
Google™
10

GWT's UI overhaul: UiBinder, ClientBundle, and Layout Panels

Ray Ryan, Joel Webber
May 2010

View live notes and ask questions about this session on Google Wave

http://bit.ly/io2010-gwt5

A lot new in GWT 2.0 and beyond

- UiBinder
- ClientBundle (especially CssResource)
- LayoutPanel

was: *http://gwt.google.com/samples/Mail-1.5.2/Mail.html*

is: *http://gwt.google.com/samples/Mail-2.0.0/Mail.html*

- Cell Widgets (coming in GWT 2.1)

UiBinder ♥ ClientBundle

UiBinder: Declarative User Interfaces

Write your HTML in HTML

Boilerplate begone

Fewer widgets, smaller footprint

Compare UI in Java (squint)…

```
HorizontalPanel outer = new HorizontalPanel();
VerticalPanel inner = new VerticalPanel();

outer.setHorizontalAlignment(HorizontalPanel.ALIGN_RIGHT);
inner.setHorizontalAlignment(HorizontalPanel.ALIGN_RIGHT);

HorizontalPanel links = new HorizontalPanel();
links.setSpacing(4);
links.add(signOutLink);
links.add(aboutLink);

final Image logo = images.logo().createImage();
outer.add(logo);
outer.setCellHorizontalAlignment(logo, HorizontalPanel.ALIGN_LEFT);

outer.add(inner);
inner.add(new HTML("<b>Welcome back, foo@example.com</b>"));
inner.add(links);

signOutLink.addClickHandler(this);
aboutLink.addClickHandler(this);

initWidget(outer);
setStyleName("mail-TopPanel");
links.setStyleName("mail-TopPanelLinks");
```

…to UI in HTML

```
<g:HTMLPanel>
  <div class='{style.logo}'/>

  <div class="{style.statusDiv}">
    <div>
      <b>Welcome back, foo@example.com</b>
    </div>

    <div class='{style.linksDiv}'>
      <g:Anchor ui:field='signOutLink'>Sign Out</g:Anchor>
      <g:Anchor ui:field='aboutLink'>About</g:Anchor>
    </div>
  </div>
</g:HTMLPanel>
```

Let's wire it up

```
<g:HTMLPanel>
  <div class='{style.logo}'/>

  <div class="{style.statusDiv}">
    <div>
      <b>Welcome back, foo@example.com</b>
    </div>

    <div class='{style.linksDiv}'>
      <g:Anchor ui:field='signOutLink'>Sign Out</g:Anchor>
      <g:Anchor ui:field='aboutLink'>About</g:Anchor>
    </div>
  </div>
</g:HTMLPanel>
```

Inner class? What inner class?

```
@UiField Anchor signOutLink;
@UiField Anchor aboutLink;

public TopPanel( ) {
  initWidget(Binder.createAndBindUi(this));
}

@UiHandler("signOutLink")
void onSignOut(ClickEvent e) {
  /* ... do sign out ... */
}

@UiHandler("aboutLink")
void onAbout(ClickEvent e) {
  /* ... show about box ... */
}
```

ClientBundle

Bake your resources into your code

- CSS
- Images
- Text
- Whatever you come up with next

Fewer round trips, lower latency
Built in image spriting
Name spaces
Not your grandfather's CSS

Hello <ui:style>

```
<ui:UiBinder
    xmlns:ui='urn:ui:com.google.gwt.uibinder'>
  <ui:style>
    .main { color: Red; }
  </ui:style>

  <div class={style.main}>Hello world.</div>
</ui:Binder>
```

Inline is cute, but...

```
<ui:UiBinder
    xmlns:ui='urn:ui:com.google.gwt.uibinder'>
  <ui:style src='my.css' />

  <div class='{style.main}'>Hello world.</div>
</ui:Binder>
```

Pretty piccies

```
<ui:UiBinder
    xmlns:ui='urn:ui:com.google.gwt.uibinder'>
  <ui:image field='bgImage'
     repeatStyle='Both'/>

  <ui:style src='my.css'>
    @sprite .bgnd { gwt-image: "bgImage"; }
  </ui:style>

  <div class='{style.bgnd}'>Hello world.</div>
</ui:Binder>
```

Bonjour, tout le monde

```
<ui:UiBinder
    xmlns:ui='urn:ui:com.google.gwt.uibinder'>
  <ui:image field='bgImage'
      repeatStyle='Both'/>

  <ui:style src='my.css'>
    @sprite .bgnd { gwt-image: "bgImage"; }
  </ui:style>

  <div class='{style.bgnd}'>
    <ui:msg>Hello world.</ui:msg>
  </div>
</ui:Binder>
```

Bonjour, tout les widgets

```
<g:HTMLPanel>
  <ui:msg>
    To do the thing, <g:Hyperlink targetHistoryToken="
/doThe#thing"
      >click here</g:Hyperlink>
    and massage vigorously.
  </ui:msg>
</g:HTMLPanel>
```

```
# 0=arg0 (Example: <span>), 1=arg1 (Example: </span>)
8EF…094=To do the thing, {0}click here{1} and massage
vigorously.
```

The sad truth: Issue 4335

- **"Dealing with UiBinder i18n properties files is kind of a nightmare"**

http://code.google.com/p/google-web-toolkit/issues/detail?id=4355

- VERY configurable, VERY confusing

- What we do in our apps 'round these parts
  - Use md5 hashes
  - Gather the generated .properties files into one big one
  - Translators provide one alternative .properties per locale

- Docs have been updated with the details

http://code.google.com/webtoolkit/doc/latest/DevGuideUiBinderI18n.html

Layout Overhaul

Background

- Old GWT layout widgets attempt to build static structures
- Behavior isn't very predictable
  - works fine, except when it doesn't
- Doesn't work well in standards mode
- Alternatives (e.g., GXT/JS) implement all layout in JS
  - Predictable and flexible, but quite slow
- Turns out you can do a remarkable amount with CSS:
  - top: left: right: bottom: width: height:
  - Simple but effective constraint system
  - Much in common with Cocoa "springs & struts"

What is it?

- Replacement for "application-level" layout widgets
- Absolutely predictable
- Standards mode
- Supports animation
- Supports all browsers (even IE6)

What is it *not*?

- Replacement for all HTML layout
- Swing/SWT-style layout based on 'preferred' sizes

Demo: Mail Sample

Under the hood

```
<g:DockLayoutPanel unit='EM'>
   <g:north size='5'>
    <mail:TopPanel ui:field='topPanel' />
   </g:north>

   <g:center>
    <g:SplitLayoutPanel>
     <g:west size='192'>
      <mail:Shortcuts ui:field='shortcuts' />
     </g:west>

     <g:north size='200'>
      <mail:MailList ui:field='mailList' />
     </g:north>

     <g:center>
      <mail:MailDetail ui:field='mailDetail' />
     </g:center>
    </g:SplitLayoutPanel>
   </g:center>
 </g:DockLayoutPanel>
```

Constraints

- Not sensitive to 'preferred' size
- Often requires explicit sizes (but in arbitrary units)
- Provides resize events
  - But only for outermost layout widgets
- Not appropriate for fine-grained layout
  - e.g., forms should still be laid out using HTML
  - What works well in HTML, should stay in HTML

Benefits

- Extremely fast
- Browser layout engine does most of the work
- No more window resize listener
  - Layout doesn't lag resize on Firefox!
- Automatically accounts for margins, borders, and padding
- Automatically reacts to font size changes
  - (when using 'em' or 'ex' units)
- Built-in support for animation
- Predictable and well-tested

Lists, Tables, and Trees, oh my!

The Problem

Rendering large numbers of widgets is slow

- innerHTML hard to use, but will always be faster

Existing collection widgets allow *too much* flexibility

- You really want to put a TabPanel in a TreeItem?

Need a way to render simple items efficiently

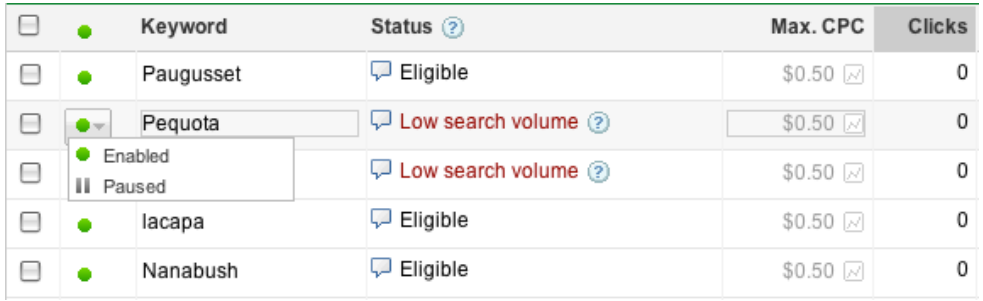
Need to support interactive items:

- Buttons, checkboxes, text input, date pickers, etc.

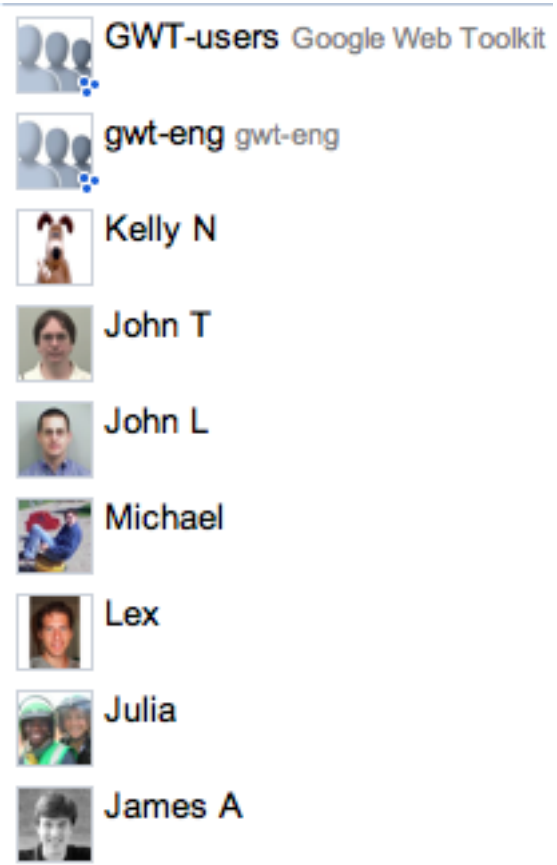
The Widgets

Tables

Trees



Lists



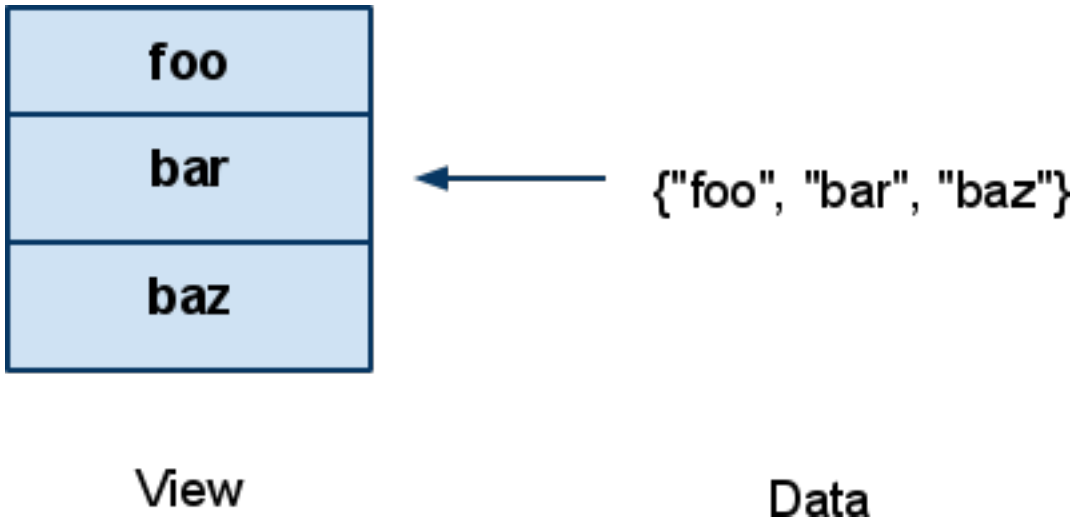
Google 10 I/O

Demo: Expense Reports

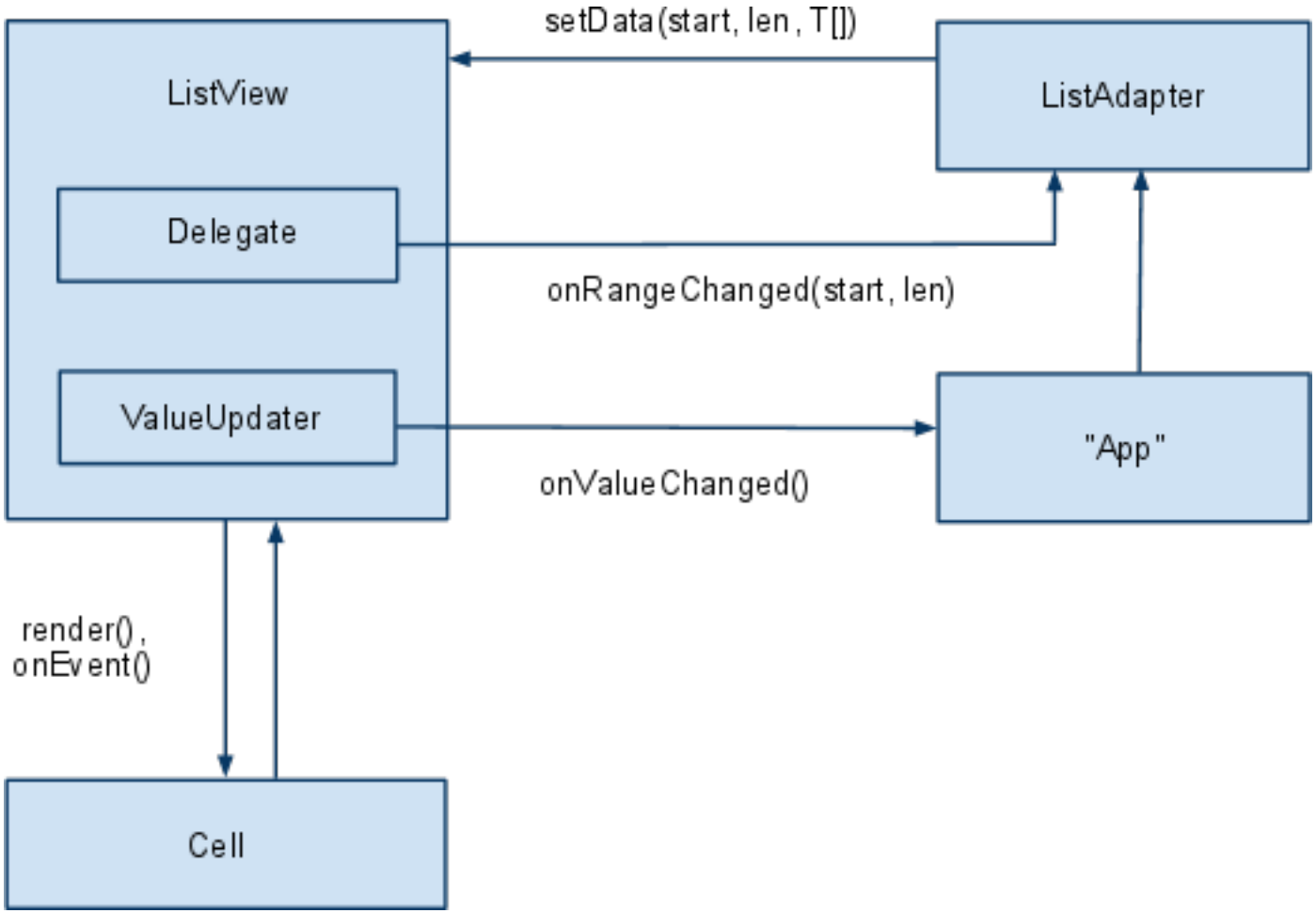
A Simple List Example

```
adapter = new ListViewAdapter<String>();
view = new CellList(new TextCell());
adapter.addView(view);

list = adapter.getList();
list.add("foo");
list.add("bar");
list.add("baz");
```



The Interfaces



Benefits

- Views are free to render cells using innerHTML
  - This is extremely efficient
- Views are easy to connect to async data sources
- Views can handle *push* data
- Working with local data remains simple
  - Inspiration from Glazed Lists

The Cell

- Simple flyweight interface
- Restricted event handling
- Used by collection widgets to render items efficiently

```
// Simplest possible cell
public class HtmlCell extends Cell<String> {

  @Override
  public void render(String value, Object viewData, StringBuilder sb) {
    if (value != null) {
      sb.append(value);
    }
  }
}
```

Wrapping Up

- ClientBundle
  - Optimizes resources and reduce HTTP requests
- UiBinder
  - Reduces boilerplate
  - Easier to use HTML
  - Makes the **easier** thing the **faster** thing
- LayoutPanel
  - No more wrestling with tables for layout
  - Faster layout, smoother animation
- Cell Widgets
  - No more widgets for every single object
  - Makes huge collections easy

View live notes and ask questions about this session on Google Wave

http://bit.ly/io2010-gwt5


Google™
10