Google Web Toolkit Architecture Best Practices

+ GWT 2.1 news

papick g. taboada
dipl. wi-ing. (TH)

http://pgt.de
Session Topics

- Developing Rich Clients
  - Navigating through the app
  - Talking to the server
  - A word on generic frameworks
  - What’s in GWT 2.1
architectural freedom
know-how?

yes, we can.
Ray Ryan - lessons learned

use eventbus
...to fight spaghetti

don’t call me, I’ll call you

Master view

Detail view

EventBus
don’t call me, I’ll call you

Master view \rightarrow \text{fire selection event} \rightarrow \text{EventBus} \rightarrow \text{Detail view}

Dienstag, 7. Dezember 2010
don’t call me, I’ll call you

Master view

Detail view

---

don’t call me, I’ll call you, and you too

Bread crumb View

Master view

Detail view
don’t call me, I’ll call you, and you too

Bread crumb View

Master view

Detail view

EventBus

fire selection event

receive selection event

fire selection event

receive selection event

© 2009 Papick G. Taboada | pgt technology scouting GmbH | http://pgt.de
Dienstag, 7. Dezember 2010
don’t call me, I’ll call you

Bread crumb View

Master view

Detail view

receive selection event

fire selection event

receive selection event

Dienstag, 7. Dezember 2010

unreal de-coupling?

Master view

Detail view

EventBus

Dienstag, 7. Dezember 2010
unreal de-coupling?

Master view

Detail view

(1) change event

EventBus

RPC

(2) store data

Master view

Detail view

(1) change event

EventBus

RPC
unreal de-coupling?

Master view

EventBus

(1) change event

(2) store data

(3) updated data

RPC

Detail view

(4) updated data

Master view

EventBus

(1) change event

(4) updated data

(2) store data

(3) updated data

RPC

Detail view

Dienstag, 7. Dezember 2010
Use GWT native events

Divide & Conquer
integration?

GWT modularization
Dependency Injection

- convention + runtime
  - no globals
  - no service locator
  - push in dependencies
  - runtime wires
  - instances together
common DI containers

use reflection.

reflection does **not** work with GWT

deferred binding

GWT offers similar approach through code generators and implementation replacement

“reflection at compile time”
GIN

Model View Presenter
M - V - Controller
M - V - Presenter

M - V - Presenter
Use interfaces

M - V - Presenter
M - V - Presenter

it’s all about de-coupling

GWT brings software engineering to web application development
Session Topics

- Developing Rich Clients
- Navigating through the app
- Talking to the server
- A word on generic frameworks
- What’s in GWT 2.1

Navigation - no problem?
„just do it“ pattern

no navigation engine
eventbus + place abstraction

fire „place A“ event

deliver place event

decoupled place navigation
+ place management

fire place request event
handle place request
fire place event
handle place event

place management
rendering management

PlaceManager

Place "View"

root "View"

fire place request event

handle place request

fire place event

handle place event

fire place render event

handle place render event

rendering management
+ history management

history management

from day one!

back button and refresh as a feature
(not a catastrophe)
complete view management?

Session Topics

- Developing Rich Clients
- Navigating through the app
- **Talking to the server**
- A word on generic frameworks
- What’s in GWT 2.1
command pattern
GOF Pattern
commonly used in Rich Clients

server communication
command pattern

XAction
XResult

dispatch service

XHandler
YHandler
ZHandler
boundaries

Browser / JS       GWT RPC       Servlet / Java

handler registry

by hand, Spring, Guice, you name it
connecting
to the backend

by Hand, Spring, Guice,
you name it

leverage point for
caching
batching
exception handling
securing communication
Session Topics

- Developing Rich Clients
- Navigating through the app
- Talking to the server
- A word on generic frameworks
- What's in GWT 2.1

solving common technical issues
maintainace matters

one size fits prototyping
DefaultFormBuilder builder = new DefaultFormBuilder(layout);

builder.appendSeparator("Flange");

builder.append("Identifier:", new TextBox());
builder.nextLine();

builder.append("PTI/kW:", new TextBox());
builder.append("Power/kW:", new TextBox());
builder.append("s/mm:", new TextBox());
builder.nextLine();

FieldModel<Boolean> likesCheese = fieldOfType(Boolean.class).boundTo(someBean, "cheeseLover");

FieldModel<String> whyILikeCheese = fieldOfType(String.class).boundTo(someBean, "reasonForLikingCheese");

enable(whyILikeCheese).when(likesCheese);

watermark(whyILikeCheese).with("Tell us why you like cheese!");

validateField(whyILikeCheese)
 .using(newNotEmptyValidator("Please share your cheese passion with us.")
 .when(likesCheese);
different jobs to be done

Session
Topics

‣ Developing Rich Clients
‣ Navigating through the app
‣ Talking to the server
‣ A word on generic frameworks
‣ What's in GWT 2.1
actual version is 2.1
new & noteworthy

- Cell Widgets
- MVP Framework (Activities and Places)
- Request Factory
- Editors
- Server-side Speed Traces
- Logging
- Safe HTML
- Integration with SpringSource developer tools

finally! cell widgets!

- data centric widgets
- data-table, data-table, data-table
- concept of reusable cell widgets
- lightweight, fast, efficient
### cell list

<table>
<thead>
<tr>
<th>First Name</th>
<th>Last Name</th>
<th>Category</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Erica</td>
<td>Knight</td>
<td>Businesses</td>
<td>26 Baker St</td>
</tr>
<tr>
<td>Gregory</td>
<td>McDonald</td>
<td>Contacts</td>
<td>172 Fourteenth Way</td>
</tr>
<tr>
<td>Darrell</td>
<td>Woods</td>
<td>Contacts</td>
<td>170 Williams St</td>
</tr>
<tr>
<td>Pamela</td>
<td>Ross</td>
<td>Coworkers</td>
<td>303 Fourteenth Blvd</td>
</tr>
<tr>
<td>Joann</td>
<td>Ward</td>
<td>Family</td>
<td>24 Fifth Blvd</td>
</tr>
<tr>
<td>Gloria</td>
<td>Bailey</td>
<td>Coworkers</td>
<td>331 Peachtree Blvd</td>
</tr>
<tr>
<td>Marlo</td>
<td>Harrison</td>
<td>Family</td>
<td>393 Fourth St</td>
</tr>
<tr>
<td>Michelle</td>
<td>Ruiz</td>
<td>Contacts</td>
<td>574 Bell St</td>
</tr>
<tr>
<td>Paula</td>
<td>Simpson</td>
<td>Coworkers</td>
<td>656 Fowler Blvd</td>
</tr>
<tr>
<td>Howard</td>
<td>Rice</td>
<td>Businesses</td>
<td>385 Forsyth Rd</td>
</tr>
<tr>
<td>Tim</td>
<td>Thompson</td>
<td>Coworkers</td>
<td>782 Juniper Ln</td>
</tr>
<tr>
<td>Jay</td>
<td>Cunningham</td>
<td>Businesses</td>
<td>106 Sixth Ave</td>
</tr>
<tr>
<td>Bill</td>
<td>Barnes</td>
<td>Contacts</td>
<td>151 Central Cir</td>
</tr>
<tr>
<td>Melissa</td>
<td>Roberts</td>
<td>Businesses</td>
<td>243 Piedmont Blvd</td>
</tr>
<tr>
<td>Yolanda</td>
<td>Owens</td>
<td>Businesses</td>
<td>486 Central Pkwy</td>
</tr>
</tbody>
</table>

### cell table
cell tree

cell browser
available cell types

• Text
  • **TextCell** - A non-editable cell that displays text
  • **ClickableTextCell** - A text field; clicking on the cell causes its ValueUpdater to be called
  • **EditTextCell** - A cell that initially displays text; when clicked, the text becomes editable
  • **TextInputCell** - A field for entering text

available cell types

• Buttons, Checkboxes and Menus
  • **ActionCell**<C> - A button that takes a delegate to perform actions on mouseUp
  • **ButtonCell** - A button whose text is the data value
  • **CheckboxCell** - A checkbox that can be checked or unchecked
  • **SelectionCell** - A drop-down menu for selecting one of many choices
available cell types

- Dates
  - **DateCell** - A date that conforms to a specified date format
  - **DatePickerCell** - A date picker that displays a month calendar in which the user can select a date
- Numbers
  - **NumberCell** - A number that conforms to a specified number format

- Images
  - **ImageCell** - A cell used to render an image URL
  - **ImageResourceCell** - A cell used to render an ImageResource
  - **ImageLoadingCell** - A cell used to render an image URL. A loading indicator is initially displayed
available cell types

- Compositions
  - **CompositeCell</C>** - A composition of multiple Cells.
- Decorators
  - **IconCellDecorator</C>** - A decorator that adds an icon to another Cell

MVP Framework

- Views
  - IsWidget
- ClientFactory
  - EventBus
  - PlaceController
- Activities
- Places
- PlaceHistoryMapper
- ActivityMapper
request factory

- data-centric transfer protocol
- only deltas are sent back to server
- proxies for easy DTO implementation

```java
@Entity
public class Employee {
    @Size(min = 3, max = 30)
    private String userName;

    private String department;

    @NotNull
    private String displayName;

    private String password;

    @JoinColumn
    private Long supervisorKey;
}
```
public interface EmployeeProxy extends EntityProxy {
    String getDepartment();
    String getDisplayName();
    Long getId();
    String getPassword();
    EmployeeProxy getSupervisor();
    String getUserName();
    void setDepartment(String department);
    void setDisplayName(String displayName);
    void setPassword(String password);
    void setSupervisor(EmployeeProxy supervisor);
    void setUserName(String userName);
}

public interface ExpensesRequestFactory extends RequestFactory {
    EmployeeRequest employeeRequest();
    ExpenseRequest expenseRequest();
    ReportRequest reportRequest();
}

RequestFactory
@Service(Employee.class)
public interface EmployeeRequest extends RequestContext {

    Request<Long> countEmployees();
    Request<Long> countEmployeesByDepartment(String department);
    Request<List<EmployeeProxy>> findAllEmployees();
    Request<EmployeeProxy> findEmployee(Long id);
    Request<List<EmployeeProxy>> findEmployeeEntries(int firstResult, int maxResults);
    Request<List<EmployeeProxy>> findEmployeeEntriesByDepartment(String department, int firstResult, int maxResults);
    InstanceRequest<EmployeeProxy, Void> persist();
    InstanceRequest<EmployeeProxy, Void> remove();
}

// (....)
requestFactory = GWT.create(ExpensesRequestFactory.class);

// (....)
requestFactory.employeeRequest().findEmployee(employeeId).fire(
    new Receiver() {
        @Override
        public void onSuccess(EmployeeProxy employee) {
            ...
        }
    });
server implementation

@Entity
public class Employee {

    public static final EntityManager entityManager() {
        return EMF.get().createEntityManager();
    }

    @SuppressWarnings("unchecked")
    public static List<Employee> findAllEmployees() {
        EntityManager em = entityManager();
        try {
            List<Employee> list = em.createQuery("select o from Employee o").getResultList();
            // force to get all the employees
            list.size();
            return list;
        } finally {
            em.close();
        }
    }

    // (...)

Entity + DAO + Service == 1
standard java ee thinking

request factory thinking
best of both worlds?

RequestFactory_2_1_1

Issue 5111:
Add a service layer to RequestFactory
Thanks!

Schulung: Einführung in GWT
http://www.oio.de/google-web-toolkit-schulung.htm

Artikel: Google Web Toolkit: Webanwendungen mit GWT entwickeln

Schulung: Design Patterns mit Java

mehr von OIO zum Thema