



Google
Developers



GWT Roadmap for the Future

Ray Cromwell, Daniel Kurka



What's Happened since I/O 2012

Changes to GWT Project

- New Git Source Repository
 - <http://gwt.google.com>
- New (and faster) Code Review System
 - <http://gwt-review.google.com>
- Mirroring with GitHub
 - <http://github.com/gwtproject>
- RedHat now hosting the Continuous Integration server for GWT
 - <http://build.gwtproject.org>
- New Home For GWT Website
 - <http://gwtproject.org>



Steering Committee Updates

- JetBrains joins the Steering Committee!
- Steering Committee approves GWT Roadmap for 2.6+ and beyond.
- Start of decentralization: Pieces of Roadmap now have external owners
 - e.g. Vaadin for IE10, Thomas Broyer for Mavenization and RequestFactory



Releases

- GWT 2.5.1 released (mostly bug fixes)
- Improvements:
 - DirectInstallLinker now works with Content Security Policy (Chrome Apps and Extensions)
 - SuperDevMode partially supports IE8, faster startup, and uses gzip for serving code
 - RequestBuilder now supports HTTP CORS Requests
 - Support for @media selectors in CssResources
 - Improved quality for resized images in ClientBundle
 - Upgraded IC4UJ and CLDR support to latest versions
 - Improved memory usage of StackTraceDeobfuscator
 - Minor Compiler Optimization improvements





The Future

Agreed Upon Roadmap Principles and Areas of Focus

- Openness and Simplicity
- Speed
- Interoperability
- Mobility
- Reliability
- Embeddability



Openness and Simplicity

- Complete the move to Open Source
 - GWT Steering Committee members allowed as direct committers
 - Public Continuous Integration server / BuildBot
 - Full Mavenization
 - Regular, Public Steering Committee meetings
- Simplicity
 - Disentangle dependencies, make GWT more modular
 - Reduce need for .gwt.xml files
 - Delete @Deprecated code



Speed

- Increase compile speed by 50%
- Improve speed of SuperDevMode refresh
- Continue to improve compiled code size
- Improved CodeSplitter
- Tune JS output for modern JS VMs
- Better reporting and profiling tools to spot performance problems



Interoperability

- Java <-> Javascript calls even easier to use
- Improve support and integration with Closure Compiler
- Support hybrid apps, compiling GWT code and external JS libraries together
- Java 7 and Java 8 support!



Reliability

- Close 100 of our top bugs
- Improve speed and reliability of GWT unit testing
- Deprecate older browsers (ie6/7/8?)



Embeddability

- Break up monolithic GWT SDK into smaller modules
- More "hook points" for integration with other tools



Mobility

Mobile is increasingly becoming more important

- Support modern mobile web browsers
- Mobile Optimized Widgets
- Application Life Cycle and Offline
- Packaged apps





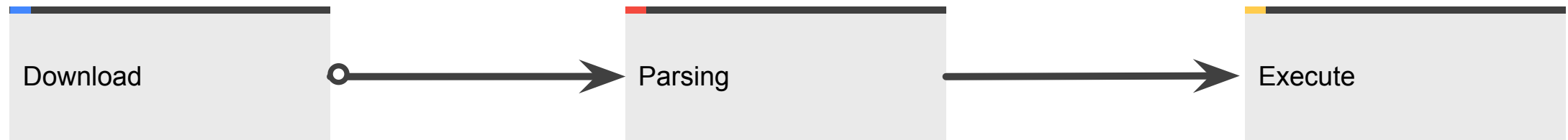
GWT & Mobile

GWT & mobile

- Performance matters on mobile
 - slow CPUs
 - slow networks
 - running on battery
- GWT is in a unique position regarding performance
 - GWT knows all resources thus can optimize all of them
 - this is very important for mobile performance
- [Ilya Grigorik on web performance basics](#)
- Different kinds of performances
 - Startup
 - Runtime
 - Resource usage



Startup performance on mobile



Startup performance - Download

- Gzip for HTML, CSS and Javascript => ServletFilter
- Async, defer scripts => easy with GWT
- HTTP Headers for caching => easy with GWT
- Offline Support => custom Linker
- Minification for JavaScript and CSS => GWT Compiler + CssResources (remove selectors)
- Spriting / Inlining => ClientBundles + ImageResources
- Code Splitting? => GWT.runAsync()
- Batch data calls => command pattern with Scheduler (e.g. gwt-dispatch)



Startup Performance - Parsing & Executing

- Parsing
 - Parsing Javascript blocks the UI
 - All resources need to be downloaded & parsed before initial painting
 - HTML \Leftrightarrow DOM
 - CSS \Leftrightarrow CSS tree
- Executing
 - Execute JavaScript & Setup DOM
 - DOM + CSS \Rightarrow Render Tree
 - Simple DOM means small render tree \Rightarrow optimized mobile widgets
 - Simple CSS means better DOM / CSS matching performance \Rightarrow CcssResouce optimizations



Runtime performance

- "Native" layout
 - All layout in CSS
 - flexible box model => [Eric Bidelmann IO 12](#)
 - => case for mobile optimized widgets
- "Native" animations
 - No animations in javascript
 - Animations IN CSS => Execution on the GPU
 - => case for mobile optimized widgets



GWT & performance

- GWT is designed for performance
- Google production cares deeply about performance
- Mobile needs good performance
- GWT for mobile is a no brainer



GWT for writing apps - Phonegap

- GWT apps run on all modern phones & tablets using Phonegap
- Native look and feel
- Native APIs through web standards
- Same code for phones & tablets & desktop
 - Same Presenters
 - Same Models
 - Same RPC
 - ...
 - Different Views
 - Different navigation
- [mgwt](#) - mobile GWT widgets
- [gwt-phonegap](#) - using phonegap from GWT apps



Mobile demos





gwtproject.org

gwtproject.org

- all GWT related stuff in one place
- open source
- documentation
 - markdown
 - convert markdown to HTML using maven
- Contribute to the webpage & docs
 - same review process
 - same guidelines
 - talk to us if you got a cool idea



Start Contributing!

- *git clone https://gwt.googleusercontent.com/gwt-site*
- Fix docs, contribute docs or articles
- *mvn clean install*
- look in target/generated-site for the result
- *git commit -am 'I fixed your docs'*
- *git push origin HEAD:refs/for/master*
- Wait for review
- Visit <http://gwt-reviews.googleusercontent.com>
- If you receive a positive review, commit!
- Watch the website update
- <http://gwtproject.org/makinggwtbetter.html#webpage>





TABLE OF CONTENTS

This is a **beta version**. If you find errors, please [report them](#) or [change them](#)!

- Home
- ▶ Articles
- ▶ Documentation
- Books
- Case Studies
- Community
- Community Group
- Developer Spotlight
- Download
- Examples
- Getting Started
- SDK Learn More
- Life of an Issue
- Making GWT Better
- Overview
- Presentations
- Privacy
- Release Notes
- Resources
- Steering Committee
- Terms
- Using Eclipse

GWT

GWT (pronounced 'gwit') is the official open source project for GWT releases 2.5 and onwards. This site houses links to the documentation, source code repository, issues list and information related to GWT roadmap and release. It is intended for developers interested in contributing to GWT, and for keeping people informed on new and upcoming changes to GWT, GWT related events and other news.

Productivity for developers, performance for users

GWT is a development toolkit for building and optimizing complex browser-based applications. GWT is used by many products at Google, including [Google AdWords](#) and [Google Wallet](#). It's open source, completely free, and used by thousands of developers around the world.



Learn More

Learn about GWT, the features and tools it offers, and how you can quickly develop performance AJAX applications across all major browsers.

When?

A Tick/Tock Schedule





**Thanks:
Steering Committee
Redhat,
Vaadin,
Julien Dramaix (GwtQuery)**



**Special Thanks To:
Thomas Broyer**

Summary

- Faster Development
- In an open setting
- For the modern, and mobile, web
- With fewer surprises
- More compatibility with Java
- And Easier Interoperability with the Rest the World





Questions?

<Thank You!>

Questions?

Homepage: <http://gwtproject.org>

Twitter: [#gwt](#)

Google+: [GWT Community](#)





Google
Developers