The Google Web Toolkit (GWT): Overview & Getting Started
(GWT 2.4 Version)

Originals of Slides and Source Code for Examples:
http://courses.coreservlets.com/Course-Materials/gwt.html

Customized Java EE Training: http://courses.coreservlets.com/
GWT, Java, JSF 2, PrimeFaces, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android.
Developed and taught by well-known author and developer. At public venues or onsite at your location.

For live Ajax and GWT training, please see courses at http://courses.coreservlets.com/.

Taught by the author of Core Servlets and JSP, More Servlets and JSP, and this tutorial. Available at public venues, or customized versions can be held on-site at your organization.

- Courses developed and taught by Marty Hall
  - JSF 2, PrimeFaces, servlets/JSP, Ajax, jQuery, Android development, Java 6 or 7 programming, custom mix of topics
  - Ajax courses can concentrate on 1 library (jQuery, Prototype/Scriptaculous, Ext-JS, Dojo, etc.) or survey several
- Courses developed and taught by coreservlets.com experts (edited by Marty)
  - Spring, Hibernate/JPA, EJB3, GWT, Hadoop, SOAP-based and RESTful Web Services
Contact hall@coreservlets.com for details
GWT
(GWT development team also pronounces it as ‘gwit’)

Customized Java EE Training: http://courses.coreservlets.com/
GWT, Java, JSF 2, PrimeFaces, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android.
Developed and taught by well-known author and developer. At public venues or onsite at your location.

Topics in This Section

• What GWT is all about
  – Motivation, pros, cons
• Installing GWT
  – Core GWT
  – Eclipse plugin
  – Browser plugin
• Making a project
  – Using Eclipse plugin
• Testing process
  – Development mode
  – Production mode
  – Deployed to standard Java server
Motivation

Why Web Apps?

• Downsides to browser-based apps
  – GUI is poor
    • HTML is OK for static documents, but lousy for programs
  – Communication is inefficient
    • HTTP is poor protocol for the way we now use Web apps

• So why does everyone want Web apps?
  – Universal access
    • Everyone already has a browser installed
    • Any computer on the network can access content
  – Automatic “updates”
    • Content comes from server, so is never out of date
Why Ajax?

- **Solve three key problems of Web apps**
  - Coarse-grained updates
  - Synchronous: you are frozen while waiting for result
  - Extremely limited options for widgets (GUI elements)
- **Still browser based**
  - Ajax is about “what is the best you can do with what everyone already has in their browser?”
- **“Real” browser-based active content**
  - Newer and less proven
    - Microsoft Silverlight (struggling)
    - JavaFX (gasping for air)
  - Failed: Java Applets
    - Not universally supported; can’t interact with the HTML
Why Ajax? (Continued)

- “Real” browser-based active content
  - Serious alternative: Flash/Flex
- Not preinstalled on all PCs; not available for iPhone/iPad

Traditional Web Apps vs. Ajax Apps

- Traditional Web Apps: Infrequent Large Updates
  - Web Page 1.
    - Blah, blah, blah, blah, Yadda, yadda, yadda.
    - Blah, blah, blah, blah.
    - Yadda, yadda, yadda.
  - Web Page 2.
    - Blah, blah, blah, blah, Yadda, yadda, yadda.
    - Blah, blah, blah, blah.
    - Yadda, yadda, yadda.

- Ajax Apps: Frequent Small Updates
  - Web Page.
    - Blah, blah, blah, blah, Yadda, yadda, yadda.
    - Blah, blah, blah, blah.
    - Yadda, yadda, yadda.
  - Server
Google Home Page (formerly Google Suggest)

Ajax Jobs
Indeed.com compiles data from multiple jobs sites

Job Trends from Indeed.com
- java
- c#
- vb
- php
- ajax
Ajax Problems that GWT Tries to Address

• At least two languages are required
  – JavaScript on the client
  – Java or another language on the server

• Invoking server is cumbersome
  – Even with JSON-RPC, DWR, or jQuery support
  – Want more powerful RPC mechanism

• JavaScript scales poorly to large projects
  – No strong typing
  – Weak object hierarchies
  – Many run-time errors instead of compile-time errors
  – Incompatibility across platforms
  – Can’t use preexisting Java libraries or utilities

GWT Jobs (Absolute and Relative)
Overview and Pros/Cons

Overview of the Google Web Toolkit (GWT)

• Big Idea
  – You write *both* client-side and server-side code in Java
  – Client-side code
    • Uses an API similar to Swing
    • Most basic JDK 1.6 constructs and classes supported
    • Gets compiled into JavaScript that runs in your browser
  – Server-side code
    • Client uses a callback API and specifies data source URL
      – Once you define callback, you are mostly using regular Java method calls with complex arguments and return values
    • Server extends special class and defines *regular* methods
      – These are not servlet-style *doBlah* methods linked to HTTP
      – Arguments and return values can be
        » Primitives, strings, arrays, a few java.util collections, Serializable custom classes
Sites that Use GWT

- **Google Sites**
  - Apache Wave, Google Base, Google Checkout, Google Moderator, Google Web, Google AdSense
- **Outside Sites**
  - See more at http://gwtgallery.appspot.com/

Advantages of GWT

- **No JavaScript syntax errors**
  - Use a reliable, strongly-typed language (Java) for development and debugging
  - No JavaScript programming at all!
- **Can use complex Java on the client**
  - Turned into JavaScript, but you still use String, array, Math class, ArrayList, HashMap, custom classes, etc.
  - Full IDE-based Java support for development and debugging
  - Your client-side code can share data structures with your server-side code
    - Not JSONified equivalents, but the same classes
    - See details on next slide
Drawbacks of JSONification

- **Sends state, not behavior, of object**
  - So, properties that are derived from other properties will be inconsistent on client if other properties are changed

- **Example: Java “Circle” class**
  - getRadius
    - Returns instance variable
  - getArea
    - Returns Math.PI*radius*radius

- **JSONified instance**
  - Example
    - Make new Circle(10). Pass to JSONObject, call toString
    - Result: { "radius": 10.0, "area": 314.15927…}
  - Problem
    - If client-side code changes radius, area doesn’t match

- **GWT sends state & behavior**

Advantages of GWT (Continued)

- **Can send complex Java types to/from the server**
  - Real Java objects gets serialized across network.
  - More powerful and convenient approach than JSON-RPC or DWR.

- **Standalone test environment**
  - Can test within Eclipse without installing a new server

- **Support by a major company**
  - Supported by the company that popularized Ajax in the first place
  - Support won’t go away like perhaps with AjaxTags

- **Active user mailing list (~80 messages daily)**
Advantages of GWT (Continued)

- **Faster Apps**
  - GWT uses deferred binding
    - Sends exactly the right code for the user’s circumstances
      - User agent
      - Locale
      - Debug vs. production
      - Network characteristics
  - Quote from Joel Webber, GWT co-creator:
    “The fastest code is that which does not run.”
  - Java is not compiled “blindly” into JavaScript
    - Many advanced compiler design techniques are used to
      make everything faster
      - E.g., automatic in-lining methods, etc.
  - Watch: youtube.com/watch?v=nvti32k4xyU

Disadvantages of GWT

- **Less incremental; big learning curve**
  - JSP developers can learn AjaxTags in an hour
  - JSF developers can learn JSF 2.0 Ajax support in an hour
    - It takes much longer to get anything running with GWT.
  - JavaScript developers can adapt to new library (e.g.,
    move from jQuery to Dojo) relatively easily
    - GWT seems foreign to them
- **Nonstandard approach to HTML and JavaScript**
  - Programmatic creation of HTML can make use of
    standard HTML and CSS tools more difficult.
  - You never explicitly put JavaScript in your HTML.
    - Very powerful in the long run; hard to get used to at first.
Disadvantages of GWT (Continued)

- **Not search engine friendly**
  - Google and others won’t index the dynamic content
- **Only for Java developers**
  - Most Ajax environments do JavaScript on the client and have a choice for the server. GWT is based entirely around Java.
    - GWT does support “normal” Ajax requests, but if you didn’t plan to mostly use Java on server, would you choose GWT in the first place?
  - Can’t make use of the good features of JavaScript
    - Fast prototyping, functional programming, easy DOM querying and manipulation (e.g., with jQuery)
- **Unusual approach**
  - Fundamentally different strategy than all other Ajax tools makes evaluation and management buyoff harder

Alternatives: Rough Cuts

- **Incremental updates to existing project**
  - Where you want to add Ajax functionality to Web app
    - Use traditional JavaScript library such as jQuery, Prototype/Scriptaculous, Dojo, Ext-JS, YUI, or Google Closure
- **New project**
  - Project is hybrid of regular Web app (with page navigation) and Ajax (with dynamic page updates)
    - Java developers: use JSF 2.0, Struts 2.0, or Spring MVC 3
    - .NET developers: use ASP.NET Ajax
- **New project**
  - Project has complex client-server comms, and/or
  - Project wants to look like a desktop app
    - Use GWT
Installation and Setup

Customized Java EE Training: http://courses.coreservlets.com/
GWT, Java, JSF 2, PrimeFaces, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android.
Developed and taught by well-known author and developer. At public venues or onsite at your location.

Installation: Overview

- **Install Eclipse Plugin**
  - http://code.google.com/eclipse/docs/getting_started.html
  - Choose to install the GWT SDK at the same time
  - *See details on next slide*
- **Install Browser Plugin**
  - The first time you run in a browser in development mode, it will prompt you to install browser plugin.
  - *See details in section on running in Production Mode*
- **(Optional) Install GWT SDK**
  - http://code.google.com/webtoolkit/download.html
  - Download ZIP file and unzip into folder of your choice
    - Optional because you can get it bundled with Eclipse plugin
- **(Optional) Install Speed Tracer**
  - Performance profiler. See top-level download page from first bullet. Not discussed here.
Details: Eclipse Plugin (Eclipse 3.7 Indigo)

- **Go to Eclipse “Install New Software” page**
  - Start Eclipse (3.7 – Indigo – is latest version)
  - Help → Install New Software → Work With

- **Load plugin**
  - Enter http://dl.google.com/eclipse/plugin/3.7
    - http://code.google.com/eclipse/docs/getting_started.html
      has info for other Eclipse versions
  - Press “Add”, then “OK”
    (you can leave name blank)
  - Select all Plugin and SDK options
    - You can exclude GAE Tools for Android
  - Press “Next”, “Finish”, Restart Eclipse

GWT Designer

- **Idea**
  - Optional visually-oriented GWT builder
  - Formerly a commercial product. Purchased from instantiations.com and released for free in 2010

- **Features**
  - Rich drag-and-drop GUI builder
    - Many, many widgets and panels supported
    - Support for GXT (Ext-JS in GWT)
  - Many wizards
    - Project creation
    - RPC
  - Details
    - http://code.google.com/webtoolkit/tools/gwtdesigner/
Creating Eclipse Projects

Customized Java EE Training: http://courses.coreservlets.com/
GWT, Java, JSF 2, PrimeFaces, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android.
Developed and taught by well-known author and developer. At public venues or onsite at your location.

General Application Development Approach

• **Create a new project**
  – Click “g” (New Web Application Project) button at top
  – Enter project name and base package name
  – Delete extraneous files

• **Edit auto-generated HTML file**
  – war/MyApp.html

• **Edit auto-generated Java file**
  – src/package.client/MyApp.java

• **Test in Development Mode**
  – R-click project, Run As ➔ Web Application (or Debug As…)

• **Test in Production Mode**
  – R-click project, Google ➔ GWT Compile

• **Deploy to Web server**
  – Copy and rename “war” folder, deploy to any Java server
Making and Testing Sample Project

• Create a new project
  – Click g (New Web Application Project) button at top
  – Or File → New → Web Application Project
  – Enter project name and base package name

• Test in Development Mode
  – R-click project, Run As → Web Application
  – Or, click Run arrow at top
  – “Development mode” means that client-side code runs in Java in special browser plugin, so Eclipse Java debugger can be used, print statements go to Java console, and changes are deployed automatically

• Test in Production Mode
  – R-click project, Google → GWT Compile
  – Or, click red toolbox (GWT Compile) under on toolbar
  – “Production mode” means that client-side Java code is compiled to JavaScript and then executed in normal browser

Creating a New Project

• Click “g” at top
  – Or File → New → Web Application Project

• Enter info
  – Choose project name and base package name
  – Use GWT 2.4 as SDK
  – Unless you will deploy to Google cloud, it is not required to select Google App Engine
  – Press Finish
Resulting Eclipse Project

- **MyGwtApp**
  - **src**
    - myPackage
      - myPackage.client
        - GreetingService.java
        - GreetingServiceAsync.java
      - MyGwtApp.java
    - myPackage.server
      - GreetingServiceImpl.java
    - myPackage.shared
      - FieldVerifier.java
  - **test**
    - GWT SDK [GWT - 2.4.0]
    - JRE System Library [jdk1.7.0]
  - **war**
    - mygwtapp
    - WEB-INF
      - deploy
        - lib
          - web.xml
          - MyGwtApp.css
          - MyGwtApp.html

Client-side code must be in “client” package. Server-side code can also import and use classes in this package (other than main app). All code in this package will eventually get translated into JavaScript.

Samples. You can use these files as examples upon which to model your code, but in real projects you will usually delete these samples and replace them with your own files.

Starting point client-side code (“main”). You will edit this extensively.

If you have done “GWT Compile” (see later slides on Production Mode), you can find this folder on the file system, copy it, rename it to “MyGwtApp” (or any other name), and deploy it to a Java-based server.

Created the first time you run the app. When you do “GWT Compile” (see later slides on Production Mode), this folder gets populated with JavaScript files.

Starting point HTML file. You will edit this extensively.

Testing Projects

Customized Java EE Training: [http://courses.coreservlets.com/](http://courses.coreservlets.com/)

GWT, Java, JSF 2, PrimeFaces, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android. Developed and taught by well-known author and developer. At public venues or onsite at your location.
Installing the Browser Plugin

**Idea**
- Google created browser plugin that lets your GWT apps run in Java and communicate with Eclipse
- The first time you use a given browser for GWT, you are prompted to install the plugin
  - Versions exist for Firefox, IE, Safari, and Chrome

**Steps**
- R-click on project name at left
- Run As → Web Application
  - Or, if you set breakpoints, Debug As → Web Application
- R-click on URL that is shown in tab at bottom, use URL in browser
- Paste this URL into browser
- Follow browser plugin installation instructions

Installing GWT Developer Plugin on Firefox

**Problem:**
- GWT Developer Plugin sometimes lags behind Firefox releases
  - Especially when Mozilla starts releasing new major version every 1-2 months (as with FF 6 and FF 7)
  - As of now (end of October, 2011) Firefox 7 is not supported

**Solution:**
- Install latest Firefox 6.x side-by-side Firefox 7
- To download earlier versions of Firefox, go to: ftp://archive.mozilla.org/pub/mozilla.org/firefox/releases/
Installing GWT Developer Plugin on Firefox (Continued)

- **Important! – Before install, go to current FF**
  - Go to Options – Advanced – Update and uncheck Firefox for Automatically check for updates
- **Installing earlier version of Firefox**
  - Setup Type – choose Custom
  - Choose Install Location – choose non-default location
    - E.g., C:\Program Files (x86)\Mozilla Firefox 6\ instead of leaving it as C:\Program Files (x86)\Mozilla Firefox\
  - Summary – uncheck Use Firefox as my default browser
  - On first launch, do not make this your default browser when it asks & tell it not to ask you anymore

Testing the Project in Development Mode

- **Idea**
  - New projects have sample behavior built in
- **Steps**
  - R-click project and Run As Web Application
    - Or, press green Run arrow at top
  - Open With or cut/paste URL into browser
  - Once you have installed Google browser plugin, app will run automatically

Note: “Development mode” means running in Java using browser plugin. IDE has embedded version of the Jetty server.
- Code almost always runs identically once compiled to JavaScript and deployed in a regular browser, but Development mode lets you use the Java console (System.out.println), Eclipse debugger (breakpoints, etc.), and so on. It also means you can change Java code, save the file, and see new result just by clicking Reload in browser.
Making Changes to Project

• Idea
  – Main Java file to edit: src/package.client.YourApp.java
  – Main HTML file to edit: war/YourApp.html

• Steps
  – Make a few changes, save, go to browser (already in development mode) and hit Reload to see results.
  – For now, just guess on what changes you can make. Details given in next tutorial section.
    • Java: change some of the labels (don’t change widget class names or main method name)
    • HTML: change some of the text (don’t remove or rename the JavaScript file; don’t change the ids of the two td elements that have ids)

Making Changes to Project

• HTML (war/MyGwtApp.html)
  – Changed title of heading
    • From: <h1>Web Application Starter Project</h1>
    • To: <h1>Auto-Created Sample Project</h1>
  – Changed name prompt
    • From: <td …>Please enter your name:</td>
    • To: <td …>Enter sample name:</td>

• Java (myPackage.client.MyGwtApp.java)
  – Button name (in onModuleLoad near top of file)
    • From: final Button sendButton = new Button("Send");
    • To: … = new Button("Send to Server");
  – Dialog box title (in onSuccess near bottom of file)
    • From: dialogBox.setText("Remote Procedure Call");
    • To: …setText("Response from Server");
New Result

• Note
  – Did not restart development mode. Just saved files, went to browser, and hit the Reload button.

Testing a Different Project in Development Mode

• Before testing a new project, stop Jetty
  – When you run a project in development mode, the Jetty Java server (servlet engine) is automatically started on port 8888.
  – Stop Jetty before starting a new project. Otherwise you will get error message about port being in use.
    • Stop Jetty by clicking red square at top right of Development Mode tab at bottom of Eclipse.
Testing Project in Production Mode

• Idea
  – GWT will compile all of your client-side Java code to JavaScript and run in a regular browser with no special plugin
    • Development mode is much more useful for most development and debugging, but you should periodically check that code works (and especially looks) the same in pure client-side JavaScript. In Production Mode, changes to client-side Java code are not automatically reflected when you reload page.
    • Note that the server-side code is still handled by the bundled Jetty server. Later slides will talk about running in an arbitrary Java server.

• Steps
  – Run 1st in development mode (so Jetty server is running)
    • Can skip this for simple test apps that have no server code
  – R-click project, Google  GWT Compile
  – Enter http://localhost:8888/YourProject.html in a browser
    • Or just http://localhost:8888/ since YourProject.html is default file

Testing Project in Production Mode

• Ran in development mode
  – R-clicked project, Run As  Web Application

• Compiled client-side code to JavaScript
  – R-clicked project, Google  GWT Compile
    • Or clicked red toolbox at top

• Ran in normal browser
    • Can invoke this URL by deleting ?gwt.codesvr=... from end of Development Mode URL.
    • If you want to test only client-side code, you don’t need to run in development mode first. GWT  Compile and then drag war/YourProject.html to browser
Deploying the Project to Normal Java App Server

**Idea**
- Once you have done GWT Compile, the “war” folder contains a portable exploded Web app

**Deploying as exploded app (folder)**
- R-click and Google → GWT Compile to build JavaScript files
- Copy “war” folder to another location
- Rename “war” folder (e.g., mygwtapp)
- Copy new folder to server’s deployment directory
  - URL: http://servername/mygwtapp/

**Deploying as zipped app (WAR file) - manually**
- R-click and Google → GWT Compile to build JavaScript files
- Create empty ZIP file (e.g., mygwtapp.zip)
- Copy contents of “war” folder into ZIP file
- Change extension to .war (e.g., mygwtapp.war)
- Copy WAR file to server’s deployment directory
  - URL: http://servername/mygwtapp/

Deploying the Project to Normal Java App Server (continued)

**Deploying as zipped app (WAR file) - Eclipse**
- R-click and Google → GWT Compile to build JavaScript files
- R-click on the war directory in Eclipse, choose Export…
- Choose Archive File and click Next
- Select only files/folder under the ‘war’ folder, but not the ‘war’ folder itself & select all the files on the right
- You can give the location of your server’s autodeploy directory here.
- Make sure the file extension is .war NOT .zip (Eclipse will fill in .zip automatically.)
- Make sure Create only selected directories is selected! The other option is the default one.

URL: http://servername/mygwtapp
Deploying the Project to Normal Java App Server: Results

Documentation

- **GWT overview page**
  - Good starting point with links to tutorials and docs
    - http://code.google.com/webtoolkit/overview.html

- **Tutorials**
  - Includes Getting Started and many specialized topics

- **Reference guide**
  - Includes JavaDoc API, guide to which Java features are supported on client side, and widget library reference

- **Developer Forum**
  - Pretty active with experts to answer questions
    - http://groups.google.com/group/Google-Web-Toolkit
Documentation (Continued)

• Books
  – See http://code.google.com/webtoolkit/books.html
  – Very few are worth it as all but one are out of date
    • http://code.google.com/webtoolkit/books.html
  – Most GWT learning is through GWT web site

Wrap-Up

Customized Java EE Training: http://courses.coreservlets.com/
GWT, Java, JSF 2, PrimeFaces, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android.
Developed and taught by well-known author and developer. At public venues or onsite at your location.
Summary

- **Setup**
  - Install Eclipse plugin
    - [http://code.google.com/eclipse/docs/getting_started.html](http://code.google.com/eclipse/docs/getting_started.html)
    - This will install GWT 2.4 at the same time
  - Bookmark [http://code.google.com/webtoolkit/overview.html](http://code.google.com/webtoolkit/overview.html)

- **Create Eclipse project**
  - Press “g” button at top
    - Or File → New → Web Application Project

- **Test**
  - Test in development mode (always)
    - R-click → Run As → Web Application
      - Or click Run button at top
  - Test in production mode (occasionally)
    - R-click → Google → GWT Compile
  - Deploy to normal Java server (at end)
    - Copy/rename “war” folder or put contents into .war file

Questions?

Customized Java EE Training: [http://courses.coreservlets.com/](http://courses.coreservlets.com/)

© 2012 Marty Hall & Yaakov Chaikin

JSF 2, PrimeFaces, Java 7, Ajax, jQuery, Hadoop, RESTful Web Services, Android, Spring, Hibernate, Servlets, JSP, GWT, and other Java EE training.

Developed and taught by well-known author and developer. At public venues or onsite at your location.