



Architecting Applications using Apache Wicket

Andrew Lombardi

Owner, Tech Evangelist

Mystic Coders, LLC

andrew AT mysticcoders DOT com

kinabalu @ irc://irc.freenode.net - ##wicket, ##java

kinabalu @ twitter



To our success!

Recovery steps for MVC UI developers



Recovery steps for MVC UI developers






Admit powerless over clean, easy to read code





Recovery steps for MVC UI developers

-  **Admit powerless over clean, easy to read code**
-  **Came to believe Wicket could restore our sanity**






Recovery steps for MVC UI developers

-  **Admit powerlessness over clean, easy to read code**
-  **Came to believe Wicket could restore our sanity**
-  **Made a decision to refactor our codebase from Struts**







Recovery steps for MVC UI developers

-  **Admit powerless over clean, easy to read code**
-  **Came to believe Wicket could restore our sanity**
-  **Made a decision to refactor our codebase from Struts**
-  **Introspection of self on why we made web development so difficult before**

Recovery steps for MVC UI developers

-  **Admit powerless over clean, easy to read code**
-  **Came to believe Wicket could restore our sanity**
-  **Made a decision to refactor our codebase from Struts**
-  **Introspection of self on why we made web development so difficult before**
-  **Published our code fails in MVC to thedailywtf**

Recovery steps for MVC UI developers

-  **Admit powerless over clean, easy to read code**
-  **Came to believe Wicket could restore our sanity**
-  **Made a decision to refactor our codebase from Struts**
-  **Introspection of self on why we made web development so difficult before**
-  **Published our code fails in MVC to thedailywtf**
-  **Prepared our computer for the shift in framework technology**

Recovery steps for MVC UI developers




Recovery steps for MVC UI developers

 **Read through the Migration to Wicket from Struts
on the wiki**





Recovery steps for MVC UI developers

-  **Read through the Migration to Wicket from Struts on the wiki**
-  **Listed out people / co we told MVC was the light**

Recovery steps for MVC UI developers






-  **Read through the Migration to Wicket from Struts on the wiki**
-  **Listed out people / co we told MVC was the light**
-  **Offered to help refactor their codebase to Wicket**

Recovery steps for MVC UI developers

-  **Read through the Migration to Wicket from Struts on the wiki**
-  **Listed out people / co we told MVC was the light**
-  **Offered to help refactor their codebase to Wicket**
-  **Searched mailing lists / blogs to apologize for our MVC rants**









Recovery steps for MVC UI developers

-  **Read through the Migration to Wicket from Struts on the wiki**
-  **Listed out people / co we told MVC was the light**
-  **Offered to help refactor their codebase to Wicket**
-  **Searched mailing lists / blogs to apologize for our MVC rants**
-  **Read the wiki, the mailing list, and articles so that we may learn more of him, almighty Wicket**



Recovery steps for MVC UI developers

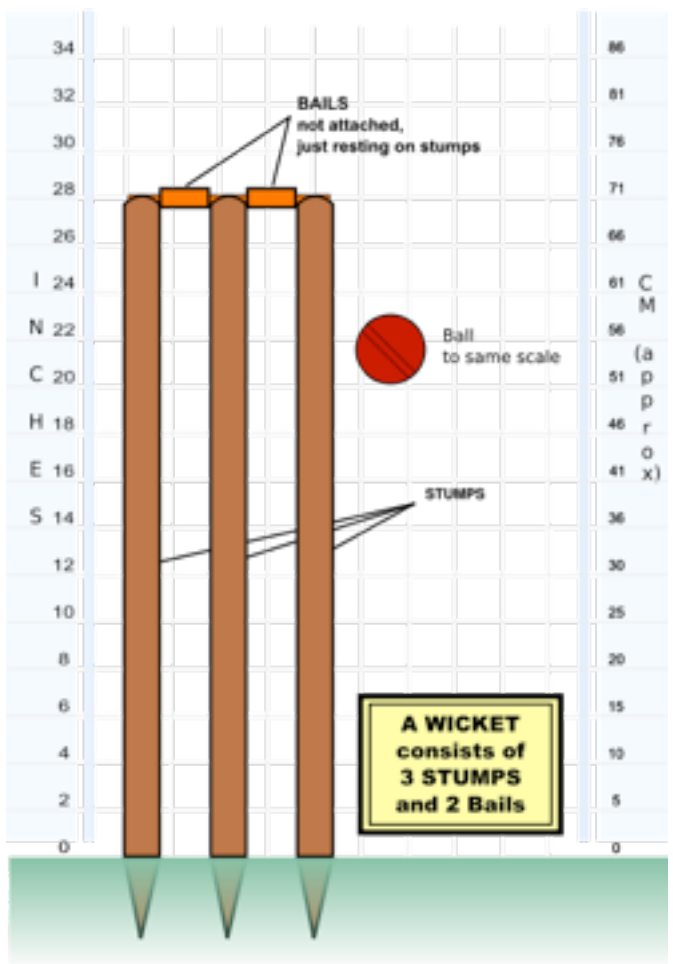
-  **Read through the Migration to Wicket from Struts on the wiki**
-  **Listed out people / co we told MVC was the light**
-  **Offered to help refactor their codebase to Wicket**
-  **Searched mailing lists / blogs to apologize for our MVC rants**
-  **Read the wiki, the mailing list, and articles so that we may learn more of him, almighty Wicket**
-  **Carry out the Wicket message**



We're on the 12th step



What is a Wicket?





+







That's it.



```
public class MyPage
    extends WebPage {
    ...
}
```

```
<html>
<head>
<title>My Page</title>
</head>
<body>
...
</body>
</html>
```



No XML



```
public class MysticPasteApplication extends WebApplication
{

    protected void init() {
        super.init();

        addComponentInstantiationListener(new SpringComponentInjector(this));

        mountBookmarkablePage("/home", HomePage.class);
        mountBookmarkablePage("/new", PasteItemPage.class);
        mountBookmarkablePage("/history", HistoryPage.class);
        mount(new IndexedParamUrlCodingStrategy("/view", ViewPublicPage.class));
        mount(new IndexedParamUrlCodingStrategy("/private", ViewPrivatePage.class));

        ServletContext servletContext = super.getServletContext();
        applicationContext = WebApplicationContextUtils.getWebApplicationContext(servletContext);

        org.apache.wicket.util.lang.Objects.setObjectStreamFactory(
            new IOObjectStreamFactory.DefaultObjectStreamFactory());
    }

    public static MysticPasteApplication getInstance() {
        return ((MysticPasteApplication) WebApplication.get());
    }
}
```





M



M

Model



M

Model

V



M

Model

V

View



M

Model

V

View

C

M

Model

V

View

C

Component

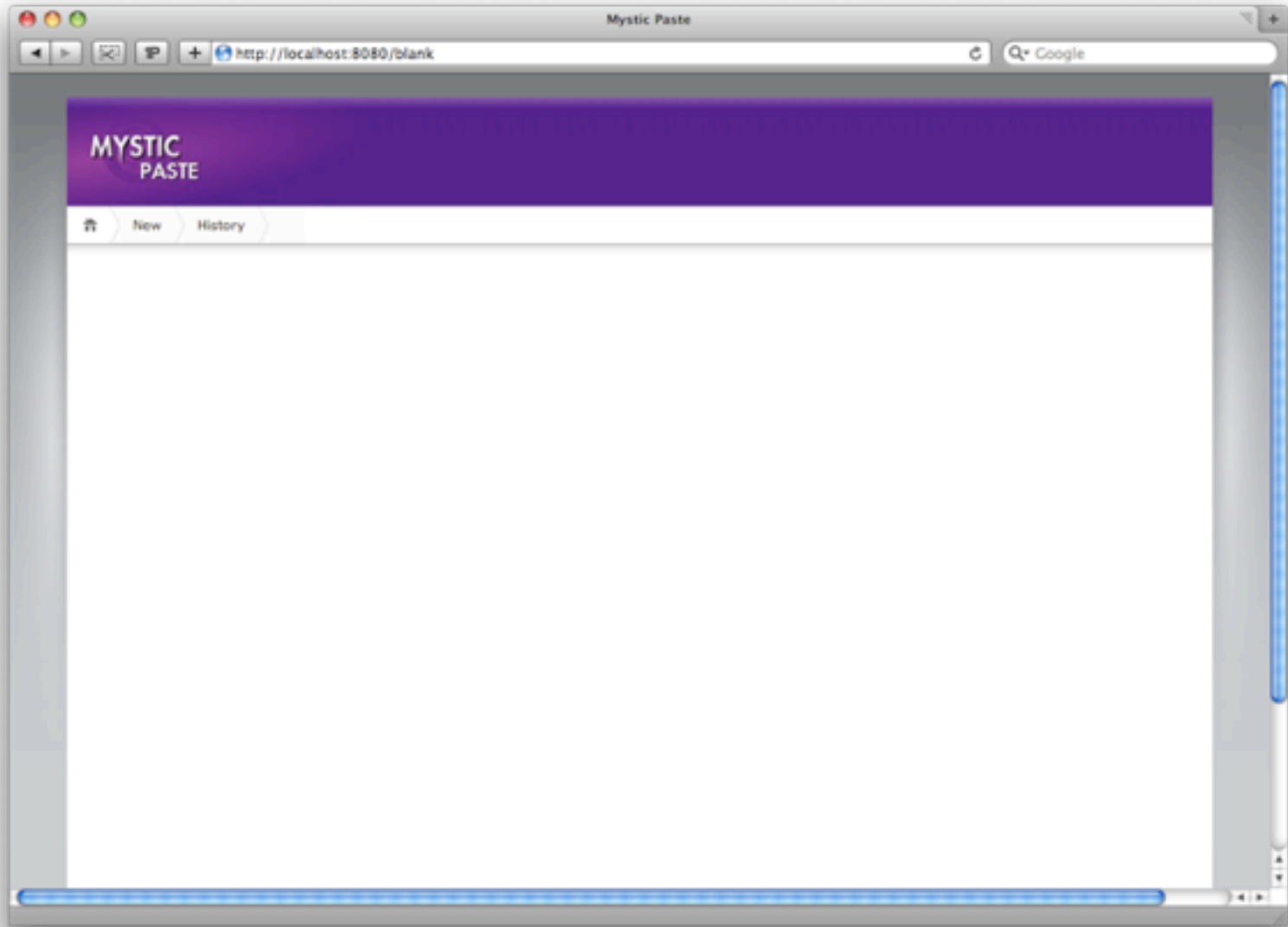


View

Renders the model into a form suitable for interaction.



```
<html>  
<head>  
<title>My Page</title>  
</head>  
<body>  
...  
</body>  
</html>
```





```
public class BasePage extends WebPage {

    Class activePage;

    public BasePage() {
        init();
    }

    public BasePage(Class activePage) {
        this.activePage = activePage;
        init();
    }

    private void init() {
        WebMarkupContainer newLinkContainer = new WebMarkupContainer("newLinkContainer");
        if (activePage != null && activePage.equals(PasteItemPage.class)) {
            newLinkContainer.add(new SimpleAttributeModifier("class", "current_page_item"));
        }
        newLinkContainer.add(new BookmarkablePageLink("newLink", PasteItemPage.class));
        add(newLinkContainer);

        WebMarkupContainer historyLinkContainer = new WebMarkupContainer("historyLinkContainer");
        if (activePage != null && activePage.equals(HistoryPage.class)) {
            historyLinkContainer.add(new SimpleAttributeModifier("class", "current_page_item"));
        }
        historyLinkContainer.add(new BookmarkablePageLink("historyLink", HistoryPage.class));
        add(historyLinkContainer);
    }
}
```



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns:wicket="http://wicket.apache.org/">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
    <title>Mystic Paste</title>
    <link rel="stylesheet" type="text/css" href="css/style.css"/>
    <!--[if IE]>
      <link rel="stylesheet" type="text/css" href="css/ie.css" />
    <![endif]-->
  </head>
  <body>
    <div id="leftSide">&nbsp;</div>
    <div id="rightSide">&nbsp;</div>
    <div id="center">

      <!-- header -->
      <div id="header">
        <div id="logo"><a class="logo" href="http://projects.mysticcoders.com/paste"> </a></div>
      </div>
      <div id="nav">
        <ul id="menus">
          <li class="cat-item"><a class="home" title="Home" href="http://www.mysticcoders.com/">Home</a></li>
          <li wicket:id="newLinkContainer" class="cat-item"><a wicket:id="newLink" href="#" title="New Paste">New</a>
          <li wicket:id="historyLinkContainer" class="cat-item"><a wicket:id="historyLink" href="#" title="View Paste">
          <li><a class="lastmenu" href="javascript: return false;"> </a></li>
        </ul>
      </div>
      <div id="header_bottom"> </div>

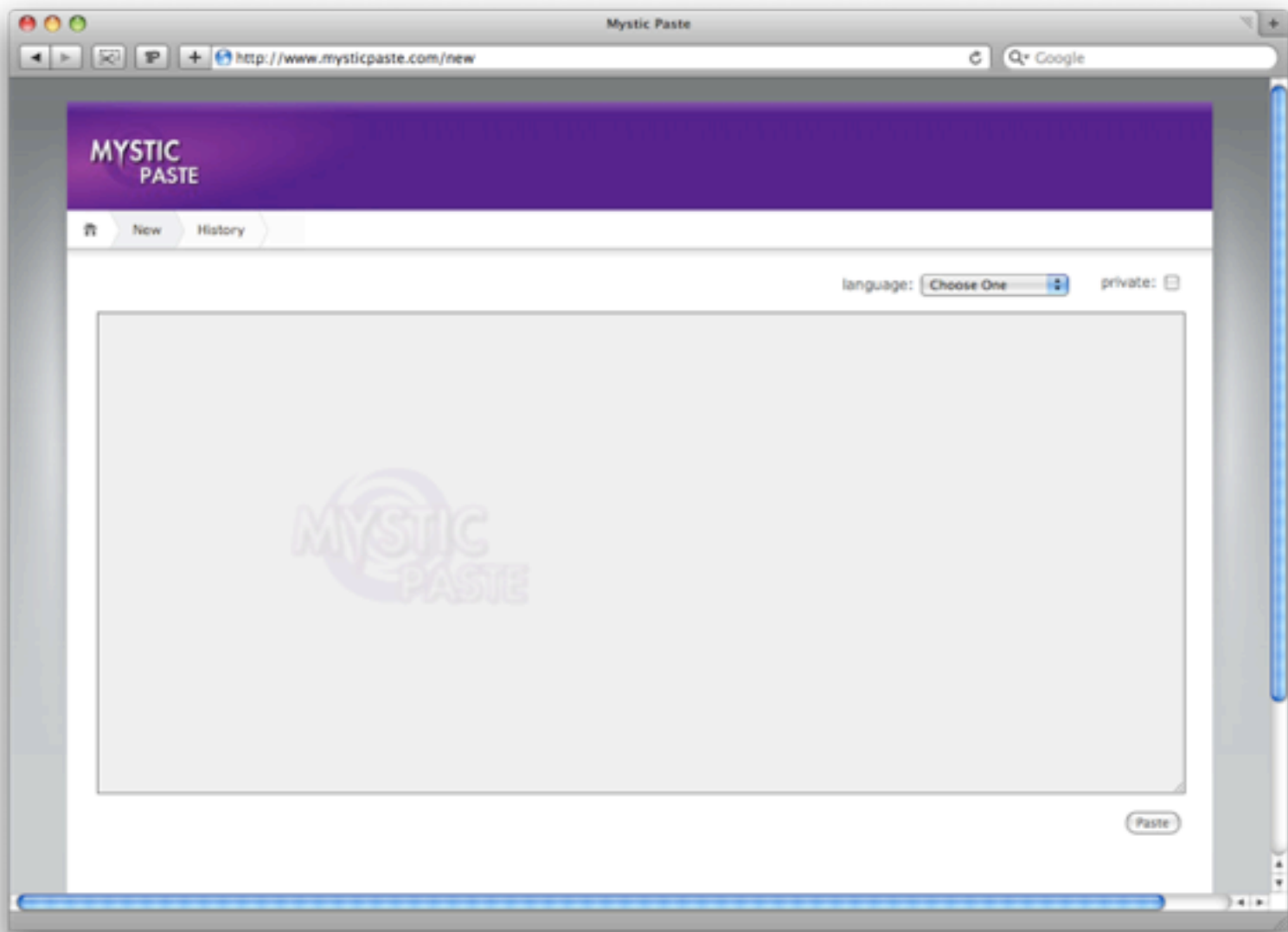
      <!-- content -->
      <div id="content">

        <wicket:child/>

      </div>
      <div class="clear"> </div>

      <!-- footer -->
      <div id="footer_left"> </div>
      <div id="footer_right"> </div>
      <div id="footer_center">
        <div id="copyright">Copyright &copy; 2000-2009 Mystic Coders, LLC</div>
      </div>
      <div id="logo_footer"></div>

    </body>
  </html>
```





```
public class PasteItemPage extends BasePage {

    @SpringBean
    PasteService pasteService;

    public PasteItemPage() {
        super(PasteItemPage.class);
        add(new PasteForm("pasteForm", new CompoundPropertyModel(new PasteItem())));
    }

    public class PasteForm extends Form {
        public PasteForm(String id, IModel model) {
            super(id, model);

            add(new CheckBox("private"));
            add(new DropDownChoice("type", Arrays.asList(LanguageType.JAVA, LanguageType.CSS, LanguageType.HTML)));
            add(new TextArea("content"));
        }

        @Override
        protected void onSubmit() {
            PasteItem pasteItem = (PasteItem) PasteForm.this.getModelObject();
            //todo: validation if any
            try {
                pasteService.createItem("web", pasteItem);
                PageParameters params = new PageParameters();
                if (pasteItem.isPrivate()) {
                    params.put("0", pasteItem.getPrivateToken());
                    setResponsePage(ViewPrivatePage.class, params);
                } else {
                    params.put("0", Long.toString(pasteItem.getId()));
                    setResponsePage(ViewPublicPage.class, params);
                }
            } catch (InvalidClientException e) {
                // Do nothing at this point as we haven't defined what an "Invalid Client" is.
                e.printStackTrace();
            }
        }
    }
}
```



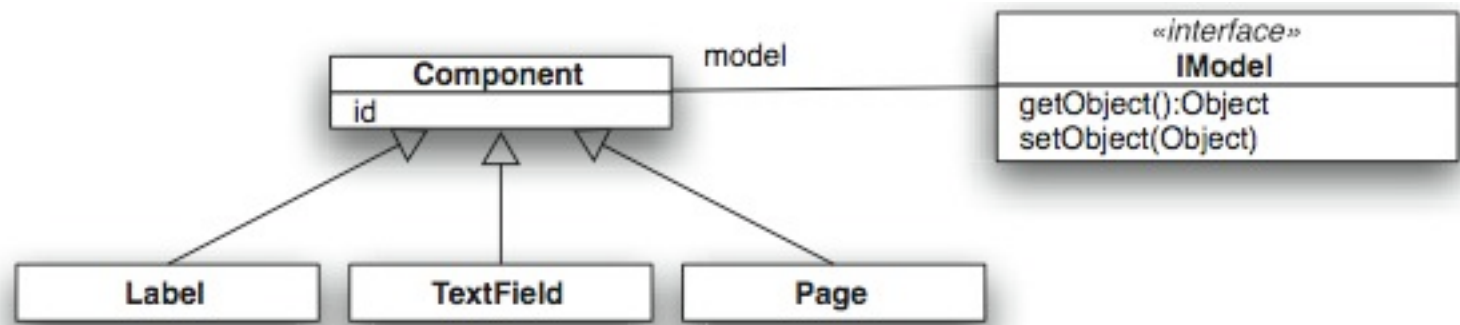

```
<html xmlns:wicket="http://wicket.apache.org/">
<wicket:extend>

  <form wicket:id="pasteForm">
    <div id="paste_options">
      <ul>
        <li>private:</li>
        <li><input wicket:id="private" type="checkbox" /></li>
      </ul>
      <ul>
        <li>language:</li>
        <li>
          <select wicket:id="type" class="language">
            <option>Choose One</option>
            <option>Java</option>
            <option>Groovy</option>
            <option>PHP</option>
          </select>
        </li>
      </ul>
    </div>
    <div id="paste_content">
      <div id="textLeft"> </div>
      <div id="textRight"> </div>
      <div id="textCenter"><textarea wicket:id="content" wrap="off"></textarea></div>
    </div>
    <div id="paste_submit"><input type="submit" value="Paste" /></div>
  </form>

</wicket:extend>
</html>
```

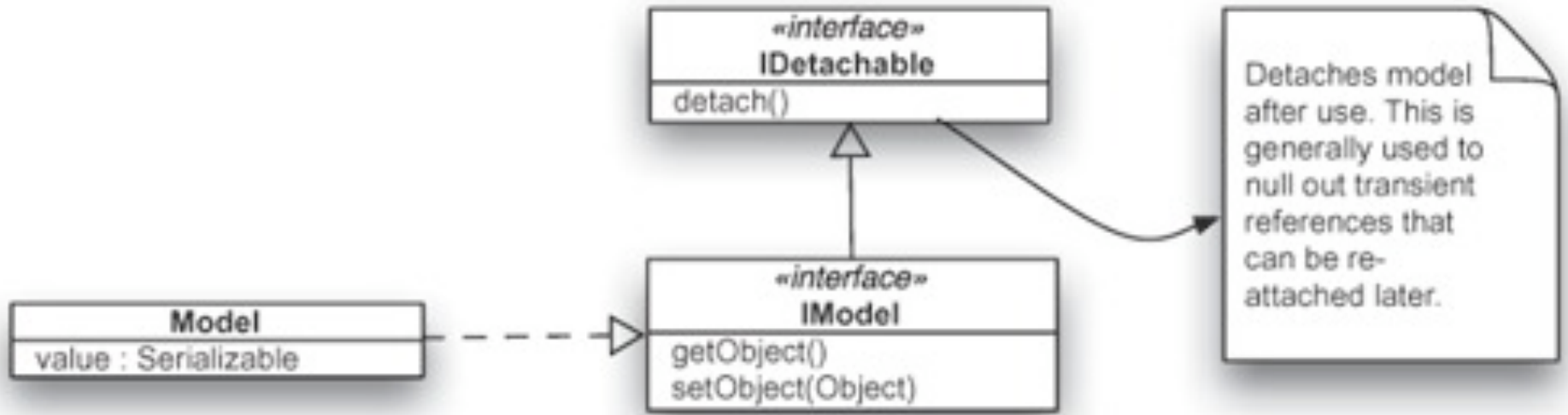
Model

the domain-specific representation of the information on which
the application operates.





Model	Description
<code>Model</code>	Simple model used to store static content, or used as a base class for quick dynamic behavior.
<code>PropertyModel</code>	Uses a property expression to dynamically access a property in your domain objects.
<code>CompoundPropertyModel</code>	Uses component identifiers as property expressions to bind components to its domain object.
<code>LoadableDetachableModel</code>	Abstract model for quickly creating detachable models.
<code>ResourceModel</code>	Easy-to-use model for retrieving messages from resource bundles.
<code>StringResourceModel</code>	Advanced model for retrieving messages from resource bundle; supports property expressions and <code>MessageFormat</code> substitutions.





LoadableDetachableModel



```
private class PasteModel extends LoadableDetachableModel {  
  
    String id;  
  
    public PasteModel(String id) {  
        this.id = id;  
    }  
  
    protected Object load() {  
        try {  
            return pasteService.getItem("web", Long.parseLong(id));  
        } catch (InvalidClientException e) {  
            // Do nothing as the validation is not yet implemented  
            e.printStackTrace();  
        }  
        return null;  
    }  
}
```



Component

an abstract part of something.



Panel



Reusable

capable of being used again.



DataTable



Showing 1 to 8 of 47

<< < 1 2 3 4 5 6 > >>

Actions	ID	First Name	Last Name	Home Phone	Cell Phone
select	32	Abby	Nelson	528-555-2757	770-555-4660
select	43	Abby	Bailey	768-555-8564	530-555-2364
select	34	Brianna	Fisher	701-555-7833	704-555-5175
select	47	Christopher	Lee	755-555-2234	414-555-6666
select	50	Christopher	Cruz	382-555-8835	654-555-7762
select	2	Debra	Ortiz	566-555-5021	736-555-8123
select	17	Debra	Cruz	804-555-7778	536-555-7211
select	30	Debra	Johnson	848-555-1871	685-555-7637





Pageable



Pageable

Sortable



Pageable

Sortable

Searchable



Pageable

Sortable

Searchable

Easy to Style



Pageable

Sortable

Searchable

Easy to Style

Filterable



```
add(historyDataView = new DataView("history", new HistoryDataProvider(pasteService), 10) {
    protected void populateItem(Item item) {
        PasteItem pasteItem = (PasteItem) item.getModelObject();

        PageParameters params = new PageParameters();
        params.put("0", Long.toString(pasteItem.getId()));
        item.add(new BookmarkablePageLink("viewLink", ViewPublicPage.class, params));

        final String[] contentLines = pasteItem.getContent().split("\n");
        item.add(new Label("lineCount", "(" + contentLines.length + " Line" +
            (contentLines.length > 1 ? "s" : "") + ")"));

        item.add(new Label("posted", getElapsedTimeSincePost(pasteItem)));

        List<String> lines = new ArrayList<String>();
        int count = 0;
        for (String contentLine : contentLines) {
            count++;
            if (count > 5) {
                break;
            } else {
                lines.add(contentLine);
            }
        }
        item.add(new ListView("content", lines) {
            protected void populateItem(ListItem item) {
                String content = (String) item.getModelObject();
                Label contentLine = new Label("contentLine", ((item.getIndex() + 1) +
                    " ").substring(0, 5) + content.replaceAll("\r", "").replaceAll("\n", ""));
                item.add(contentLine);
                if ((item.getIndex() + 1) % 2 == 0) {
                    item.add(new SimpleAttributeModifier("class", "highlight"));
                }
            }
        });

        item.add(new BookmarkablePageLink("viewLink2", ViewPublicPage.class, params) {
            @Override
            public boolean isVisible() {
                return contentLines.length > 5;
            }
        });
    }
});
```



```
public class HistoryDataProvider extends DefaultDataProvider {  
    PasteService pasteService;  
  
    public HistoryDataProvider(PasteService pasteService) {  
        this.pasteService = pasteService;  
    }  
  
    public Iterator iterator(int first, int count) {  
        try {  
            return pasteService.getLatestItems("web", count, first, false).iterator();  
        } catch (InvalidClientException e) {  
            e.printStackTrace();  
        }  
        return null;  
    }  
  
    public int size() {  
        try {  
            return new Long(pasteService.getLatestItemsCount("web")).intValue();  
        } catch (InvalidClientException e) {  
            e.printStackTrace();  
        }  
        return 0;  
    }  
  
    public IModel model(Object object) {  
        return new Model((PasteItem) object);  
    }  
}
```



Mystic Paste

http://www.mysticpaste.com/history

Google

MYSTIC PASTE

New History

<< < 1 2 3 4 5 6 7 > >>

View Paste (4 Lines) Posted 1 hour ago

1. Provides feedback **on** visual elements of the site
2. Defines page layout, navigation, **and** flow
3. Defines graphical design **and** branding
4. Builds the HTML prototype **for** all site web pages

View Paste (43 Lines) Posted 21 hours ago

1. Note: UK People start here: <http://twitter.com/timneh/status/1351833678>
- 2.
3. HOW TO ENABLE TETHERING ON YOUR 3.0 IPHONE
- 4.
5. Step 1: Check if you own a 3G iPhone. If you're on EDGE (as I am): sorry. Tethering does not seem to work with EDGE phones.

More...

View Paste (106 Lines) Posted 1 day ago

```

1. package com.mysticcoders.mysticpaste.web.pages.history;
2.
3. import com.mysticcoders.mysticpaste.web.pages.BasePage;
4. import com.mysticcoders.mysticpaste.web.pages.view.ViewPublicPage;
5. import com.mysticcoders.mysticpaste.services.PasteService;

```

Stateful

Capable of maintaining the status of a process or transaction.



```
public class ScriptaculousExamplesSession extends WebSession
{
    private List cartItems;

    /**
     * Construct.
     *
     * @param request
     */
    public ScriptaculousExamplesSession(Request request)
    {
        super(request);
    }

    /**
     * Gets cartItems.
     *
     * @return cartItems
     */
    public List getCartItems()
    {
        return cartItems;
    }

    /**
     * Sets cartItems.
     *
     * @param cartItems
     *        cartItems
     */
    public void setCartItems(List cartItems)
    {
        this.cartItems = cartItems;
    }
}
```




```
public class ScriptaculousExamplesApplication extends WebApplication
{
    public Session newSession(Request request, Response response)
    {
        return new ScriptaculousExamplesSession(request);
    }
}
```






JSF



JSF

the “Standard”



ORACLE®

Borland®
THE OPEN ALM COMPANY





design by committee



```
<input type="text" wicket:id="enterName" />
```

VS.

```
<h:inputText id="enterName" value="#{sayHelloPage.name}" />
```





Behavior in code

VS.

Behavior in markup





XML-based Configuration





Refactorability





Wicket: HTML and Java



Wicket: HTML and Java

JSF: JSP, HTML, JSF taglib, JSF EL
Java, and XML





the “Standard”

vs.

the Right Choice
for the problem

Q&A

Thanks for listening!

<http://mysticcoders.com/tssjsvegas2009>

<http://techtarget.sessionevals.sgizmo.com/>
THANKS FOR FILLING OUT THE EVAL

Andrew Lombardi

Owner, Tech Evangelist

Mystic Coders, LLC

andrew AT mysticcoders DOT com

kinabalu @ irc://irc.freenode.net - ##wicket, ##java