Wicket at a Glance

Justin Lee

http://www.antwerkz.com

jlee@antwerkz.com

Something Wicket This Way Comes

Justin Lee http://www.antwerkz.com

Walking in a Wicket Wonderland

Justin Lee http://www.antwerkz.com

Why Wicket?

* Object oriented
* Component oriented
* No parameter parsing
* Clean separation of HTML and Java code
* 0 XML configuration

What is Wicket?

* Page composition
  * Panels, Borders, Markup inheritance
* Excellent type conversion
* Localization support
  * _Be.html
  * resource Bundles _Be.properties
* components!

Justin Lee http://www.antwerkz.com

What Does Wicket Offer?

* Sortable, filterable, pageable tables
* Date picker Google Maps
* Integration
  * Spring
  * Hibernate
  * JasperReports
  * Portlets

What Does Wicket Offer?

* Automatic state management
  * Page maps
  * Back button support
  * No hidden form fields
  * Did you forget one?
* Testing support
  * Junit support
  * Logging and error reporting

What Does Wicket Offer?

* AJAX!
  * Dojo and scriptaculous
* Header contribution
  * Javascript and CSS
* URL Mounting
  * http://localhost:8181/app?wicket:interface
    =:1:lateToDoBox:lateToDos:0:view::1LinkListener
  * http://localhost:8181/app/view-event/id/1000

Components

* Everything in wicket is a component
* Identified in HTML by 'wicket:id'
* Referenced in Java by that ID

```
<span wicket:id="hello">[[replace me]]</span>

add(new Label("hello",
    new Model("Hello, World")));
```

Justin Lee http://www.antwerkz.com

Components

* Reusable
* Nestable
* One-to-one relationship between HTML components and wicket components
* Pages
* Panels
  * Used for grouping
* Borders

Application

* Base class of any Wicket application
* Define settings
  * Authorization
  * Render
  * AJAX
  * Exception handling
  * Session handling
* Servlet mapped to /app

Pages

* Typically extend wicket.markup.html.WebPage
  * Application specific subclass
* Markup inheritance
  * <wicket:child/>
* <wicket:extends>...</wicket:extends>
* Markup type can be changed
  * Override getMarkupType()

Markup

* 9 Wicket specific tags
* http://www.wicket-wiki.org.uk/wiki/index.php/Tags
* Everything else is pure (X)HTML
* <wicket:link>
* <wicket:panel>
* <wicket:Border> & <wicket:Body>
* <wicket:extends> & <wicket:child>
* <wicket:remove>
* <wicket:head>

Justin Lee http://www.antwerkz.com

Panels

* Extends wicket.markup.html.panel.Panel
* <wicket:panel>...</wicket:panel>
* Basic component building block

```
public class MyPanel extends Panel {
    public MyPanel() {
        add(new Label("hello", "Hello World"));
    }
}
```

Panels

```
<wicket:panel>
   <span wicket:id="hello">[[label]]</span>
</wicket:panel>

<wicket:panel>
   <form wicket:id="form">...</form>
</wicket:panel>
```

Justin Lee http://www.antwerkz.com

Models

* Very swing like
* Binds your data objects to the wicket components
* PropertyModel:

```
add(new TextField("email",
    new PropertyModel(person, "email")));

public class Person {
    private String email;
    ...
}
```

Models

CompoundPropertyModel:

```
setModel(new CompoundPropertyModel(person));
add(new TextField("email"));

// Person.java
public class Person {
    private String email;
    ...
}
```

Models

LoadableDetachableModel:

```
setModel(new PersonDetachableModel(person));
add(new TextField("email"));
```

Models

```
public class PersonDetachableModel {
    private Long id;
    private PersonService svc;
    public DetachedPersonModel(final Long personId,
      final PersonService personSvc) {
        this.id = personId;
        this.svc = personSvc;
    }
    protected Object load() {
        return svc.find(id);
    }
}
```

Justin Lee http://www.antwerkz.com

Site Navigation

* No global definition like struts, JSF, and RIFE
* Typically done with setReponsePage()
* replace(myPanel)
* RestartResponseException and RestartResponseAtInterceptPageException

Site Navigation

```
add(new Link("foo") {
    public void onClick() {
        setResponsePage(MyPage.class);
    }
}
```

Justin Lee http://www.antwerkz.com

Site Navigation

```
add(new Link("foo") {
    public void onClick() {
        setResponsePage(new MyPage(param));
    }
}
```

Justin Lee http://www.antwerkz.com

Site Navigation

```
add(new Link("foo") {
    public void onClick() {
        replaceWith(new MyPanel("foo"));
    }
}
```

Justin Lee http://www.antwerkz.com

Validation

* Added to form components directly
  * Required
  * Length
* Manually checked on submission

Validation

```
FormComponent field = new TextField("name")
    .setRequired(true);
add(field);
field.add(new AbstractValidator() {
    public void validate(FormComponent comp) {
        ...
        comp.error("whoops!");
    }
});
```

Justin Lee http://www.antwerkz.com

Validation

```
add(new AbstractFormValidator() {
    public void validate(Form form) {
        ...
        form.error("whoops!");
    }
});
```

Justin Lee http://www.antwerkz.com

AJAX!

* Native support
* Almost 0 javascript knowledge required
  * Need to know javascript events
* setOutputMarkupId(true);

Justin Lee http://www.antwerkz.com

AJAX!

```
ListChoice choice = new ListChoice(...);
choice.add(
    new AjaxFormComponentUpdatingBehavior("onchange") {
    @Override
    protected void onUpdate(AjaxRequestTarget target) {
        target.addComponent(otherChoice);
    }
});
```

Justin Lee http://www.antwerkz.com

AJAX!

```
<select wicket:id="first">[[options]]</select>
<select wicket:id="other">[[options]]</select>
```

Justin Lee http://www.antwerkz.com

AJAX!

* <pant/> <pant/>
* What was all that?
* Notice the javascript?

Justin Lee http://www.antwerkz.com

Internationalization

* Not as sexy but just as transparent
* Uses the user's locale as determined by the browser
* Markup pages are i18n aware:
  Page_fr_CA.html
  Page_fr.html
  Page.html

Internationalization

* ResourceModels can be used for labels and text
* add(new Label("prompt", new ResourceModel("login.prompt"));
* Properties files:
  Page_fr_CA.properties
  Page_fr.properties
  Page.properties

Spring

* wicket-spring & wicket-spring-annot
* Application extends SpringWebApplication or AnnotSpringWebApplication
* @SpringBean FooService service;

Lists

* Several options
  * ListView
  * DataView
  * DataTable
  * GridView

List View

* Probably the simplest

```
<tr wicket:id="items">
   <td>
      <span wicket:id="name">[[name]]</span>
   </td>
</tr>
```

Justin Lee http://www.antwerkz.com

List View

```
add(new ListView("items", items) {
    @Override
        protected void populateItem(ListItem listItem) {
            Item item = (Item)listItem.getModelObject();
            listItem.add(new Label("name",
                new Model(item.getName())));
        }
});
```

Justin Lee http://www.antwerkz.com

List View

* Pageable by setting the view size
  * Defaults to Integer.MAX_VALUE
* For static lists
  * Deletes will throw off the index
* Core Wicket component

DataView & DataTable

* More complicated but more powerful
* Data is provided through the IDataProvider interface
* DataView needs HTML written for it like ListView does
* DataTable provides its own
* DataTable requires a collection of IColumns

IDataProvider

```
public interface IDataProvider extends Serializable {
    Iterator iterator(int first, int count);
    int size();
    IModel model(Object object);
}
```

```
public class ContactDataProvider
    implements IDataProvider {
     public Iterator iterator(int first, int count)     {
         return getContactsDB().find(first, count,
           "firstName", true).iterator();
     }
     public int size() {
        return getContactsDB().getCount();
    }
     public IModel model(Object object)  {
         return new DetachableContactModel(
           (Contact)object);
     }
}
```

Justin Lee http://www.antwerkz.com

```
<table cellspacing="0" class="dataview">
  <tr wicket:id="simple">
    <td><span wicket:id="actions">[actions]</span></td>
    <td><span wicket:id="contactid">[contactid]</span></td>
    <td><span wicket:id="firstname">[firstname]</span></td>
    <td><span wicket:id="lastname">[lastname]</span></td>
    <td><span wicket:id="phone">[phone]</span></td>
    <td><span wicket:id="cell">[cell]</span></td>
  </tr>
</table>
```

Justin Lee http://www.antwerkz.com

```
add(new DataView("simple", new ContactDataProvider()) {
   protected void populateItem(final Item item) {
      Contact contact = (Contact)item.getModelObject();
      item.add(new ActionPanel("actions", item.getModel()));
      item.add(new Label("firstname",
        contact.getFirstName()));
      item.add(new Label("lastname",
         contact.getLastName()));
      item.add(new Label("phone",
         contact.getHomePhone()));
      item.add(new Label("cell",
        contact.getCellPhone()));
...
});
```

Justin Lee http://www.antwerkz.com

```
List columns = new ArrayList();
columns.add(new AbstractColumn(new Model("Actions")) {
  public void populateItem(Item cellItem,
   String componentId, IModel model) {
      cellItem.add(new ActionPanel(componentId, model));
  }
});
columns.add(new PropertyColumn(new Model("ID"), "id"));
columns.add(new PropertyColumn(
   new Model("First Name"), "firstName",
   "firstName"));
...
add(new DefaultDataTable("table", columns,
  new SortableContactDataProvider(), 8));
```

Justin Lee http://www.antwerkz.com

<table wicket:id="table">[table]</table>

Justin Lee http://www.antwerkz.com

GridView

```
GridView gridView =
    new GridView("rows", dataProvider) {
    protected void populateItem(Item item) { ... }
    protected void populateEmptyItem(Item item) { ... }
};

gridView.setRows(4);
gridView.setColumns(3);

add(gridView);
add(new PagingNavigator("navigator", gridView));
```

Justin Lee http://www.antwerkz.com

GridView

```
<span wicket:id="navigator">[dataview navigator]</span>
<table cellspacing="0" cellpadding="2" border="1">
  <tr wicket:id="rows">
    <td wicket:id="cols">
      <span wicket:id="firstName">[firstname]</span>
    </td>
  </tr>
</table>
```

Justin Lee http://www.antwerkz.com

DataView & DataTable

* So why one over the other?
* DataView allows total control over the table HTML
* DataTable has sorting and pagination built in
* DataTable supports header and footer toolbars
* GridView when you just want tabular display of single elements

Justin Lee http://www.antwerkz.com

Coming Attractions

* 2.0 will break the API
  * add(Component) goes away
  * No need to call replace() anymore
* Better session management
  * Stateless pages
* "Better" spring support in 1.2.2ish
* JSE 5 required
  * Generics!
* Filter vs. Servlet

Resources

* http://www.wicketframework.org
* http://www.wicketframework.org/Examples.html
* http://www.wicket-wiki.org.uk/wiki/index.php/Main_Page
* http://qwicket.sf.net
* Blogs
  * http://www.jroller.com/page/dashorst
  * http://chillenious.wordpress.com
  * http://www.jroller.com/page/JonathanLocke
  * http://www.antwerkz.com/wp

Questions?

Justin Lee http://www.antwerkz.com