GWT Architectures and Lessons Learned

Dipl. Wi.-Ing. Papick Garcia Taboada

pgt technology scouting GmbH
http://pgt.de

Orientation in Objects GmbH
http://oio.de
GWT Development

Basics

Structuring the UI

Talking to the server
Shift happened

Java development,
JS deployment,
Async,
RPC,
RIA/ single page,
...
Web 2.0? development is all about...

...js
...html
...css

none of it

...jcp
...oracle
...IBM
...backend
what is a GWT application?

a chunk of JS that does a lot of DOM manipulation to create web applications
Widget t = new TextBox();

RootPanel.get().add(t);
Architecture shift

Web applications

Model 2 web applications

Rich internet applications

+ testability
+ maintainance
+ product development
Browser

Server

user action

full html response

user action

full html response

user action

full html response

classic web development
Web frameworks
low level, generic tools
let's build big things
TextBox t0 = new TextBox();
TextBox t1 = new TextBox();
TextBox t2 = new TextBox();
TextBox t3 = new TextBox();
TextBox t4 = new TextBox();
VerticalPanel...
SplitPanel...
ScrollPanel
RootPanel.get().add(mainPanel);
maintainance hell
MacGyver

All he needed was a ballpoint pen and a paper clip
Ops
Widget t = new TextBox();

RootPanel.get().add(t);
app-framework wizardry needed
GWT Development
Basics
Structuring the UI
Talking to the server
GWT DEVELOPMENT IS COMPLICATED
HOT NEW STUFF.
NOT REALLY
GWT development is not new, but different

WHY?
WEB DEVELOPMENT IS NOT NEW...
CSS level 2 revision 1, often referred to as "CSS 2.1," fixes errors in CSS 2, removes poorly supported or not fully interoperable features and adds already-implemented browser extensions to the specification. In order to comply with the W3C Process for standardizing technical specifications, CSS 2.1 went back and forth between Working Draft status and Candidate Recommendation status for many years.
RICH CLIENT DEVELOPMENT IS NOT NEW EITHER
nothing new here
really NOTHING new here
Look! YES! This is my code!
It’s all about software engineering
Just a few tips
USE MVP!

You will get used to it
USE MVP!

You will get used to it
USE MVP!

You will get used to it

unit test this
USE MVP!

You will get used to it

MODEL

PRESENTER

MOBILE VIEW

DESKTOP VIEW
event bus please
eventbus.fireEvent(
    NotificationEvent.info(
        "Daten wurden erfolgreich gespeichert"
    )
);
SINGLETON
don’t public static instance
BUT ON IE 6 IT IS SO SLOW!
BUT ON IE 7 IT IS SO SLOW!
BUT ON IE 8 IT IS SO SLOW!
BUT ON IE 9 IT IS SO SLOW!
DID ANYONE TEST ON IE / SURFACE?
DID ANYONE TEST ON IE / SURFACE?
DID ANYONE TEST ON IE / SURFACE?

MOBILE?

CHEAP ANDROID DEVICES?
browsers day to day job
too many HTTP requests
> 2400 DOM elements

Awesome?
too many widgets
use large HTML chunks
<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>18</td>
<td>19</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td>22</td>
<td>24</td>
<td>26</td>
<td>28</td>
<td>30</td>
<td>32</td>
<td>34</td>
<td>36</td>
</tr>
<tr>
<td>7</td>
<td>30</td>
<td>33</td>
<td>36</td>
<td>39</td>
<td>42</td>
<td>45</td>
<td>48</td>
<td>51</td>
<td>54</td>
</tr>
<tr>
<td>6</td>
<td>40</td>
<td>44</td>
<td>48</td>
<td>52</td>
<td>56</td>
<td>60</td>
<td>64</td>
<td>68</td>
<td>72</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

use large HTML chunks
too many widgets ain't good
WHY?
in GWT a Widget is a JS thing holding a DOM thing
CREATE CUSTOM WIDGETS
CREATE CUSTOM EVENTS

don’t extend SimplePanel
don’t extend VerticalPanel
don’t extend FlextTable

extend composite!!!
the new native! do it in CSS…

```css
@-webkit-keyframes redPulse {
    from {
        box-shadow: 0px 0px 2px #ff0033;
    }
    50% {
        box-shadow: 0px 0px 10px #ff0033;
    }
    to {
        box-shadow: 0px 0px 2px #ff0033;
    }
}
```
use LayoutPanel
- GWT Development
- Basics
- Structuring the UI
- Talking to the server
SINGLE PAGE APPLICATION
NAVIGATION?
some action

mainPanel.setWidget(aWidget);

mainPanel.setWidget(bWidget);

„View A“

„View B“

„just do it“ pattern
hard to maintain
history
management

from day one!

back button and refresh as a feature
(not a catastrophe)
Keep It Stupid Simple

- use PLACES framework for main level navigation
- if you really need to, nest activities for a second level. try not to.
- use dialogs for user input, showing data. dialogs are easily reused.
EXAMPLE
top menue bound to places framework
switching between places with fade in and out
teach user to wait until application is ready again
gives us enough time to load the required content

300ms out  
500ms in
<table>
<thead>
<tr>
<th>Termin (Tag)</th>
<th>Termin (Zeit)</th>
<th>Behandlung</th>
<th>Patient</th>
</tr>
</thead>
<tbody>
<tr>
<td>03.11 So.</td>
<td>09:45-10:45m</td>
<td>Behandlung</td>
<td>Lila Blaufuß</td>
</tr>
<tr>
<td>06.11 Mi.</td>
<td>08:00-09:00m</td>
<td>Behandlung</td>
<td>Papick Tabada</td>
</tr>
<tr>
<td>07.11 Do.</td>
<td>11:00-12:00m</td>
<td>Behandlung</td>
<td>Emil Trantütte</td>
</tr>
</tbody>
</table>

Vorschau

<table>
<thead>
<tr>
<th>#</th>
<th>€</th>
<th>€/h</th>
<th>€/m</th>
</tr>
</thead>
<tbody>
<tr>
<td>28.10-03.11</td>
<td>870,00 €</td>
<td>114,72 €</td>
<td></td>
</tr>
<tr>
<td>04.11-10.11</td>
<td>176,00 €</td>
<td>60,34 €</td>
<td></td>
</tr>
<tr>
<td>11.11-17.11</td>
<td>90,00 €</td>
<td>50,00 €</td>
<td></td>
</tr>
<tr>
<td>18.11-24.11</td>
<td>0,00 €</td>
<td>0,00 €</td>
<td></td>
</tr>
</tbody>
</table>

Von 01.05 bis 31.10 noch offen:

<table>
<thead>
<tr>
<th>Prio</th>
<th>Eintrag</th>
<th>Patient</th>
<th>Alter</th>
<th>Notiz</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>14.09.13</td>
<td>Alfred Automatix</td>
<td>100y 6m 17d</td>
<td>kann jederzeit</td>
</tr>
<tr>
<td>50</td>
<td>14.09.13</td>
<td>Frida Feundfertig</td>
<td>43y 9m 5d</td>
<td>40kw</td>
</tr>
<tr>
<td>30</td>
<td>04.10.13</td>
<td>Karl Käfer</td>
<td>nur nichts dienstags</td>
<td></td>
</tr>
<tr>
<td>390d</td>
<td>18.04.13</td>
<td>Trulla Latraviata</td>
<td>40y 8m 7d</td>
<td>Erst bei Vollmond wieder Termin vergeben</td>
</tr>
<tr>
<td>50d</td>
<td>14.09.13</td>
<td>Professor Helmut Hase</td>
<td>66y 10m 4d</td>
<td>ab 15 Uhr</td>
</tr>
</tbody>
</table>
Components inside of „activity“ fire non public, CUSTOM events
datepicker sends WEEK selected event
BOOKMARKABLE

presenter may goto itself, view may be cached
### Hashtag!
STATELESS VIEW

URL contains EVERYTHING needed to rebuild view

user hits reload

GWT apps starts, activity gets fired

view is back again

presenter loads data from server

https://my.lemniscus.de/#TerminplanPlace:47:2013:1383304457011
some actions don‘t require PLACE navigation at all
use POPUPS/ DIALOGS to stay ABOVE navigation
let POPUP/DIALOGS move slowly into view

pin POPUP to one side of the window
Don’t move your user away from his „PLACE“ unless you have to.

Search DIALOG slides in from right side, stays on TOP.
Navigation should not hurt

- The application shown uses only 3 levels of navigation, DOES NOT NEED MORE
- PLACES used for bookmarkable entry points/ back button navigation consistency
- Activities should be STATELESS, to survive page reloads
- Learn from OTHERS, lookout for hashtags…
Once upon a time, a young good designer did a good looking design...

- he will be using photoshop or dreamweaver
- he will not use the software
- he will not build the software
- he will not maintain the software
BEFORE YOU ADD THE LOGO TO THE TOP

how many pixels do your USERS have?

the designer or marketing guy using photoshop is probably sitting in front of a 27“ apple cinema display
GWT Development
Basics
Structuring the UI
Talking to the server
GWT-RPC is a good solution if handled with care.

SomeResult someMethodName(SomeParameter spo)

Interface Versioning is a monstrous thing.

GWT-RPC binds many methods into one interface.
SomeResult someMethodName ( SomeParameter spo )

this will be an object

this will be an object too
the method names bind the requests to the result

SomeResult  someMethodName ( SomeParameter spo )

typesafety all the way
USING GENERICS FOR TYPESAFETY,
GET RID OF METHODS AND INTERFACES
now we just have one interface with one method

\[ \text{<A extends Action<R>, R extends Result> R execute(A action);} \]

typesafety all the way
command pattern

GOF Pattern
commonly used in Rich Clients
someAction -> EXECUTE <- someResult

someActionHandler
GWT client

someAction

GWT-RPC

someActionHandler

someResult

batching
caching
security
caching
exception translation
security

client

server
public class TerminLoadAction
    implements Action<DataResult<TerminData>> {
    private String terminId;

    public TerminLoadAction(String terminId) {
        this.terminId = terminId;
    }

    public String getTerminId() {
        return terminId;
    }
}

public class DataResult<DATA extends Data>
    implements Result {
    private DATA data;

    public DataResult(DATA data) {
        this.data = data;
    }

    public void setData(DATA data) {
        this.data = data;
    }

    public DATA getData() {
        return data;
    }
}

public class DataResult<DATA extends Data> implements Result {
    private DATA data;

    public DataResult(DATA data) {
        this.data = data;
    }

    public void setData(DATA data) {
        this.data = data;
    }

    public DATA getData() {
        return data;
    }
}
void execute(A action, AsyncCallback<R> callback)

dispatch.execute(
    new TerminLoadAction(terminId),
    new AsyncCallback<DataResult<TerminData>>() {
    @Override
    public void onFailure(Throwable caught) {
    }
    @Override
    public void onSuccess(DataResult<TerminData> result) {
    }
    }
);
public interface ActionHandler
   <A extends Action<R>, R extends Result> {

   Class<A> getActionType();

   R execute(
      A action,
      ExecutionContext context)
      throws DispatchException;

}
Server side

custom annotations

spring

@ActionHandlerBean
@Transactional
public final class TerminDataLoadHandler
  implements ActionHandler<TerminLoadAction, DataResult<TerminData>> {

  @Autowired
  private TerminDAO terminDao;

  @Override
  public DataResult<TerminData> execute(TerminLoadAction action,
                                         ExecutionContext context)
                                  throws DispatchException {

    TerminBean termin = …
    TerminData data = …
    return new DataResult<TerminData>(data);
  }

  @Override
  public Class<TerminLoadAction> getActionType() {
    return TerminLoadAction.class;
  }
}
interface versioning hell?
public interface SomeNiceService extends RemoteService {
    String someService(String param);
    String someServiceV2(String param);
    String someServiceV3(String param);
}

public interface SomeNiceServiceV2 extends RemoteService {
    String someService(String param);
}

public interface SomeNiceServiceV3 extends RemoteService {
    String someService(String param);
}

easy way?

right way?

maintainability?
someAction

POJOS

someActionResult

someActionHandler
different versions can coexist!

same result
someAction

someActionV2

someResult

someResultV2

multiple versions

different results

someActionHandler

someActionHandlerV2
why batch?
do it, then this, finally that

done
• one batch call is better than 10 single calls
• less data
• less roundtrip latency
• avoid connection bottleneck
• ensure server side execution order
• less roundtrips
batching can be manual or automatic

server executes actions in given order
automatic batching?
GWT code execution

Scheduler.scheduleEntry(…)

Scheduler.scheduleFinally(…)

Scheduler.scheduleDeferred(…)

browser event loop
GWT code execution

Scheduler.scheduleEntry(…)

Scheduler.scheduleFinally(…)

IDLE
GWT code execution

IDLE

Scheduler.scheduleEntry(…)

collect commands

- cmd 1
- cmd 2
- cmd 3
- cmd ...

fire batch command

Scheduler.scheduleFinally(…)

BATCH EXECUTION

ENABLES FINE GRAINED COMMANDS AND REUSE

+ 

ENABLES ORDERING ACTIONS FOR EXECUTION ON THE SERVER
### Termin

<table>
<thead>
<tr>
<th>Tag</th>
<th>Uhrzeit</th>
<th>Art der Behandlung</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mi.</td>
<td>24.04 10:30 (45m)</td>
<td>Behandlung</td>
</tr>
<tr>
<td>Mi.</td>
<td>24.04 14:00 (60m)</td>
<td>Behandlung</td>
</tr>
<tr>
<td>Do.</td>
<td>25.04 09:00 (45m)</td>
<td>Behandlung</td>
</tr>
</tbody>
</table>

**Buttons:**
- toggleTerminMetadata
- reloadDashboardTermine
- BooleanResult
- DataListResult<Termin>
+ caching
+ reauth when session expired
+ exception handling
Thanks!

- GWT Development
- Basics
- Structuring the UI
- Talking to the server
http://j.mp/gwt-javaland-2014