

# GWT UiBinder framework, Code splitting

# UiBinder framework

- Назначение UiBinder
- Преимущества использования
- UiBinder framework позволит...
- Примеры
- Code splitting

# Назначение UiBinder

Разработка экранных форм пользовательского интерфейса GWT приложений привычным способом компоновки HTML страниц

```
<!-- HelloWorld.ui.xml -->

<ui:UiBinder xmlns:ui='urn:ui:com.google.gwt.uibinder'>
  <div>
    Hello, <span ui:field='nameSpan' />.
  </div>
</ui:UiBinder>
```

# Преимущества

- + декларативный способ построения экранных форм пользовательского интерфейса является более знакомым и привычным способом для web-разработчиков;
- + экранные формы пользовательского интерфейса, построенные с помощью UIBinder более эффективно обрабатываются браузером.

# UiBinder framework позволит...

1. увеличить производительность и управляемость;
2. проще взаимодействовать с дизайнерами;
3. предоставить постепенный переход при разработке от HTML макетов к настоящему интерактивному интерфейсу пользователя;
4. разделить представление экранных форм от программирования поведения;
5. выполнять во время компилирования контроль перекрестных ссылок из Java кода к XML и в обратном направлении.

# Простейший пример

```
public class HelloWorld {  
    interface MyUiBinder extends UiBinder<DivElement, HelloWorld> {}  
    private static MyUiBinder uiBinder = GWT.create(MyUiBinder.class);  
  
    @UiField SpanElement nameSpan;  
  
    public HelloWorld() {  
        setElement(uiBinder.createAndBindUi(this));  
    }  
  
    public void setName(String name) { nameSpan.setInnerText(name); }  
  
    public void Element getElement() { return nameSpan; }  
}
```

# Простейший пример: продолжение

```
HelloWorld helloWorld = new HelloWorld();  
Document.getDocument().getBody().appendChild(helloWorld.getElement());  
helloWorld.setName("World");
```

# Простейший Widget-пример

```
<!-- HelloWorldWorld.ui.xml -->

<ui:UiBinder xmlns:ui='urn:ui:com.google.gwt.uibinder'
  xmlns:g='urn:import:com.google.gwt.user.client.ui'>

  <g:HTMLPanel>
    Hello, <g:ListBox ui:field='listBox' visibleItemCount='1'/.
  </g:HTMLPanel>

</ui:UiBinder>
```



# Простейший Widget-пример: продолжение

```
public class HelloWidgetWorld extends Composite {

    interface MyUiBinder extends UiBinder<Widget, HelloWidgetWorld> {}
    private static MyUiBinder uiBinder = GWT.create(MyUiBinder.class);

    @UiField ListBox listBox;

    public HelloWidgetWorld(String... names) {
        // sets listBox
        initWidget(uiBinder.createAndBindUi(this));
        for (String name : names) {
            listBox.addItem(name);
        }
    }
}

// Use:

HelloWidgetWorld helloWorld =
    new HelloWidgetWorld("able", "baker", "charlie");
```

# Пример использования панелей

```
<ui:UiBinder xmlns:ui='urn:ui:com.google.gwt.uibinder'  
xmlns:g='urn:import:com.google.gwt.user.client.ui'>  
  
  <g:HorizontalPanel>  
    <g:Label>Keep your ducks</g:Label>  
    <g:Label>in a row</g:Label>  
  </g:HorizontalPanel>  
  
</ui:UiBinder>
```

# Пример использования панелей: продолжение

```
<g:DockLayoutPanel unit='EM'>
  <g:north size='5'>
    <g:Label>Top</g:Label>
  </g:north>
  <g:center>
    <g:Label>Body</g:Label>
  </g:center>
  <g:west size='10'>
    <g:HTML>
      <ul>
        <li>Sidebar</li>
        <li>Sidebar</li>
        <li>Sidebar</li>
      </ul>
    </g:HTML>
  </g:west>
</g:DockLayoutPanel>
```

# Пример обработки событий: «как обычно»

```
public class MyFoo extends Composite {  
    Button button = new Button();  
  
    public MyFoo() {  
        button.addClickHandler(new ClickHandler() {  
            public void onClick(ClickEvent event) {  
                handleClick();  
            }  
        });  
        initWidget(button);  
    }  
  
    void handleClick() {  
        Window.alert("Hello, AJAX");  
    }  
}
```

# Пример обработки событий: в UiBinder

```
public class MyFoo extends Composite {  
    @UiField Button button;  
  
    public MyFoo() {  
        initWidget(button);  
    }  
  
    @UiHandler("button")  
    void handleClick(ClickEvent e) {  
        Window.alert("Hello, AJAX");  
    }  
}
```

# Widget-пример «конструктор с аргументом»

```
public CricketScores(String... teamNames) {...}
```

```
<!-- UserDashboard.ui.xml -->
```

```
<ui:UiBinder
```

```
xmlns:ui='urn:ui:com.google.gwt.uibinder'
```

```
xmlns:g='urn:import:com.google.gwt.user.client.ui'
```

```
xmlns:my='urn:import:com.my.app.widgets' >
```

```
<g:HTMLPanel>
```

```
<my:WeatherReport ui:field='weather' />
```

```
<my:Stocks ui:field='stocks' />
```

```
<my:CricketScores ui:field='scores' />
```

```
</g:HTMLPanel>
```

```
</ui:UiBinder>
```

# Widget-пример «конструктор с аргументом»: продолжение

```
public class UserDashboard extends Composite {  
    interface MyUiBinder extends UiBinder<Widget, UserDashboard> {}  
    private static MyUiBinder uiBinder = GWT.create(MyUiBinder.class);  
  
    public UserDashboard() {  
        initWidget(uiBinder.createAndBindUi(this));  
    }  
}
```

# Widget-пример «конструктор с аргументом»: ошибка

```
[ERROR] com.my.app.widgets.CricketScores has no default (zero args) constructor. To fix this, you can define a @UiFactory method on the UiBinder's owner, or annotate a constructor of CricketScores with @UiConstructor.
```



# Widget-пример «конструктор с аргументом»: @UiFactory метод

```
public class UserDashboard extends Composite {
    interface MyUiBinder extends UiBinder<Widget, UserDashboard>;
    private static final MyUiBinder uiBinder = GWT.create(MyUiBinder.class);

    private final String[] teamNames;

    public UserDashboard(String... teamNames) {
        this.teamNames = teamNames;
        initWidget(uiBinder.createAndBindUi(this));
    }

    /** Used by MyUiBinder to instantiate CricketScores */
    @UiFactory CricketScores makeCricketScores() { // method name is insignificant
        return new CricketScores(teamNames);
    }
}
```

# Widget-пример «конструктор с аргументом»: @UiConstructor

```
public @UiConstructor CricketScores(String teamNames) {  
    this(teamNames.split("[, ]+"));  
}
```

```
<!-- UserDashboard.ui.xml -->  
<g:HTMLPanel xmlns:ui='urn:ui:com.google.gwt.uibinder'  
    xmlns:g='urn:import:com.google.gwt.user.client.ui'  
    xmlns:my='urn:import:com.my.app.widgets' >  
  
    <my:WeatherReport ui:field='weather'/>  
    <my:Stocks ui:field='stocks'/>  
    <my:CricketScores ui:field='scores' teamNames='AUS, SAF, WA, QLD, VIC'/>  
  
</g:HTMLPanel>
```

# Widget-пример «конструктор с аргументом»: @UiField(provided=true)

```
public class UserDashboard extends Composite {  
    interface MyUiBinder extends UiBinder<Widget, UserDashboard>;  
    private static final MyUiBinder uiBinder = GWT.create(MyUiBinder.class);  
  
    @UiField(provided=true)  
    final CricketScores cricketScores; // cannot be private  
  
    public UserDashboard(CricketScores cricketScores) {  
        // DI fans take note!  
        this.cricketScores = cricketScores;  
        initWidget(uiBinder.createAndBindUi(this));  
    }  
}
```

# Пример CSS стилизации

```
<ui:UiBinder xmlns:ui='urn:ui:com.google.gwt.uibinder'>

  <ui:style>
    .pretty { background-color: Skyblue; }
  </ui:style>

  <div class='{style.pretty}'>
    Hello, <span ui:field='nameSpan'/>.
  </div>

</ui:UiBinder>
```

# Пример CSS стилизации: продолжение

```
<ui:UiBinder xmlns:ui='urn:ui:com.google.gwt.uibinder'>
  <ui:style>
    .pretty { background-color: Skyblue; }
  </ui:style>

  <ui:style field='otherStyle'>
    .pretty { background-color: Orange; }
  </ui:style>

  <div class='{style.pretty}'>
    Hello, <span class='{otherStyle.pretty}'
ui:field='nameSpan' />.
  </div>

</ui:UiBinder>
```

# Пример ссылки на CSS-файл

```
<ui:UiBinder xmlns:ui='urn:ui:com.google.gwt.uiBinder'>
  <ui:style src="MyUi.css" />
  <ui:style field='otherStyle' src="MyUiOtherStyle.css">

  <div class='{style.pretty}'>
    Hello, <span class='{otherStyle.pretty}' ui:field='nameSpan' />.
  </div>
</ui:UiBinder>
```

# Пример применения CSS правил к виджету

```
<ui:UiBinder xmlns:ui='urn:ui:com.google.gwt.uibinder'  
  xmlns:g='urn:import:com.google.gwt.user.client.ui'>  
  <ui:style>  
    .hot { color: magenta; }  
    .pretty { background-color: Skyblue; }  
  </ui:style>  
  
  <g:PushButton styleName='{style.pretty}'>This button doesn't look like one</g:PushButton>  
  <g:PushButton addStyleNames='{style.pretty} {style.hot}'>Push my hot button!</g:PushButton>  
  
</ui:UiBinder>
```

# Пример программного изменения CSS правил

```
<ui:UiBinder xmlns:ui='urn:ui:com.google.gwt.uibinder'>

  <ui:style type='com.my.app.MyFoo.MyStyle'>
    .redBox { background-color:pink; border: 1px solid red; }
    .enabled { color:black; }
    .disabled { color:gray; }
  </ui:style>

  <div class='{style.redBox} {style.enabled}'>I'm a red box widget.</div>

</ui:UiBinder>
```



# Пример программного изменения CSS правил: продолжение

```
public class MyFoo extends Widget {
    interface MyStyle extends CssResource {
        String enabled();
        String disabled();
    }

    @UiField MyStyle style;

    /* ... */

    void setEnabled(boolean enabled) {
        getElement().addClassName(enabled ? style.enabled() : style.disabled());
        getElement().removeClassName(enabled ? style.disabled() : style.enabled());
    }
}
```

# Импорт HTML сущностей в XML шаблон

```
<!DOCTYPE ui:UiBinder SYSTEM "http://dl.google.com/gwt/DTD/xhtml.ent">
```

# Узнать больше

<http://code.google.com/intl/en/webtoolkit/doc/latest/DevGuideUiBinder.html>

Code splitting

<http://code.google.com/intl/en/webtoolkit/doc/latest/DevGuideCodeSplitting.html>

# Практическое задание

Создать приложение «Учет контактов» с использованием GWT UiBinder framework