Apache Wicket

Java Web Application Framework

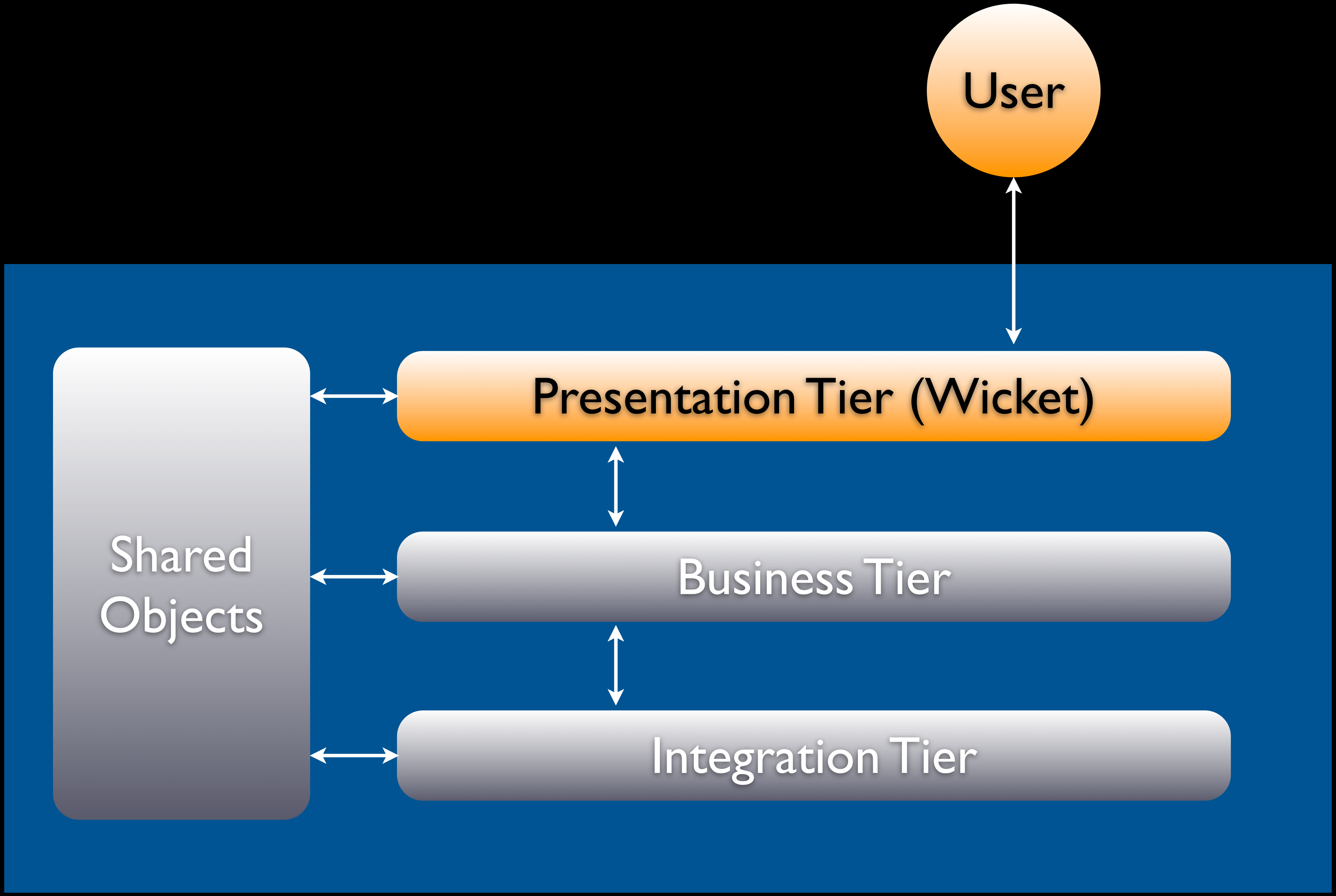
St. Louis - Java User's Group

Luther Baker • September 2009

What is Wicket?

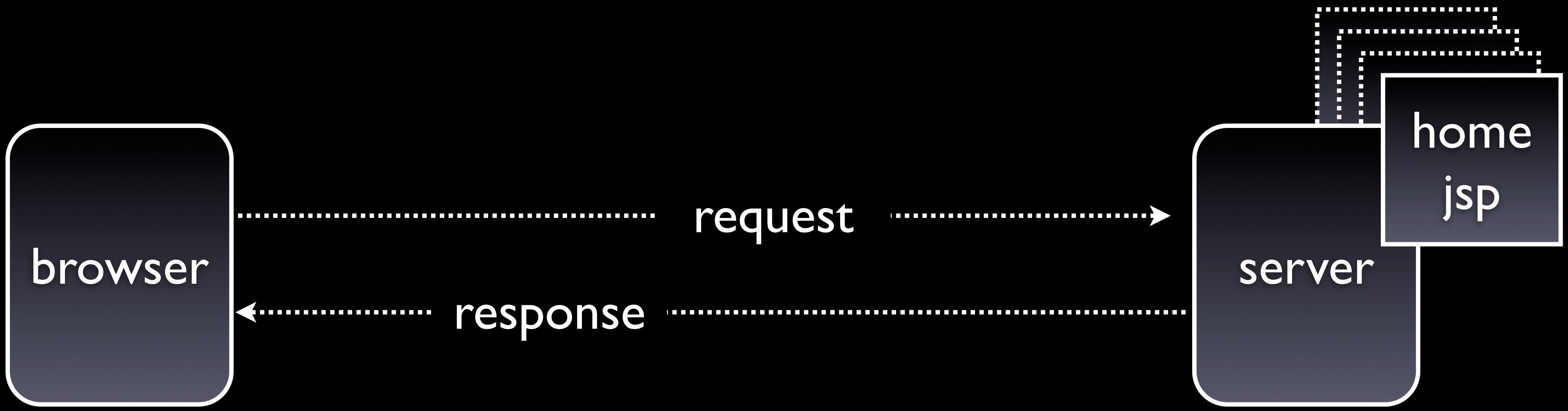
- Web Application Framework
- Component-based Framework
- Wicket 1.4 is Java 1.5+ compliant

Where does Wicket fit?



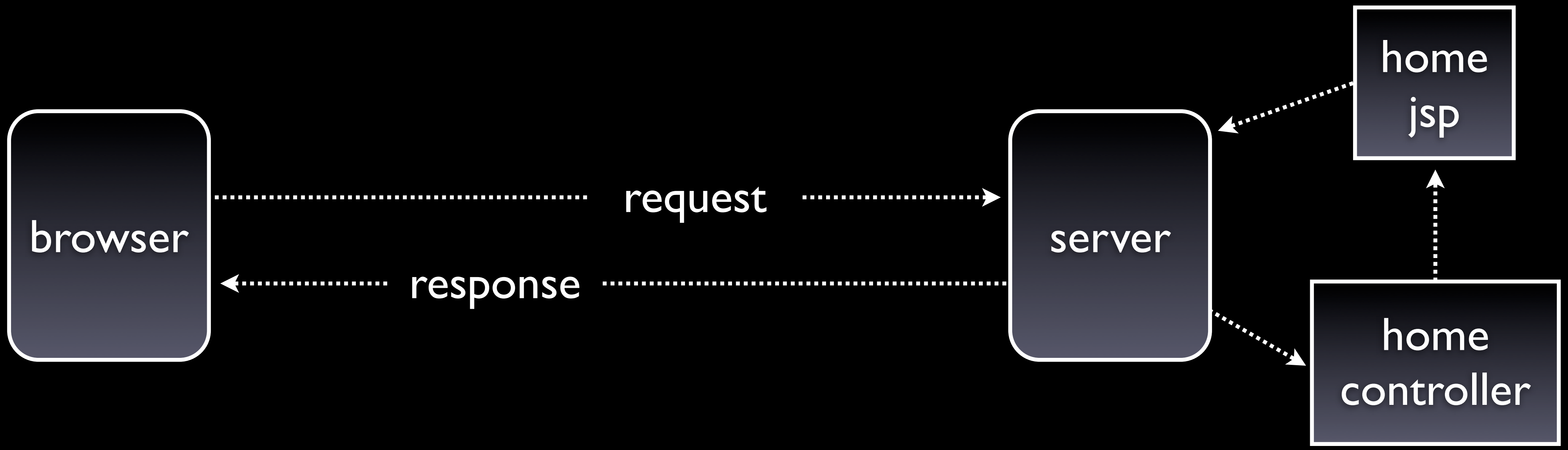
Request / Response

JSP Request Response



Model 2 (MVC)

Struts, Spring MVC, Stripes



Advantages of R/R

- Rendering views is generally quite fast
- Leverage existing tag libraries
- Recruiting developers may be easier
- Modern implementations have good support for DI and IoC frameworks

Disadvantages of R/R

- Controller implementations must be considerate of multiple concurrent users and threads
- Controllers generally work literally in terms of requests and responses
- Controllers must often explicitly manage state
- Not necessarily Object Oriented
- The programming model is skewed

The Impedance Mismatch

The Programming Model

- Programming in Java - do we regularly focus on how the JVM manages object instances and variables?
- Generally, website development must happen with the HTTP protocol in mind, manually managing state within and across requests forcing front end handlers to be protocol specific.

What if ... ?

- What if we considered a page a Page?
- What if we considered a button, a Button?
- And on a button click, handled an onClick event?
- What if a web development resembled Swing or event-driven development?
- What kind of framework could do this?

Enter ... Wicket

- Component-based framework
- Instead of creating a controller, servlet or action class, create a Page
- Place Components on the page and define how each component reacts to user input
- Build the page in Java to manipulate the HTML file, not the other way around

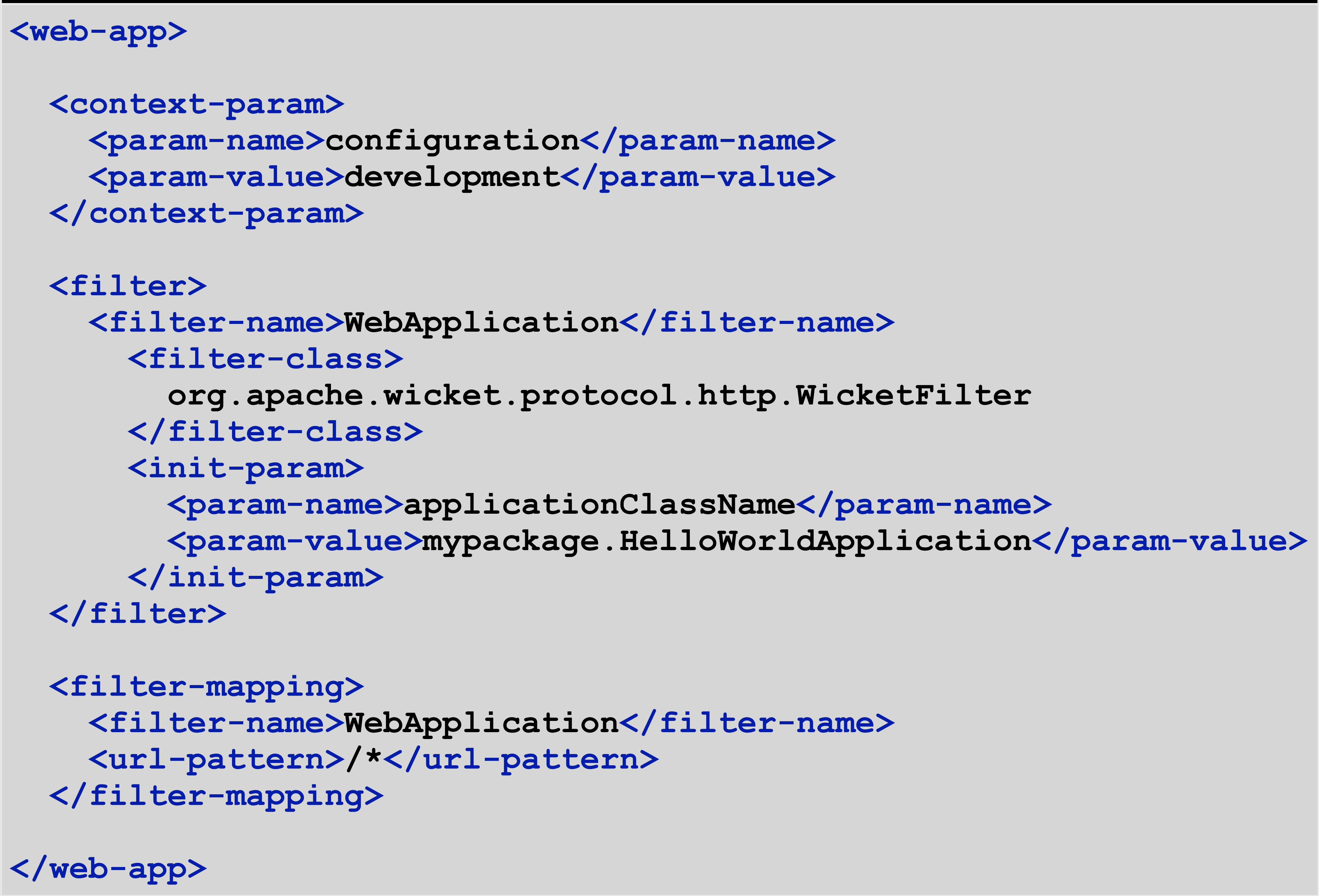
The Component Model

The Underlying Abstraction

- Graphic devices and their representations are created in Java along with an HTML counterpart that reflects the Java hierarchy of objects.
- DOM style parent/child approach
- Event driven programming model
- How can such an abstraction could be created on top of HTTP?

Application Config

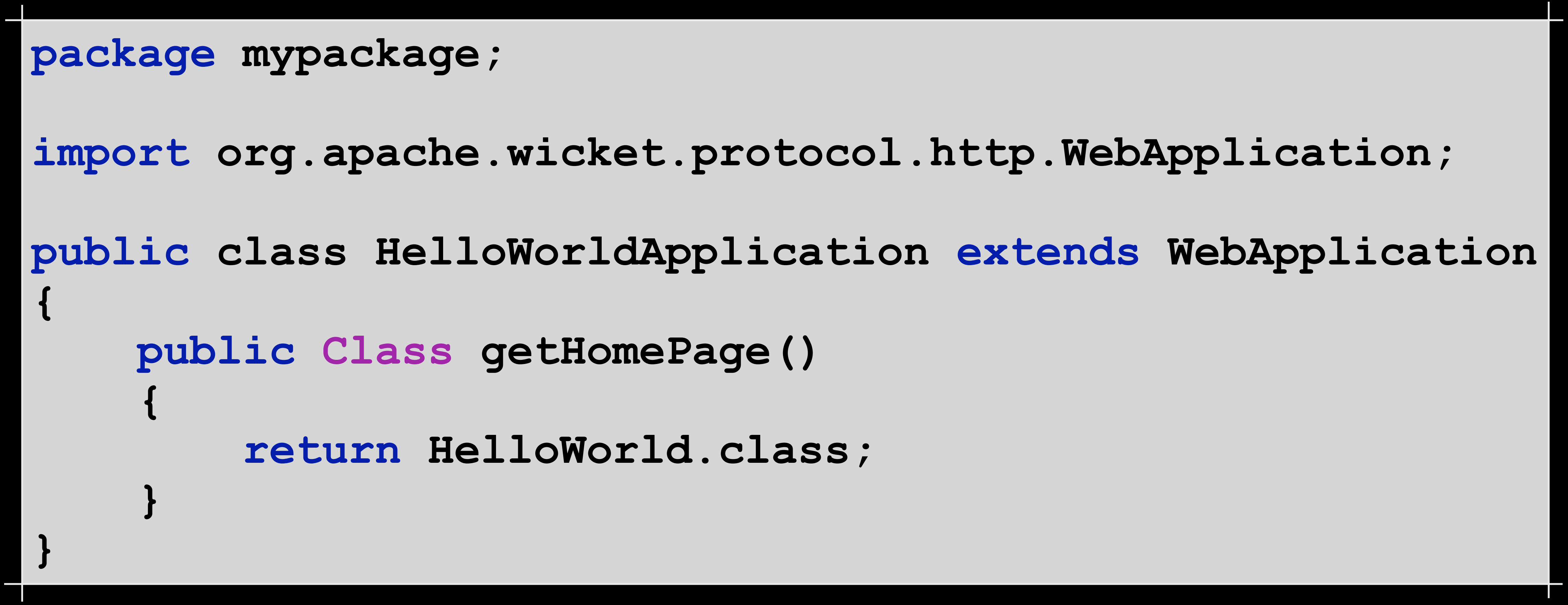
web.xml



```xml
<web-app>

  <context-param>
    <param-name>configuration</param-name>
    <param-value>development</param-value>
  </context-param>

  <filter>
    <filter-name>WebApplication</filter-name>
      <filter-class>
        org.apache.wicket.protocol.http.WicketFilter
      </filter-class>
      <init-param>
        <param-name>applicationClassName</param-name>
        <param-value>mypackage.HelloWorldApplication</param-value>
      </init-param>
  </filter>

  <filter-mapping>
    <filter-name>WebApplication</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>

</web-app>
```

Wicket Config

WicketApplication.java



```
package mypackage;

import org.apache.wicket.protocol.http.WebApplication;

public class HelloWorldApplication extends WebApplication
{
    public Class getHomePage()
    {
        return HelloWorld.class;
    }
}
```

General Structure

Page Responsiblities

- Layout the Element Hierarchy
- Styles elements

Java Role

- Matching Hierarchy
- Event Handling

Properties

- Strings and i18n

Hello World

Markup

```
<html>
  <body>
    <span wicket:id="message">Message goes here!</span>
  </body>
</html>
```

Java

```
import org.apache.wicket.markup.html.WebPage;
import org.apache.wicket.markup.html.basic.Label;

public class HelloWorld extends WebPage
{
  public HelloWorld()
  {
    add(new Label("message", "Hello World!"));
  }
}
```

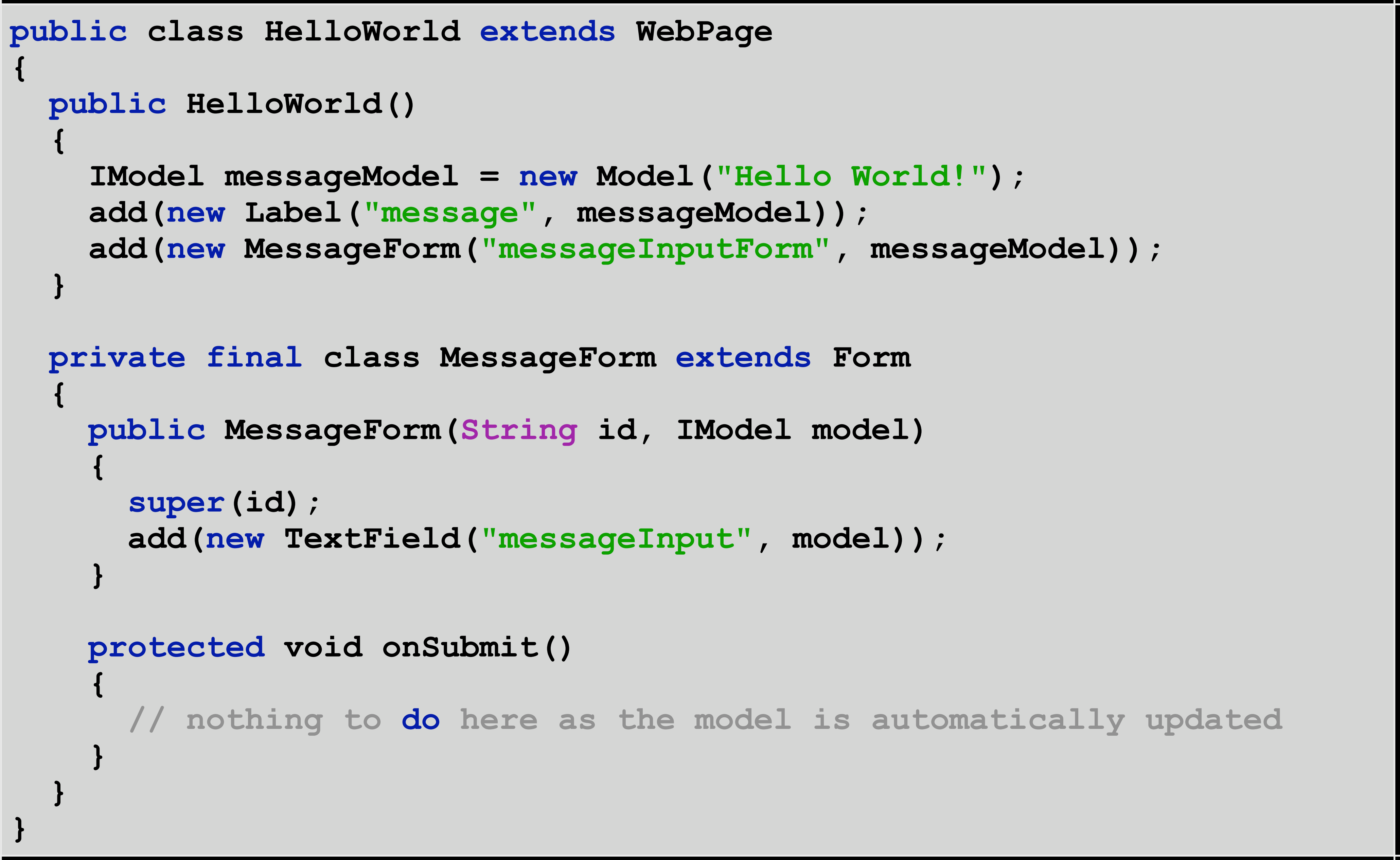
Forms (HTML)

Markup

```
<html>
  <body>
    <span wicket:id="message">Message goes here</span>
    <form wicket:id="messageInputForm">
      <input type="text" wicket:id="messageInput"/>
      <input type="submit" value="update"/>
    </form>
  </body>
</html>
```

Forms (Java)

Java



```java
public class HelloWorld extends WebPage
{
  public HelloWorld()
  {
    IModel messageModel = new Model("Hello World!");
    add(new Label("message", messageModel));
    add(new MessageForm("messageInputForm", messageModel));
  }

  private final class MessageForm extends Form
  {
    public MessageForm(String id, IModel model)
    {
      super(id);
      add(new TextField("messageInput", model));
    }

    protected void onSubmit()
    {
      // nothing to do here as the model is automatically updated
    }
  }
}
```

Component Family

Component

Page

WebComponent

Link

Panel

WebPage

AjaxLink

Button

Checkbox

TextArea

Form

TextField

DropDown

Label

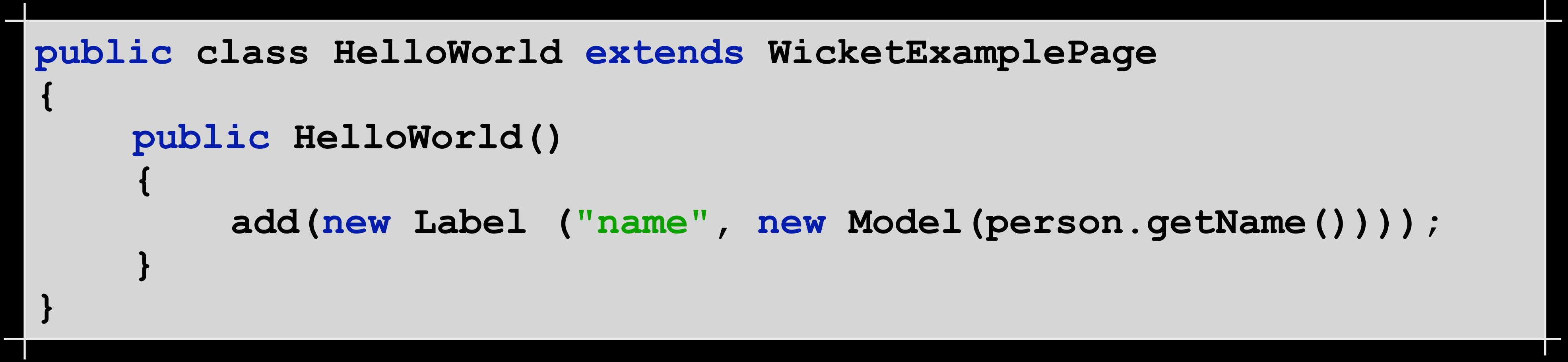
...many more

Super Models

- Static
- Dynamic
- Property
- Compound Property
- Loadable Detached

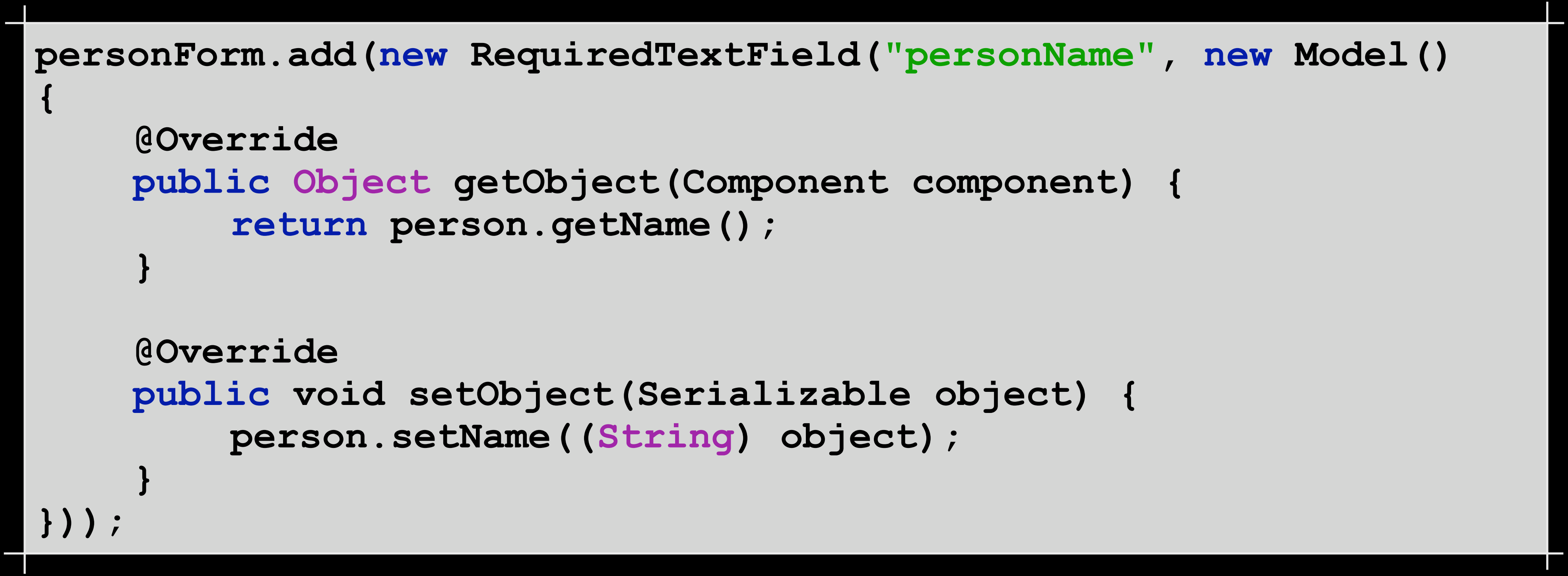
Basic Models

Static Model



```
public class HelloWorld extends WicketExamplePage
{
    public HelloWorld()
    {
        add(new Label ("name", new Model(person.getName())));
    }
}
```

Dynamic Model



```
personForm.add(new RequiredTextField("personName", new Model()
{
    @Override
    public Object getObject(Component component) {
        return person.getName();
    }

    @Override
    public void setObject(Serializable object) {
        person.setName((String) object);
    }
}));
```

More Models

Property Model

```
public PropertyModel(final Object modelObject, final String expression)
```

```
class Person
{
  private Address address;

  public Address getAddress()
  {
    return name;
  }
  ...
}
```

```
class Address
{
  private String zip;

  public String getZip()
  {
    return zip;
  }
  ...
}
```

```
personForm.add(new RequiredTextField("zip",
               new PropertyModel(person, "address.zip")));
```

Demo

WebPage Component

- Java

  Subclass Master Page

- HTML

  child & parent tags

Custom Component

- Java, HTML
- Panel
- Custom components
- Includes

Events

- HTML
- Javascript
- Ajax
- Submit

Ideally Suited For ...

- Interactive style apps
- Help desk style Ticketing applications
- Online Registration applications
- Apps with lots of Forms and Ajax processing
- Apps simulating thick clients
- Anytime an event programming model better suites the problem domain

Deeper Dive

Maven Quickstart Archetype

- Builds a simple Maven Project

Other Topics

- More on Models
- Security
- Unit Testing
- Custom components
- URL mapping, IoC Integration, Persistence ...

Resources

- http://wicket.apache.org/
- http://wicketstuff.org/
- http://cwiki.apache.org/WICKET/
- Wicket in Action (Manning)
- Pro Wicket (Apress)