MAKE YOUR GWT APPLICATION READY FOR GWT 3.0

Harald Pehl, GWTCon 2016

About Me

Senior Software Engineer at Red Hat

Component Lead of the WildFly Management Console

Working with GWT since 2008

http://hpehl.info

https://github.com/hpehl

@haraldpehl

HAL - WildFly Management Console

HAL.next

Reuse Business Logic

Rewrite most of the UI Code

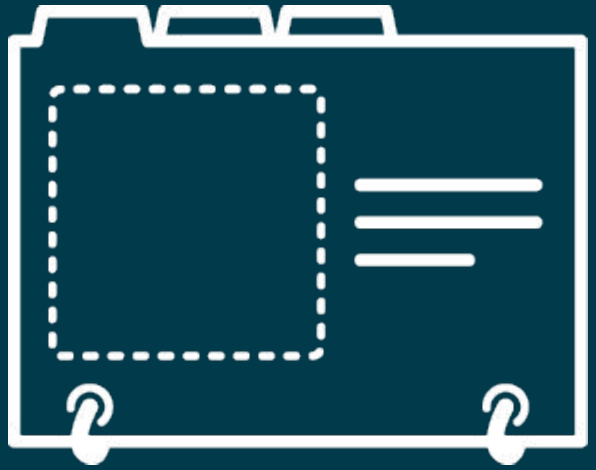
Demo

http://localhost:9990/

Get ready  for 3.0

GWT

DONT PANIC



User Interface

User Interface

Lightweight

No GWT Widgets

Builder & Templates

```
Elements.Builder builder = new Elements.Builder()
    .p()
        .a().css(clickable, pullRight).on(click, event -> update())
            .span().css(fontAwesome("refresh"), marginRight5).end()
            .span().textContent(resources.constants().refresh()).end()
        .end()
    .end()
    .section().css(centerBlock).rememberAs(LOADING_SECTION)
        .p().textContent(resources.constants().loading()).end()
        .div().css(spinner, spinnerLg).end()
    .end()
    .section().rememberAs(TOPOLOGY_SECTION).end();
```

```
<div data-element="root" class="row">
   <div class="col-lg-12 col-md-12 col-sm-12">
       <h1 data-element="header">server.log</h1>
       <div data-element="logFileControls" class="editor-controls">
           <div data-element="search"></div>
           <div data-element="status" class="editor-status"></div>
           <div class="log-file-follow">
               <label for="log-file-follow">{{resources().constants().tailMode()}}</label>
               <input data-element="tailMode" class="bootstrap-switch" type="checkbox"/>
           </div>
           <div class="editor-buttons btn-group">
               <a data-element="refresh" class="btn btn-default clickable">
                 <i class="fa fa-refresh"></i>
             </a>
               <a data-element="copyToClipboard" class="btn btn-default clickable">
                 <i class="fa fa-clipboard"></i>
             </a>
           </div>
       </div>
       <div data-element="editorContainer" class="margin-bottom-large log-file-editor-container">
           <div data-element="loading" class="spinner spinner-lg"></div>
       </div>
   </div>
</div>
```

```java
@Templated("LogFileView.html#root")
public abstract class LogFileView {

    public static LogFileView create() {
        return new Templated_LogFileView();
    }

    private Search search;

    @DataElement Element header;
    @DataElement Element tailMode;

    @DataElement("search")
    Search setupSearch() {
        ...
    }

    @EventHandler(element = "refresh", on = click)
    void onRefresh() {
        presenter.refresh();
    }
}
```

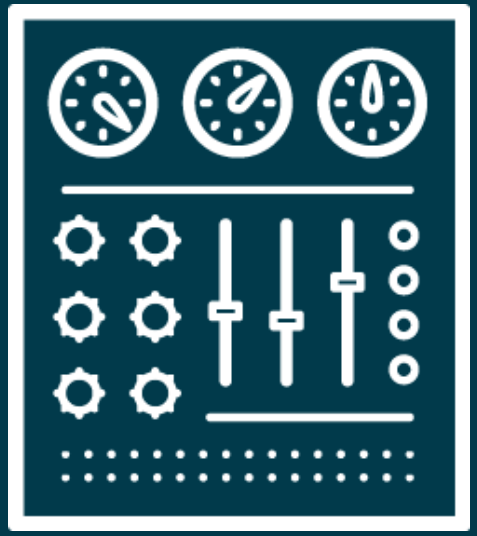
Generate UI

Reduce Boilerplate

Annotation Processor

Pragmatic

```
<view>
    <metadata address="/{selected.profile}/subsystem=ejb3/thread-pool=*">
        <h1>Thread Pool</h1>
        <p>{{metadata.getDescription().getDescription()}}</p>
        <table id="thread-pool-table" title="Thread Pool" form-ref="thread-pool-form">
            <actions>
                <action handler-ref="add-resource"/>
                <action handler-ref="remove-resource" scope="selected"
                    name-resolver="{{api.selectedRow().getName()}}"/>
            </actions>
            <columns>
                <column name="name" value="{{row.getName()}}"/>
            </columns>
        </table>
        <form id="thread-pool-form" title="Thread Pool" auto-save="true"
            name-resolver="{{form.getModel().getName()}}"/>
     </metadata>
</view>
```



Widgets

Widgets

KISS

DOM Elements

Web Components

Custom Elements

JsInterop

PatternFly

Bootstrap

Datatables

jsTree

AceEditor

```java
public class Breadcrumb implements IsElement {

    private final Element root;

    public Breadcrumb() {
        root = Browser.getDocument().createElement("ol");
        root.getClassList().add(breadcrumb);
    }

    public Breadcrumb append(final String segment) {
        Element li = new Elements.Builder()
            .li().css(active).textContent(segment).end()
            .build();
        root.appendChild(li);
        return this;
    }

    @Override
    public Element asElement() {
        return root;
    }
}
```

```java
@JsType(isNative = true)
public abstract class Modal {

   @JsFunction @FunctionalInterface
   public interface ModalHandler {
       void handle();
   }

   @JsType(isNative = true, namespace = GLOBAL, name = OBJECT)
   public static class ModalOptions {

       public boolean keyboard;

       @JsOverlay
       public static ModalOptions create(final boolean closeOnEsc) {
           ModalOptions options = new ModalOptions();
           options.keyboard = closeOnEsc;
           return options;
       }
   }

   @JsMethod(namespace = GLOBAL)
   public native static Modal $(@NonNls String selector);

   public native void modal(ModalOptions modalOptions);

   public native void on(@NonNls String event, ModalHandler handler);
}
```

CSS / Styling

CSS / Styling

Less / Sass

Follow best practices

- http://getbem.com/
- http://cssguidelin.es/

Interface with class constants

```
public interface CSS {

    String address = "address";
    String active = "active";
    String alert = "alert";
    String alertDanger = "alert-danger";
    String alertInfo = "alert-info";
    String alertSuccess = "alert-success";

    ...

    String warning = "warning";
    String warningTriangleO = "warning-triangle-o";
    String withProgress = "with-progress";

    static String fontAwesome(@NonNls String name, FontAwesomeSize size) {
        @NonNls String css = "fa fa-" + name;
        if (size != null) {
            css += " fa-" + size.size();
        }
        return css;
    }
}
```

```
Elements.Builder builder = new Elements.Builder()
    .form().attr("novalidate", "true").css(upload)
        .add(alert)
        .div().rememberAs(ICON_ELEMENT).css(uploadIcon, fontAwesome("upload", x4)).end()
        .input(file)
            .rememberAs(FILE_INPUT_ELEMENT)
            .id(Ids.UPLOAD_FILE_INPUT)
            .css(uploadFile)
            .on(change, event -> showFiles(fileInput.getFiles()))
        .label().attr("for", Ids.UPLOAD_FILE_INPUT).rememberAs(LABEL_ELEMENT)
            .a().css(clickable)
                .textContent(CONSTANTS.chooseFile())
            .end()
            .span().rememberAs(DRAG_ELEMENT)
                .textContent(CONSTANTS.orDragItHere())
            .end()
        .end()
    .end();
```

Model View Presenter

MVP

GWTP

- DI through GIN
- Simple and powerful history management
- Support for nested presenters
- Lifecycle events using GWT eventbus
- Lazy instantiation for presenters and views
- Effortless and efficient code splitting

Widget ↔ Element Adapter

```java
/**
 * Adapter between GWTPs views which are based on widgets and
 * HAL views which are based on elements.
 *
 * @author Harald Pehl
 */
public interface HalView extends View, IsElement, HasElements {

    /**
     * This method should be called <em>after</em> the view's elements are attached to the DOM.
     * Typically this method is called from {@link HalPresenter#onReveal()}.
     */
    void attach();

    /**
     * Counterpart to {@link #attach()}. Implement this method if you need to remove stuff
     * which was setup in {@link #attach()}. The default implementation does nothing.
     */
    default void detach() {}
}
```

GWT-RPC / RequestFactory

GWT-RPC / RequestFactory

Plain XmlHttpRequest

JSON, XML, ... as payload

Central class

- Provides high level API
- Contains low level code

```
public class Dispatcher {

   public void execute(final Composite composite, final CompositeCallback callback) {
      ...
   }

   public void execute(final Operation operation, final OperationCallback callback) {
      ...
   }

   public void upload(final File file, final Operation operation,
           final OperationCallback callback) {
      ...
   }
}
```

Deferred Binding

Deferred Binding

Replace with Annotation Processing

Use 3rd party libraries

- https://github.com/hpehl/apt-tools
- https://github.com/google/auto
- https://github.com/google/compile-testing

```
<#-- @ftlvariable name="generatedWith" type="java.lang.String" -->
<#-- @ftlvariable name="packageName" type="java.lang.String" -->
<#-- @ftlvariable name="className" type="java.lang.String" -->
<#-- @ftlvariable name="bindings" type="java.util.List<org.jboss.hal.processor.RegistryBinding>" -->
package ${packageName};

import javax.annotation.Generated;

import com.google.inject.Singleton;
import com.google.gwt.inject.client.AbstractGinModule;

import org.jboss.hal.spi.GinModule;

/*
 * WARNING! This class is generated. Do not modify.
 */
@GinModule
@Generated("${generatedWith}")
public class ${className} extends AbstractGinModule {

   @Override
   protected void configure() {
       <#list bindings as binding>
       bind(${binding.interface}.class).to(${binding.implementation}.class).in(Singleton.class);
       </#list>
   }
}
```

```java
public class FormTest {

    @Test
    public void simpleForm() {
        Compilation compilation = javac()
                .withOptions("-proc:only")
                .withProcessors(new MbuiViewProcessor())
                .compile(JavaFileObjects.forResource("SimpleForm.java"));

        assertThat(compilation)
                .generatedSourceFile("Mbui_SimpleForm")
                .hasSourceEquivalentTo(
                    JavaFileObjects.forResource("Mbui_SimpleForm.java"));
    }
}
```

Debug / Run / Build

Debug / Run / Build

Debug: Browser Dev Tools

Run & Build

- Maven Multi Module
- Maven Plugin for GWT
- Frontend Maven Plugin
    - NPM
    - Bower
    - Grunt

```
{
  "name": "org.jboss.hal.bower",
  "version": "0.4.3",
  "description": "Bower setup for HAL",
  "license": "Apache-2.0",
  "dependencies": {
    "ace-builds": "^1.2.4",
    "datatables.net": "~1.10.12",
    "datatables.net-buttons": "~1.2.2",
    "datatables.net-keytable": "^2.1.3",
    "datatables.net-select": "~1.2.0",
    "javascript-auto-complete": "1.0.4",
    "jquery": "~2.2.4",
    "jstree": "~3.3.1",
    "patternfly": "~3.14.0",
    "tagmanager": "~3.0.2",
    "zeroclipboard": "^2.2.0"
  }
}
```

What’s left?

What's Left

ClientBundle / i18n

Dependency Injection

Thanks!
Questions?