GWT

Say goodbye to the pains of Ajax

Yibo

The dark age of Ajax

- DOM
- JavaScript
- XML
- CSS
- Standard
- Browsers: browser-specific dependencies.
- Differences
- Complexity
- Exprerience: Minesweeper

What is GWT?

- Google Web Toolkit
- make Ajax development easier by hiding browser incompatibilities from the programmer and allowing the developer to work in a familiar Java development environment
  browser-specific dependencies.
- unify client and server code into a single application written in one language: Java.
- abstract the browser's DOM, hiding differences between browsers behind easy to extend object-oriented UI patterns. This helps make code portable over all supported browsers.

How it works?

- Write AJAX apps in the Java language, then compile to optimized JavaScript
- Step through live AJAX code with your Java debugger
- Compile and deploy optimized, cross-browser JavaScript

Install

- Before you start coding you need to install
  - Java, JDK
  - IDE: Eclipse, NetBeans
  - GWT
- GWT installation
  - Easy
  - No special install is needed
  - Unzip GWT to your machine

Set up

- Add GWT directory to your system environment
  - Windows: Environment Variable Path
  - *nix: PATH variable

Create a Project

- Create a directory
- Create a project
  - projectCreator -eclipse MyProject

  Output:
  - Created directory c:\gwt-projects\src
  - Created directory c:\gwt-projects\test
  - Created file c:\gwt-projects\.project
  - Created file c:\gwt-projects\.classpath

Create a Project cont.

- Creat an application
  - applicationCreator -eclipse MyProject org.freefood.pizza.client.NineInch

- Created directory c:\gwt-projects\myproject\src\org\freefood\pizza
- Created directory c:\gwt-projects\myproject\src\org\freefood\pizza\client
- Created directory c:\gwt-projects\myproject\src\org\freefood\pizza\public
- Created file c:\gwt-projects\myproject\src\org\freefood\pizza\NineInch.gwt.xml
- Created file c:\gwt-projects\myproject\src\org\freefood\pizza\public\NineInch.ht
- ml
- Created file c:\gwt-projects\myproject\src\org\freefood\pizza\public\NineInch.cs
- s
- Created file c:\gwt-projects\myproject\src\org\freefood\pizza\client\NineInch.ja
- va
- Created file c:\gwt-projects\myproject\NineInch.launch
- Created file c:\gwt-projects\myproject\NineInch-shell.cmd
- Created file c:\gwt-projects\myproject\NineInch-compile.cmd

Hosted vs. Web Mode

- Hosted Mode (development)
- interacting with your GWT application without it having been translated into JavaScript
- Development Shell
  - Tomcat Server
  - Shell console
- Hosted Browser
- Two connections between Shell and Browser
  - 1st: http connection (html, resources)
  - 2nd: Backdoor connection (route the interaction to Java code in the shell)

Hosted vs. Web Mode cont.

- Web Mode (in production)
- Release mode
- compile Java code into a form that can be run inside a regular browser
- Easy: Compile/Browser button
- Translate .client package into JS
- Open a browser, and point to the url
- Shell serves as a web server

Hosted vs. Web Mode cont.

- Flow of Web Mode
- 1. The web browser loads MyApp.html.
- 2. MyApp.html loads org.xxx.client.MyApp.js with a <script> tag to get the module name.
- 3. JavaScript inside looks at the browser's userAgent field to determine what kind of browser the user is running (IE5,6,7, FireFox, Opera, etc.). Then it selects the correct code (cache file) for that browser type and redirects the <iframe> there.
- 4. The JavaScript equivalent of your onModuleLoad( ) method is executed, and the rest of your application goes from there. Manipulations to the browser DOM are performed with ordinary dynamic HTML calls in the compiled JavaScript.

Hosted vs. Web Mode cont.

- Web Mode (deployment)
- Pretty easy: copy “www” directory to another location

User Interface

- buttons, lists, and tables, you add them to parents, and you interact with them via listeners.
- GWT app will appear in a web browser, so there has to be an HTML page involved somewhere.

Module

- A GWT module is a collection of client-side application code and resources you supply.

```
<module>
<!-- Inherit the core Web Toolkit stuff. -->
<inherits name='com.google.gwt.user.User' />
<!-- Specify the app entry point class. -->
<entry-point class='com.xyz.client.MyApp' />
</module>
```

- when the HTML page is loaded, GWT looks up the xml file to get the class name, and starts calling code in the com.xyz.client.MyAppclass.

Entry-Point Class

- Implements EntryPoint

  public class NineInch implements EntryPoint {

  …

  }

RPC

- GWT lets part of your application run in both server and client side, with seamless communication between them using Remote Procedure Calls
- Rich Internet Applications
- a remote procedure call is simply a way the client can execute some logic on the server and get a result back.
- GWT RPC

RPC cont.

- Define a template that describes the call
- GWT uses a standard naming convention to connect client and server

Name	Location	Purpose
interface Service	client and server	Describes the service
class ServiceImpl	server	Servlet
interface ServiceAsync	client	Lets you call the service

RPC cont.

- Server side

```
public class ServiceImpl extends RemoteServiceServlet
implements Service {
…
}
```

- Client side

```
ServiceAsync service = (ServiceAsync)
    GWT.create(Service.class);
AsyncCallback callback = new AsyncCallback(){…};
service.invokeService(params, callback);
```

RPC cont.

- RpcExample.gwt.xml
- <!-- Specify the servlet class. -->
- <servlet path="/services"
  class="com.xyz.server. ServiceImpl" />

Other Features

- JSNI
- JSON
- Internationalization
- History and Bookmarks