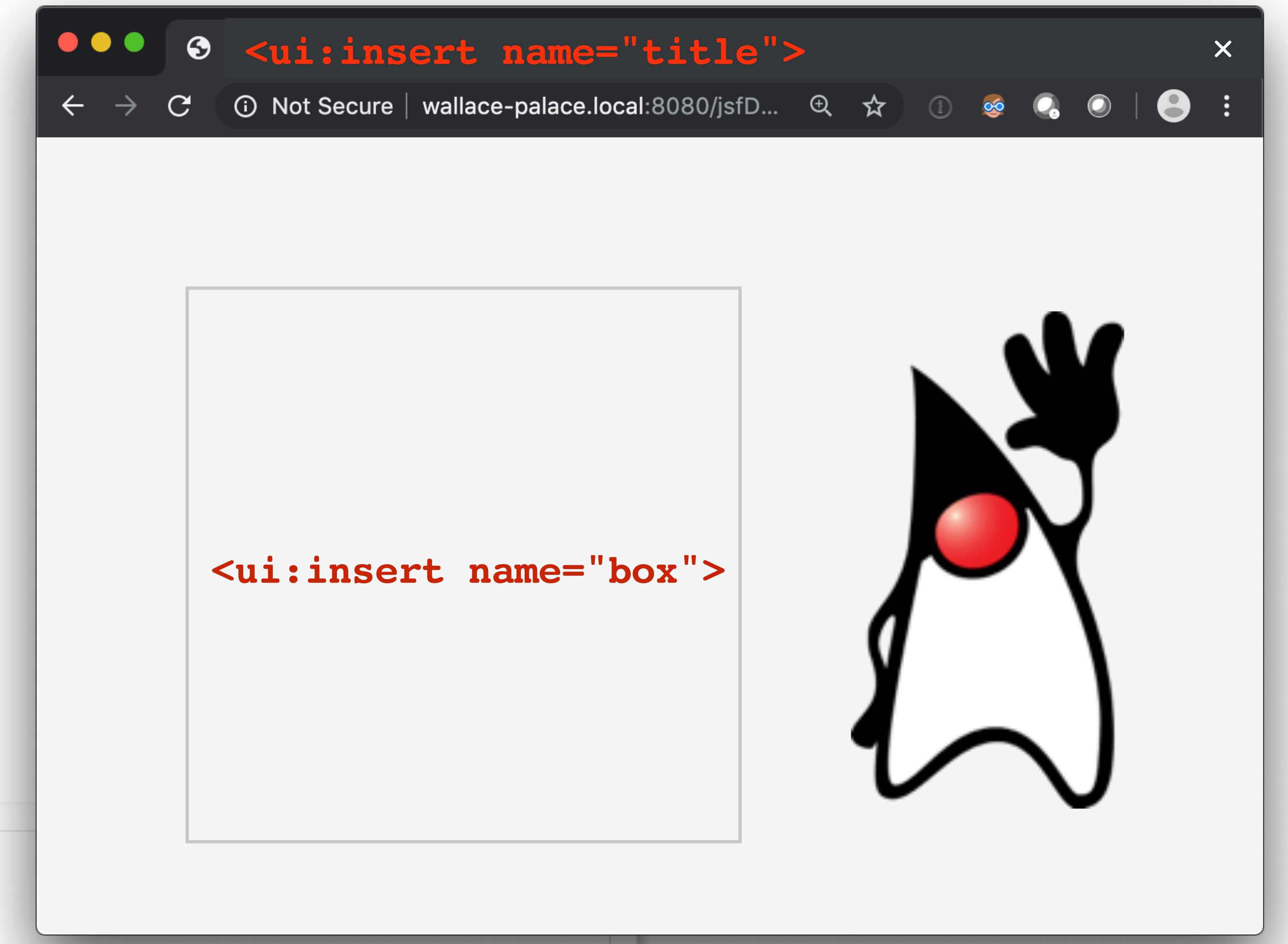
```
<ui:composition template="./template.xhtml">
    <ui:define name="title">
        Greeting
    </ui:define>

    <ui:define name="box">
        <h4>Hi, my name is Duke!</h4>
        <h5>I'm thinking of a number

            <br/>
            between
            <span class="highlight">0</span> and
            <span class="highlight">10</span>.</h5>

        <h5>Can you guess it?</h5>
        <h:form>
            <h:inputText id="userNumber" size="2" maxlength="2" value="#{userNumberBean.userNumber}"/>
            <h:commandButton id="submit" value="submit" action="response"/>
        </h:form>
    </ui:define>
</ui:composition>
```

reuse of web **graphical elements**

facelet templating

template.xhtml

```
<h:head>
     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
     <link href="css/stylesheet.css" rel="stylesheet" type="text/css" />

     <title><ui:insert name="title">Facelets Template</ui:insert></title>
 </h:head>

 <h:body>
     <div id="mainContainer">
         <div id="left" class="subContainer greyBox">
             <ui:insert name="box">Box Content Here</ui:insert>
         </div>
         <div id="right" class="subContainer">
                 <img src="duke.png" alt="Duke waving" />
         </div>
     </div>
 </h:body>
```

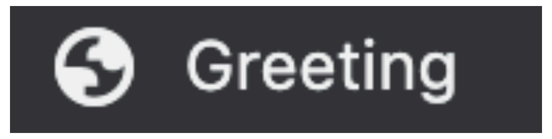


<ui:insert name="title">

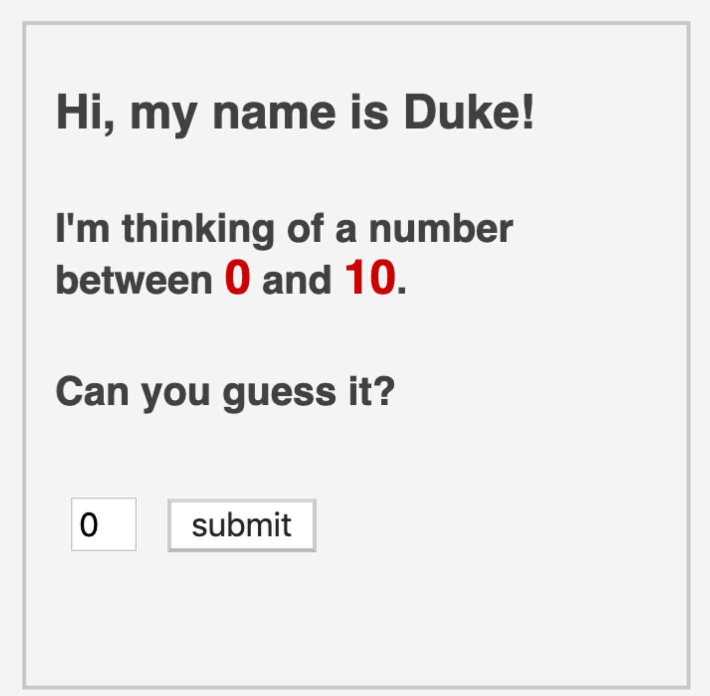
Not Secure | wallace-palace.local:8080/jsfD...

<ui:insert name="box">

```
<ui:composition template="./template.xhtml">
    <ui:define name="title">
        Greeting
    </ui:define>

    <ui:define name="box">
        <h4>Hi, my name is Duke!</h4>
        <h5>I'm thinking of a number

            <br/>
            between
            <span class="highlight">0</span> and
            <span class="highlight">10</span>.</h5

        <h5>Can you guess it?</h5>
        <h:form>
            <h:inputText id="userNumber" size="2" maxlength="2" value="#{userNumberBean.userNumber}"/>
            <h:commandButton id="submit" value="submit" action="response"/>
        </h:form>
    </ui:define>
</ui:composition>
```



Greeting

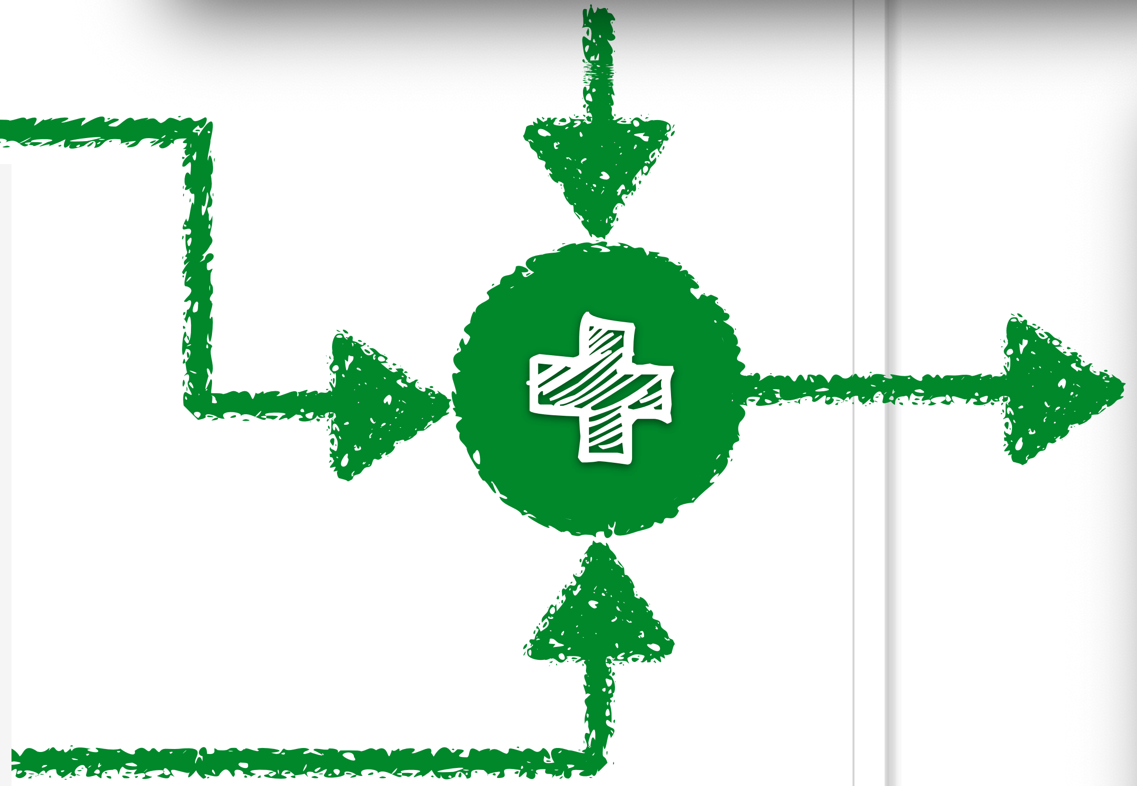


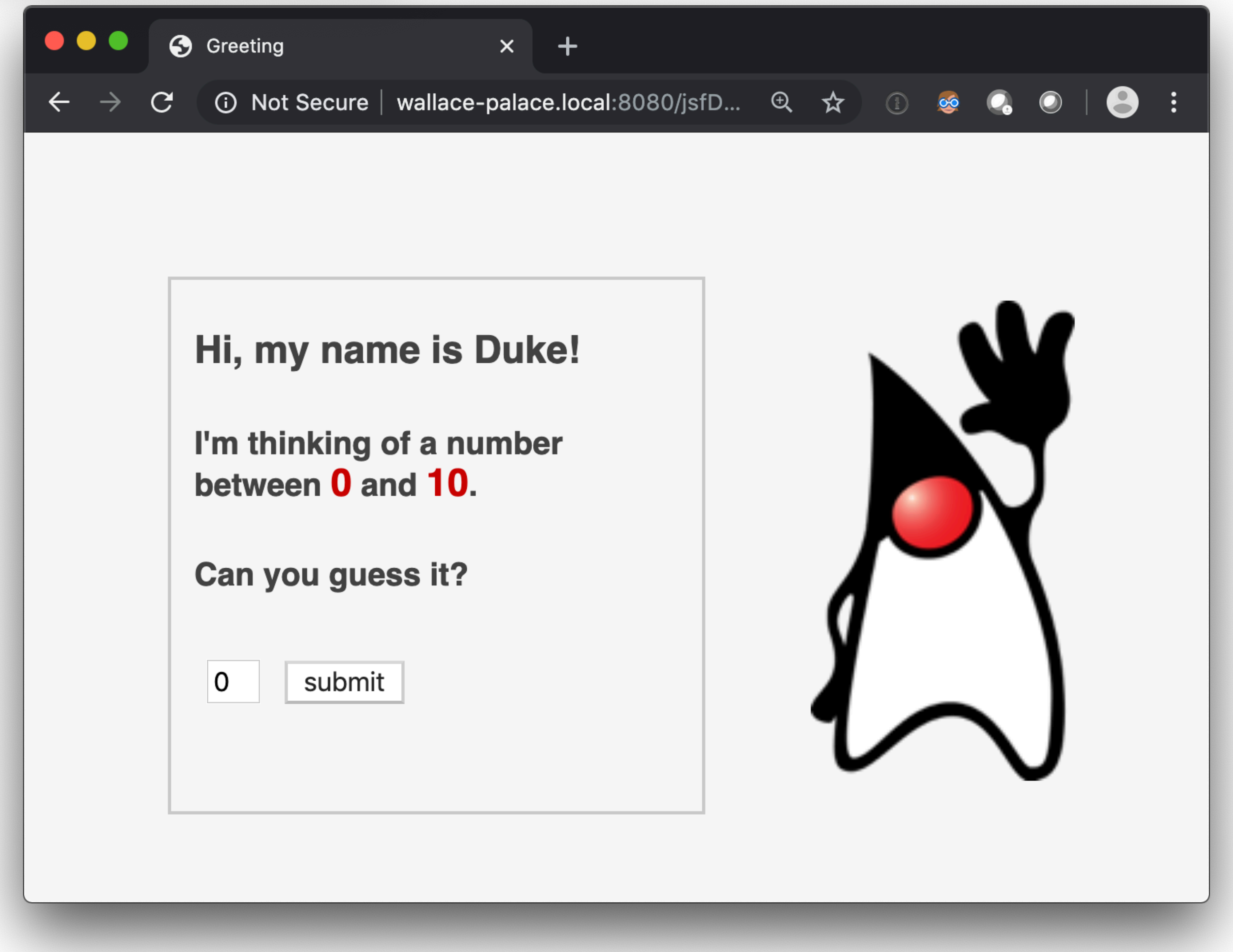
Hi, my name is Duke!

I'm thinking of a number between 0 and 10.

Can you guess it?

0 submit





Greeting

Not Secure | wallace-palace.local:8080/jsfD...

Hi, my name is Duke!

I'm thinking of a number between 0 and 10.

Can you guess it?

0 submit

greeting.xhtml