



DWR

Hands-on DWR

Joe Walker

DWR Lead Developer
Getahead
<http://getahead.org/dwr>

Geert Bevin

RIFE/DWR Developer
Uwyn bvba
<http://uwyn.com/>

TS-6410

Goal of the Talk

The plan for the next 60 minutes

Learn how to use DWR from
some cool worked examples

Agenda

Overview of DWR

Installation first steps

The first application

Explaining the feature set

Writing the game

Integration with other Ajax libraries

Summary

Agenda

Overview of DWR

Installation first steps

The first application

Explaining the feature set

Writing the game


Integration with other Ajax libraries

Summary

Web Browser

HTML / Javascript

```
function eventHandler() {  
    AjaxService.getOptions(populateList);  
}  
  
function populateList(data) {  
    dwr.util.addOptions("listid", data);  
}
```



The diagram shows a yellow document icon representing the web browser. It contains two JavaScript functions. The first function, `eventHandler()`, calls `AjaxService.getOptions(populateList)`. The second function, `populateList(data)`, calls `dwr.util.addOptions("listid", data)`. A grey arrow points from the `eventHandler` function to the `populateList` function. Below the `populateList` function, there is a small UI element: a dropdown menu with the value '1' selected, and a list of options '1', '2', '3' displayed below it. A grey arrow points from the `populateList` function to this UI element.

Web Server

Java

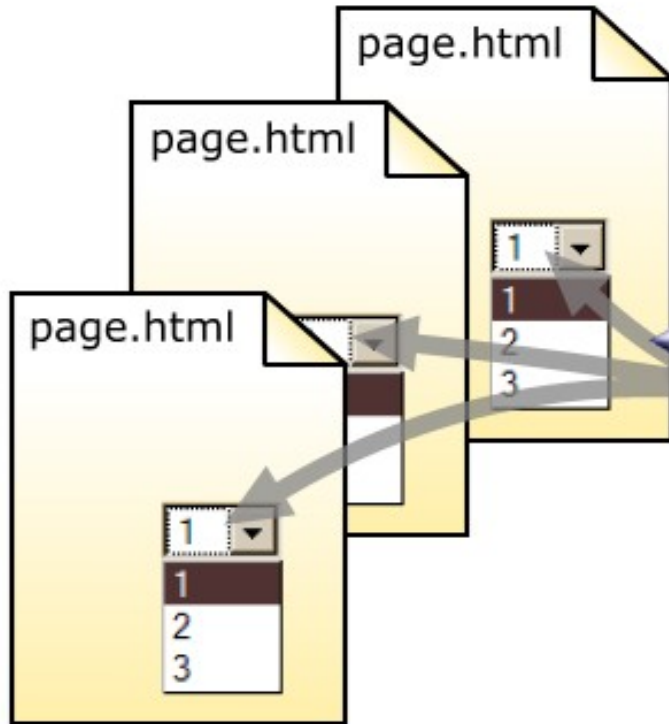
```
public class AjaxService {  
  
    public String[] getOptions() {  
        return new String[] { "1", "2", "3" };  
    }  
  
}
```

The diagram shows a blue document icon representing the web server. It contains a Java class named `AjaxService` with a `getOptions()` method that returns an array of strings: `new String[] { "1", "2", "3" }`.

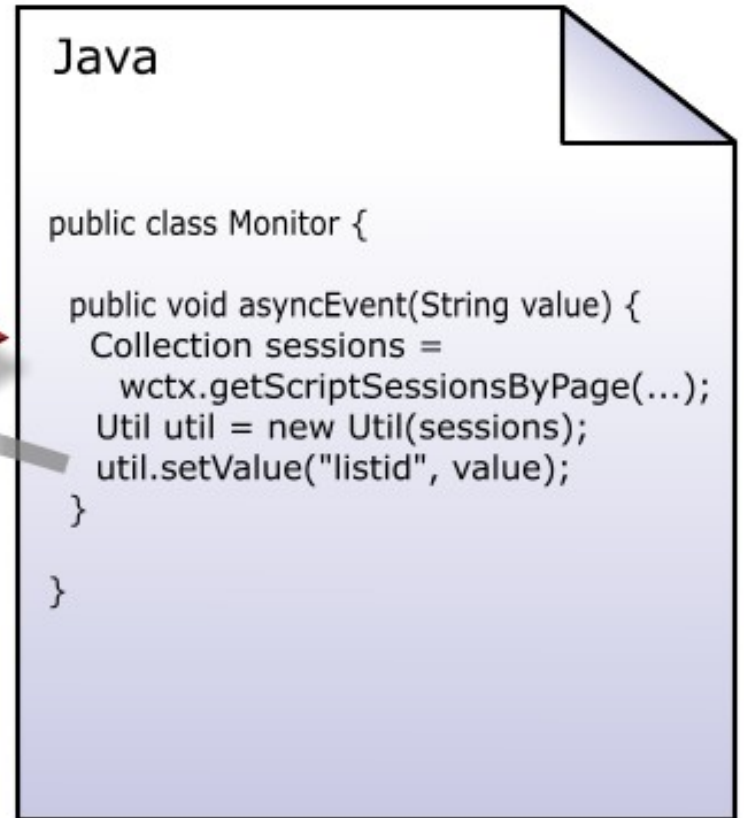


DWR

Web Browsers



Web Server



DWR

Agenda

Overview of DWR

Installation first steps

The first application

Explaining the feature set

Writing the game

Integration with other Ajax libraries

Summary

Getting Started

Installation in 4 steps

- Copy dwr.jar into WEB-INF/lib
 - <http://directwebremoting.org/>
- Add the DWR servlet to WEB-INF/web.xml
- Create dwr.xml to give DWR permission to call your code
- Browse the automatically generated test pages
 - <http://localhost:8080/WEBAPP/dwr/>

Agenda

Overview of DWR

Installation first steps

The first application

Explaining the feature set

Writing the game

Integration with other Ajax libraries

Summary



DEMO

Your first DWR page



Agenda

Overview of DWR

Installation first steps

The first application

Explaining the feature set

Writing the game

Integration with other Ajax libraries

Summary

DWR Can Marshall:

(in short, basically anything ...)

- Primitive types, and their Object counterparts
 - int, boolean, long, float, double, etc
- Obvious classes
 - String, Date, BigDecimal, BigInteger, Enums, etc
- Arrays and Collections
 - Map, List, Set, Iterator, ...
- JavaBeans™ and Objects
- XML objects
 - DOM, XOM, JDom, Dom4J

JavaBeans = A component based on Java Beans™ component architecture

Security

- DWR does not call anything without your permission
- Use method level access control over exported classes and objects
- Use standard role-based security to declare roles that can access methods

Accessibility

- `dwr.util` supports pluggable notifiers
 - `.focus()` for screen readers
 - Yellow fade for partially sighted
- New notifiers can be easily added
 - e.g. To make a sound on a change

Agenda

Overview of DWR

Installation first steps

The first application

Explaining the feature set

Writing the game

Integration with other Ajax libraries

Summary

The Game

- We're starting from scratch (nearly)
- It's going to be really dull if we really start from scratch
- So here's a few things we made earlier...

What We Start with: `boat.gif`



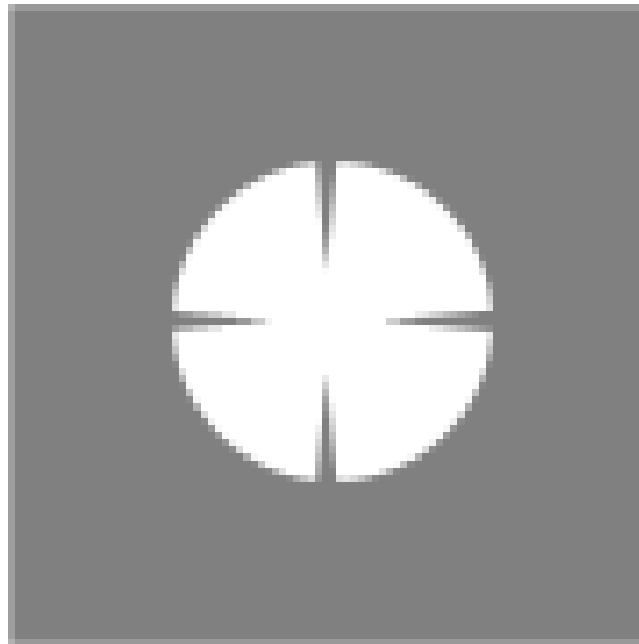
What We Start with: `title.png`



What We Start with: seamap.png



What We Start with: `crosshair.png`



What We Start with: `index.html`

```

<body onload="init();"

<table id="map"></table>
<div id="sidebar">
  <p>
    <b>Name</b>: <input type="text" size="23"/><br/>
    <b>Position</b>:
    Row <input type="text" size="1" onchange="move()"/>,
    Column <input type="text" size="1" onchange="move()"/>
  </p>
  <table>
    <tr>
      <th>Player</th>
      <th>Score</th>
    </tr>
  </table>
  <p>
    <input type="text" size="25"/>
    <input type="button" value="Send"/>
  </p>
  <ul id="chatlog"></ul>
</div>
</body>

```

What We Start with: battleships.css

```

body { margin:0px 40px 0px 40px; color:#fff; background-color:#000;
  font-family:'Century Gothic', Helvetica, Arial, clean, sans-serif;
}

select, textarea, input[type='text'], input[type='password'] {
  font-family:'Century Gothic', Helvetica, Arial, clean, sans-serif;
  font-size:1em; padding:0px 3px; margin:2px; border:1px solid #999;
  background-color:#444; color:#fff;
}

button, input[type='button'], input[type='submit'], input[type='cancel'] {
  font-family:'Century Gothic', Helvetica, Arial, clean, sans-serif;
  font-size:1em; color:#fff; background-color:#000;
}

table { border-collapse:collapse; }

#map { background-image:url("seamap.png"); background-position:right;
  border:10px dashed #484; margin-top:10px; padding-bottom:10px;
}
#map td { width:60px; height:70px; border:1px solid #999;
  background-repeat:no-repeat; background-position:center;
}
#map td:hover { background-image:url("crosshair.png"); }
#map .home { background-image:url("boat.gif"); }

#chatlog { list-style-type:none; padding:0; margin:0; }
#chatlog .chatname { font-size:70%; }
#chatlog .chattext { font-size:80%; }

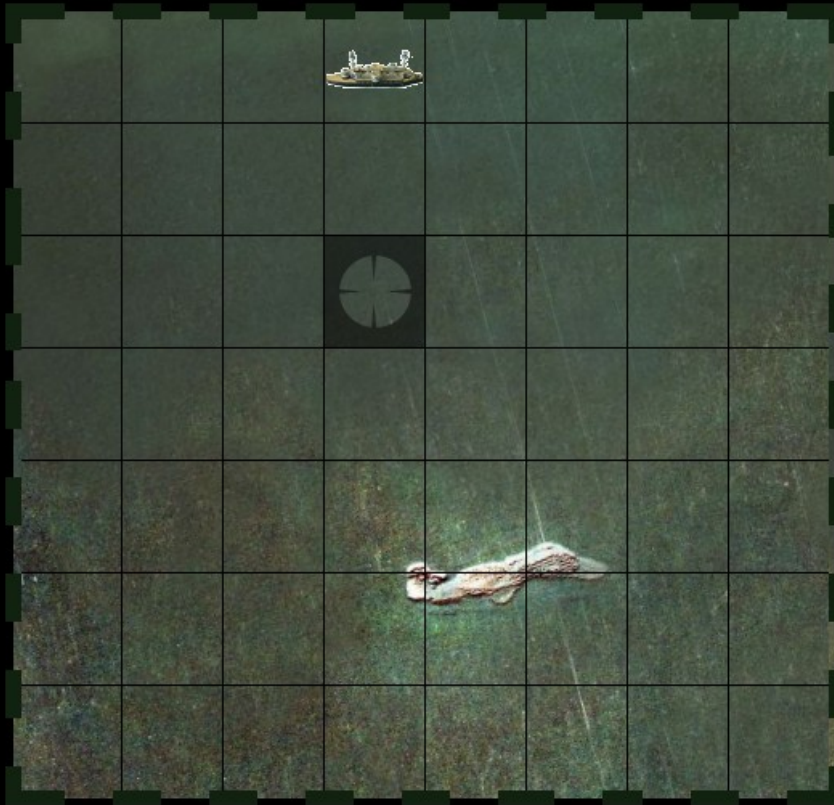
#sidebar { position:absolute; top:0; right:0; width:350px; padding:10px; border-left:1px solid #999; }

#shotouter { background-color:#112; float:right; border:1px solid #334;
  width:200px; height:24px; margin-right:80px;
}

#shotinner { width:0px; height:16px; margin:3px 6px; background-color:red;
  color:#fff; white-space:nowrap; font-size:70%;
}

```

Multi Player Battleships



Name:

Position: Row , Column

Shot:

Player	Score
Terminator Mark	1
Joe the Pitiless	0
Bloodsucker Ming	-1
Barbarian Bruce	1
Barbarian Bob	2
Leon the Hunn	-4

Barbarian Bob: heh!

System: Barbarian Bob fragged Bloodsucker Ming

System: Barbarian Bob fragged Joe the Pitiless

Bloodsucker Ming: hello

Joe the Pitiless: Sean the Pitiless has become Joe the Pitiless

Joe the Pitiless: hello everyone

System: Sean the Pitiless fragged Leon the Hunn

System: Barbarian Bruce fragged Leon the Hunn

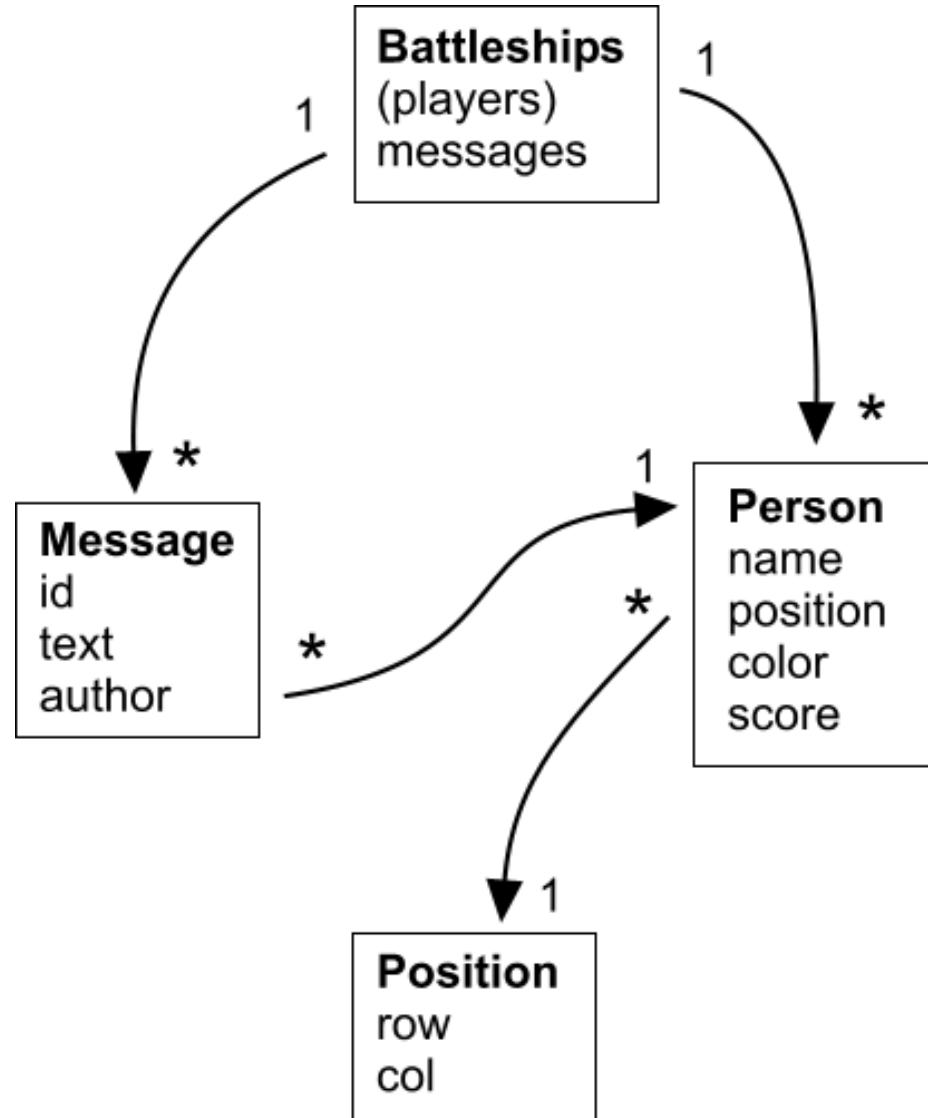
Barbarian Bruce: woot!

System: Barbarian Bruce fragged himself!

System: Barbarian Bruce fragged Leon the Hunn

System: Terminator Mark fragged Leon the Hunn

The Design

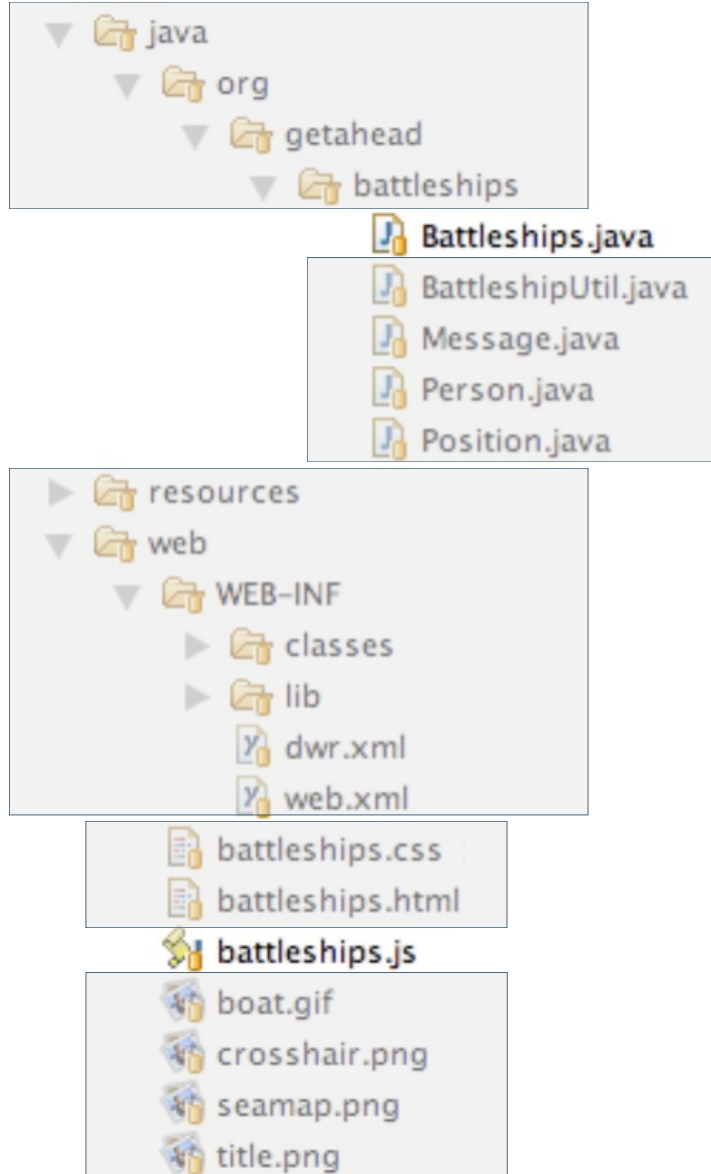


The Files

Pojos and Utilities

Directories, Java
Archive (JAR) Files
Configuration Files

HTML, CSS, Graphics



The Files that
Make the Game
Interactive

The Messages

- Browser-to-Server
 - `init()`
 - `sendMessage()`
 - `shoot()`
- Server-to-Browser
 - `serverUpdate()`



DEMO

The Game



Agenda

Overview of DWR

Installation first steps

The first application

Explaining the feature set

Writing the game

Integration with other Ajax libraries

Summary

Where Does DWR Fit?

In the tradition of good
Unix[®] software tools,
DWR does one job and does it well.

As a result; DWR fits in a wide range
of web applications

Where Does DWR Fit?

- For some Ajax magic with Struts 1.0
- With newer Spring, Webwork, JavaServer™ Faces technology or RIFE code
- By itself with `dwr.util`
- As the service layer to a Scriptaculous or Dojo application
- In the enterprise with TIBCO GI and a service-oriented architecture



DEMO

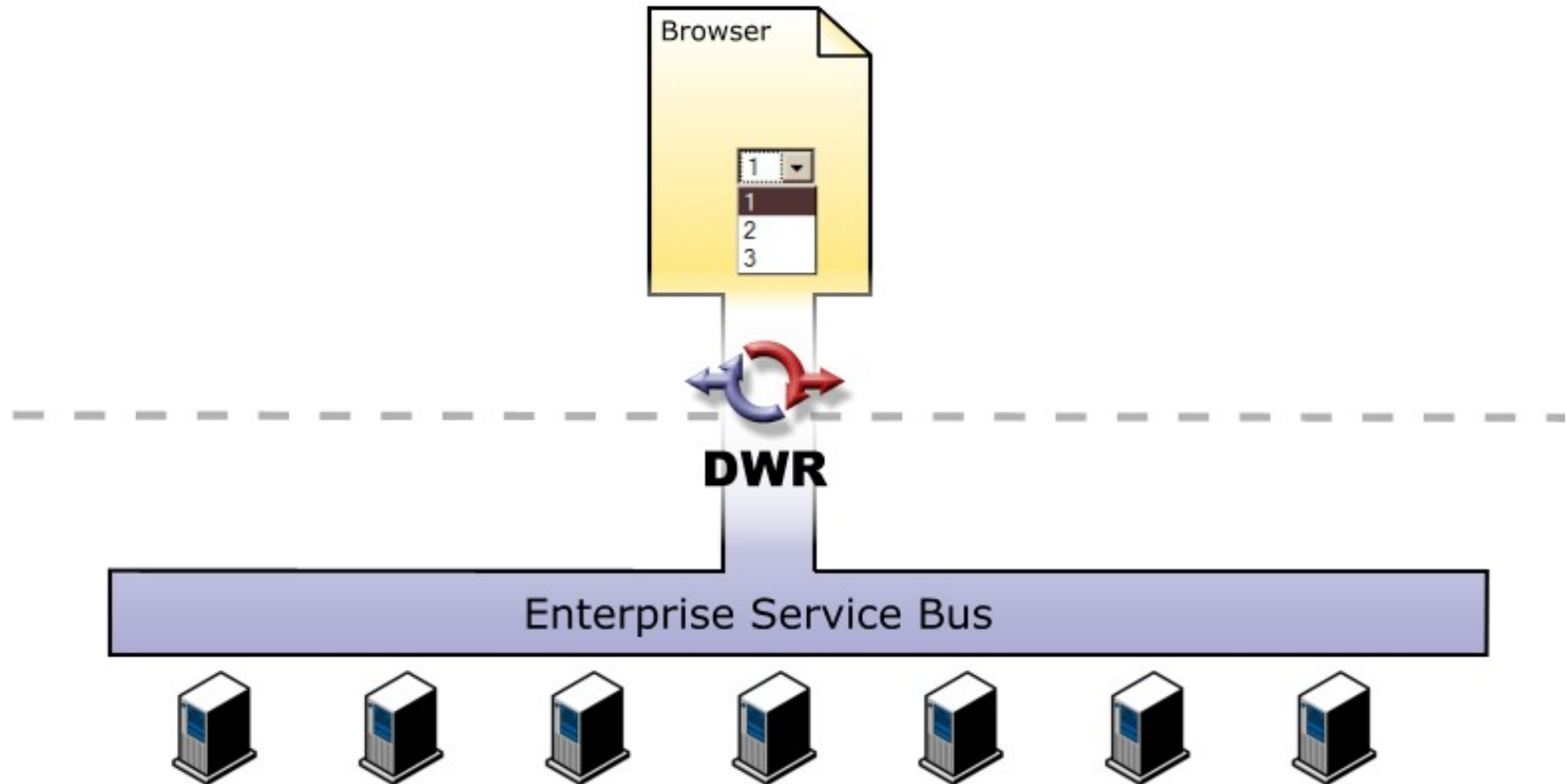
Integrating with other Ajax libraries



DWR + OpenAjax + TIBCO GI

- Publish/Subscribe architecture
- No dependency between TIBCO GI and DWR
- Benefits of decoupling:
 - Easy to provide multiple implementations
 - Easy testing
 - Better forward compatibility

Extending the Enterprise



Summary

- It's dead easy to get going with DWR
- You can use it to create advanced applications without large amounts of code
- Reverse Ajax allows your applications to have high levels of user interactivity
- It fits in your current web application without requiring you to start from scratch

For More Information

- <http://directwebremoting.org/>
- <http://getahead.org/blog/joe/>
- <http://www.tibco.com/devnet/gi/>



Q&A

Hands-on DWR
Joe Walker/Geert Bevin





DWR

Hands-on DWR

Joe Walker

DWR Lead Developer
Getahead
<http://getahead.org/dwr>

Geert Bevin

RIFE / DWR Developer
Uwyn bvba
<http://uwyn.com/>

TS-6410