

What's New in GWT?

David Chandler
Google Web Toolkit Team
Atlanta, GA USA
drfibonacci@google.com



Agenda

Why rich Web apps with GWT?

GWT Quickstart

Developer tools

GWT performance for your users

Building with GWT 2.4.2

GWT in 10 sec

Aynchronous
Java~~Script~~
And
XML++

GWT in 60 sec

Open source Java to JavaScript compiler

Lets you write rich Web apps in Java

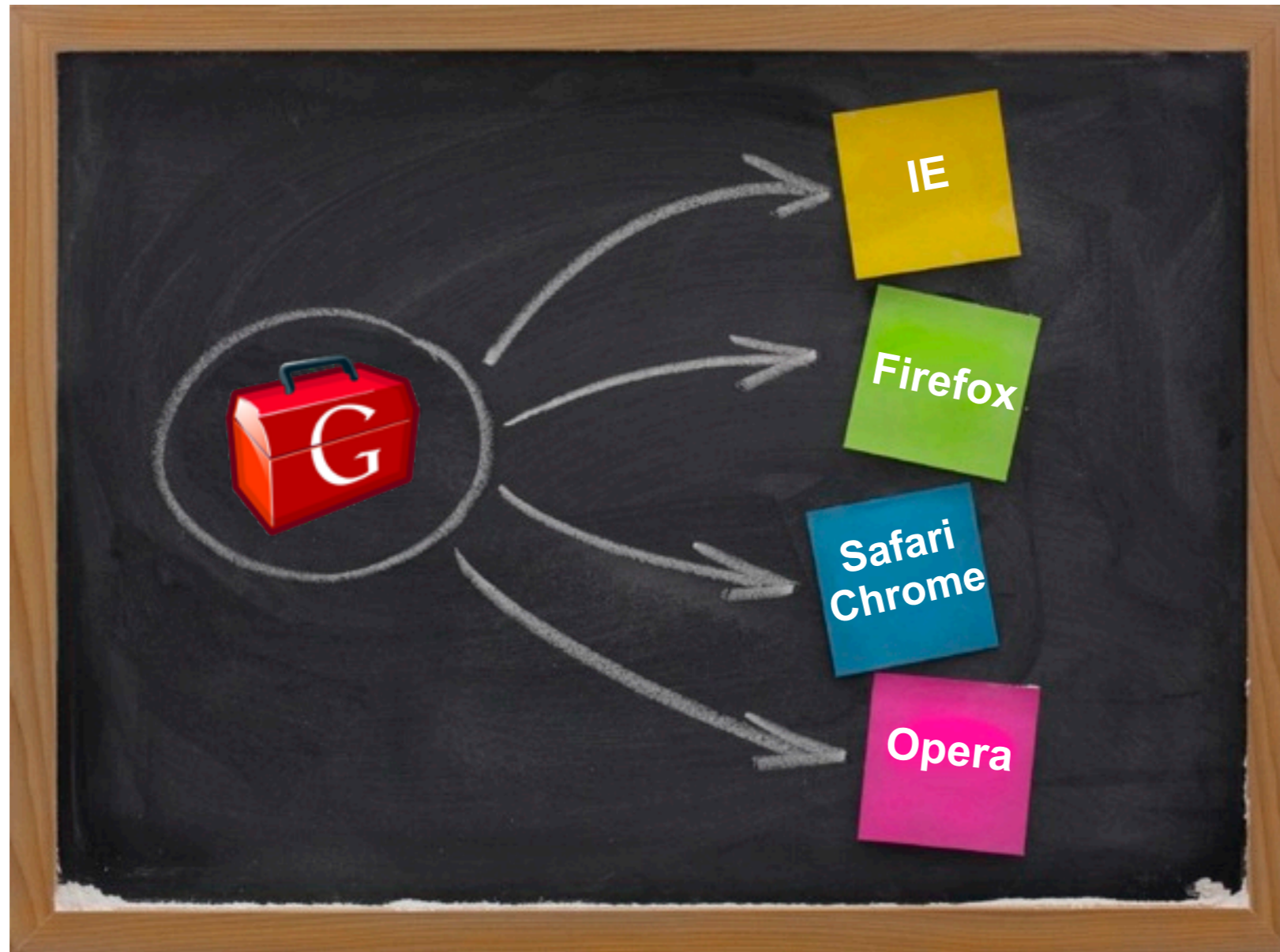
Cross-browser just works (FireFox, Chrome, Safari, IE 6+)

Produces small, fast JavaScript

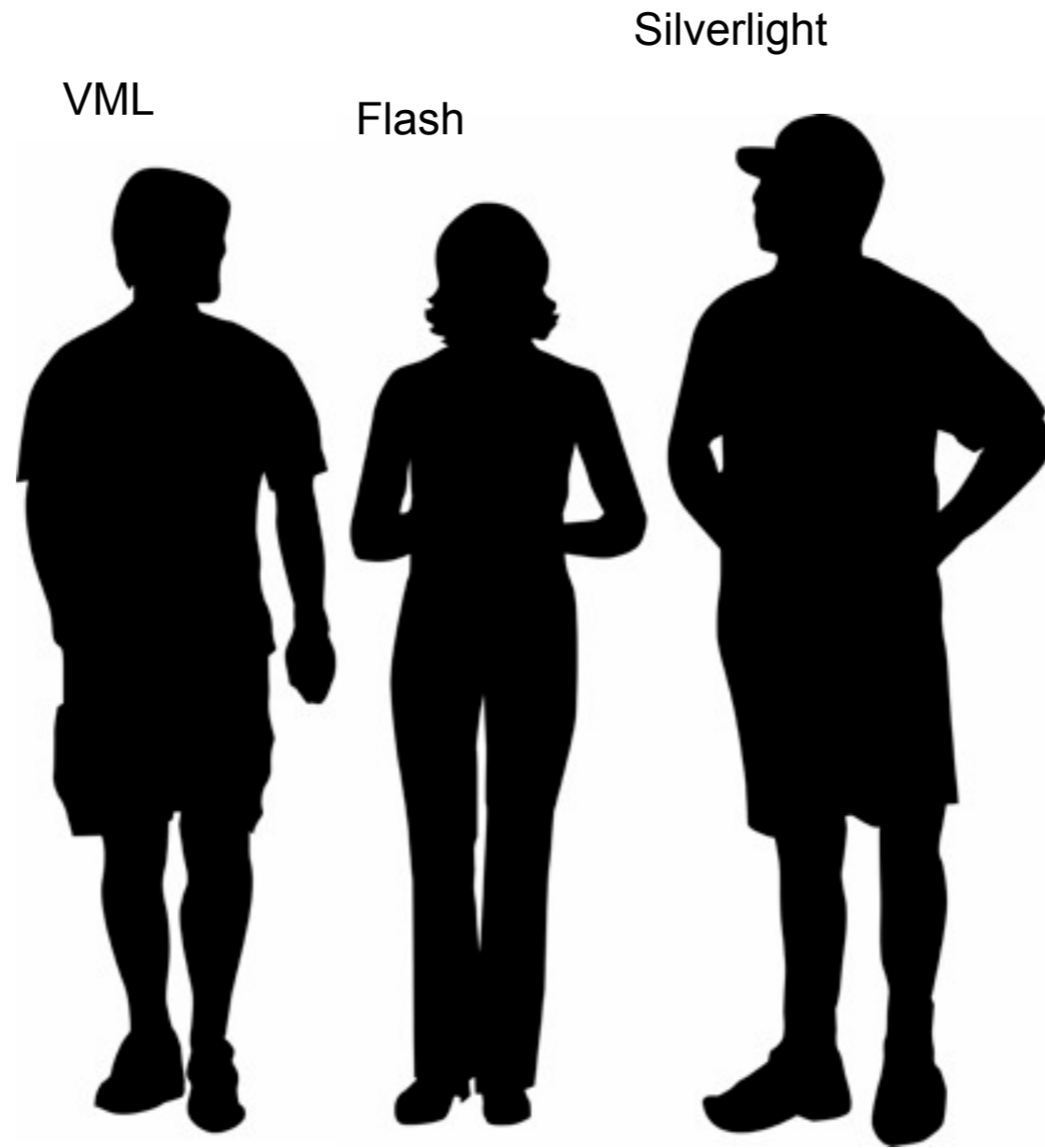
Easy (and efficient) RPC

Great for large projects / teams

Browser-Proof Your JS Code



No plugins required



Can you find the bug?

```
1 // JavaScript function to determine the maximum
2 // value among the provided array elements
3 //
4 // Can you find the bug?
5
6 function getMax(values) {
7
8     var maximum = values[0];
9
10    for ( var i = 1; i < values.length; i++) {
11        if (values[i] > maximum) {
12            maximum = values[i];
13        }
14    }
15
16    return maximum;
17 }
```

Catch errors at compile time

```
5  /**
6   * Java method to determine the maximum
7   * value among the provided array elements
8   *
9   * Our IDE already found the bug
10  */
11  public int getMax(int[] values) {
12
13      int maximum = values[0];
14
15      for (int i = 1; i < values.length; i++) {
16          if (values[i] > maximum) {
17              maxinum = values[i];
18          }
19      }
20
21      return maximum;
22  }
```


Completion, refactoring...

```
26  /**
27   * This is the entry point method.
28   */
29  public void onModuleLoad() {
30      Button button = new Button();
31      button.set
32
33      RootPanel.
34  }
35  }
36
```

- `setAccessKey(char key) : void - FocusWidget`
- `setEnabled(boolean enabled) : void - FocusWidget`
- `setFocus(boolean focused) : void - FocusWidget`
- `setHeight(String height) : void - UIObject`
- `setHTML(String html) : void - ButtonBase`
- `setPixelSize(int width, int height) : void - UIObject`
- `setSize(String width, String height) : void - UIObject`
- `setStyleName(String style) : void - UIObject`
- `setStylePrimaryName(String style) : void - UIObject`
- `setTabIndex(int index) : void - FocusWidget`
- `setText(String text) : void - ButtonBase`
- `setTitle(String title) : void - UIObject`
- `setVisible(boolean visible) : void - UIObject`
- `setWidth(String width) : void - UIObject`
- `setVisible(Element elem, boolean visible) : void - UIObject`

Press 'Ctrl+Space' to show Template Proposals

Clears all of the object's style names and sets it to the given style. You should normally use [setStylePrimaryName\(String\)](#) unless you wish to explicitly remove all existing styles.
See Also:
[setStylePrimaryName\(String\)](#)
Parameters:
style the new style name

Press 'Tab' from proposal table or click for focus

Eating our own dogfood

The screenshot shows the Google AdWords interface. At the top, the Google AdWords logo is on the left, and a yellow bar with a 'New' icon and 'Announcements (1)' is on the right. Below the logo is a navigation bar with tabs for Home, Campaigns, Opportunities, Reporting and Tools, Billing, and My account. A red banner at the top of the main content area reads: 'None of your ads are running because all of your active campaigns have ended. View'. The main content area is titled 'All online campaigns' and shows a date range of 'Last 7 days' from 'Nov 7, 2010 - Nov 13, 2010'. Below this is a sub-navigation bar with tabs for Campaigns, Ad groups, Settings, Ads, Keywords, Networks, and Audiences. A table below shows the campaign status. The table has columns for Campaign, Budget, Status, Clicks, Impr., CTR, Avg. CPC, and Cost. The data shows one campaign named 'US' with a budget of '\$5.00/day' and a status of 'Ended'. The total for all campaigns shows 0 clicks, 0 impressions, 0.00% CTR, \$0.00 Avg. CPC, and \$0.00 Cost.

Google AdWords

Home Campaigns Opportunities Reporting and Tools Billing My account

None of your ads are running because all of your active campaigns have ended. View

All online campaigns

Last 7 days
Nov 7, 2010 - Nov 13, 2010

Campaigns Ad groups Settings Ads Keywords Networks Audiences

All but deleted campaigns Segment Filter Columns Search

View Chart

Nov 7, 2010 Nov 13, 2010

+ New campaign Change status... Alerts

<input type="checkbox"/>	<input checked="" type="checkbox"/>	Campaign	Budget	Status	Clicks	Impr.	CTR	Avg. CPC	Cost
<input type="checkbox"/>	<input checked="" type="checkbox"/>	US	\$5.00/day	Ended	0	0	0.00%	\$0.00	\$0.00
Total - all but deleted campaigns					0	0	0.00%	\$0.00	\$0.00

+ = AdSense, Maps, Docs, Groups...

Demos

Booked In

Typing race

Rich ecosystem

www.gwtmarketplace.com

GWT Marketplace

Made with GWT for GWT

Home | Search

[login](#)

Categories

- [Consulting Services \(1\)](#)
- [Designers \(0\)](#)
- [Frameworks \(9\)](#)
- [JS Wrappers \(1\)](#)
- [Other \(1\)](#)
- [RPC \(0\)](#)
- [Security \(1\)](#)
- [Tools \(0\)](#)
- [UI Widgets \(0\)](#)

What is GWT Marketplace?

GWT Marketplace is a home for GWT developers to publish their products and GWT users to rate, comment and search on GWT products.

[Register a new product](#) or [Submit a bug or feature request](#)

New and Updated	
GWT Pages	9/3/10
Gwt-Platform	8/25/10
GWT Marketplace	8/23/10
Objectify-Appengine	8/23/10
GWT-dispatch	8/19/10
GWT-Presenter	8/19/10
Piriti	8/19/10
Guit	8/18/10
PureGWT	8/9/10
GWT-OpenLayers	8/9/10

Top Rated	
Gwt-Platform	5.00 (8)
Objectify-Appengine	5.00 (3)
AcrlS	5.00 (2)
GWT Pages	5.00 (2)
PureGWT	5.00 (1)
GWT-Presenter	5.00 (1)
GWT-dispatch	5.00 (1)
GWT Marketplace	4.75 (4)

Most Active

--

GWT Blog

- [\[8/30/10 4:00 PM\] How to Use Google Plugin for Eclipse with Maven](#)
- [\[8/25/10 9:20 AM\] Updated maven strategy for GWT](#)
- [\[Tue, 24 Aug 2010 23:30:00 -0700\] GWT 2.1 Milestone 3 is now available](#)
- [\[7/16/10 1:45 PM\] GWT 2.0.4 now available in maven central](#)

More ecosystem

Util: GIN, gwt-dnd, gwt-fx, gwt-comet, ...

Widgets: EXT-GWT, Smart-GWT, ...

Frameworks: Vaadin!

The screenshot displays a GWT application interface with several components:

- Log Viewer:** A window titled "gwt-log" showing a stack trace for a `ReferenceError` with the message `my_non_existant_variable`. It includes filters for TRACE, DEBUG, INFO, WARN, ERROR, FATAL, and OFF, along with a "Clear" button.
- Drag-and-Drop Interface:** A grid of 12 "Draggable" widgets arranged in 3 columns and 4 rows. A "Trash Bin" is visible on the left.
- Sound Support Matrix:** A table showing browser support for various audio formats. The user agent is identified as Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_5_7; en-us) AppleWebKit/4.0.2 Safari/530.19.
- Hornet Blast Game:** A simple game interface with a "GAME OVER" message, a ship, and a control panel with buttons for "P" (Pause), "↑" (Ship movement), "↓" (Ship movement), "←" (Ship movement), "→" (Ship movement), and "SPACE BAR" (Fire / Start Game).

MIME Type	Flash based support	Native browser or I
audio/basic	Unsupported	Supported via OBJECT
audio/mpeg	Supported	Supported via OBJECT
audio/x-aiff	Unsupported	Supported via OBJECT
audio/x-midi	Unsupported	Supported via OBJECT
audio/x-wav	Unsupported	Supported via OBJECT

4+ years in review

May 2006 GWT 1.0 Launch at JavaOne

... ..

Aug 2008 GWT 1.5 Java 5 language support

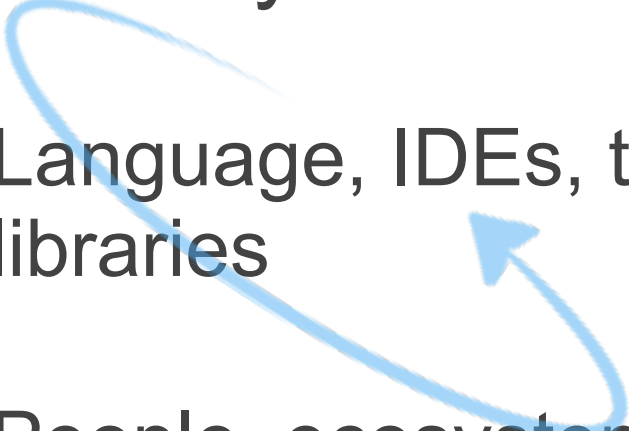
Apr 2009 GWT 1.7 Dedicated IE8 support

Fall 2009 GWT 2.0 UIBinder (XML template), runAsync()

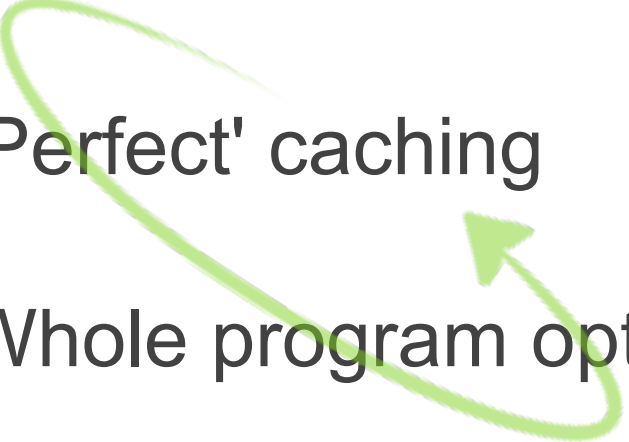
Oct 2010 GWT 2.1 MVP, RequestFactory, Spring Roo

GWT delivers...

Productivity for developers

- Language, IDEs, tools, libraries
 - People, ecosystem
- 

Performance for users

- 'Perfect' caching
 - Whole program optimization
- 

Developing with GWT



Develop

- Google Plugin for Eclipse, GWT Designer, STS / Roo



Debug

- In Eclipse with dev mode browser plugin



Optimize

- SpeedTracer for GWT, App Engine, Spring Insight



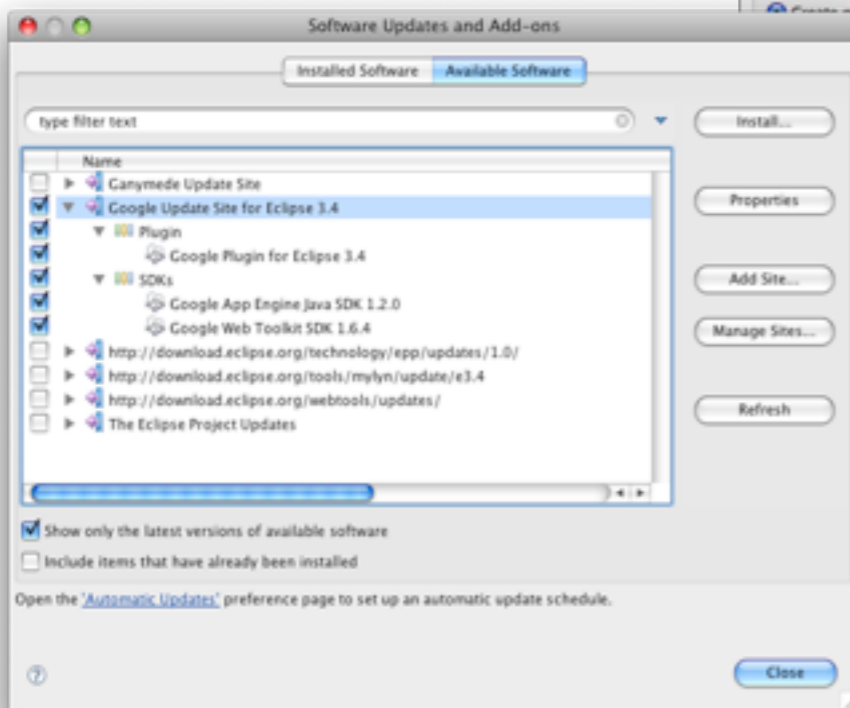
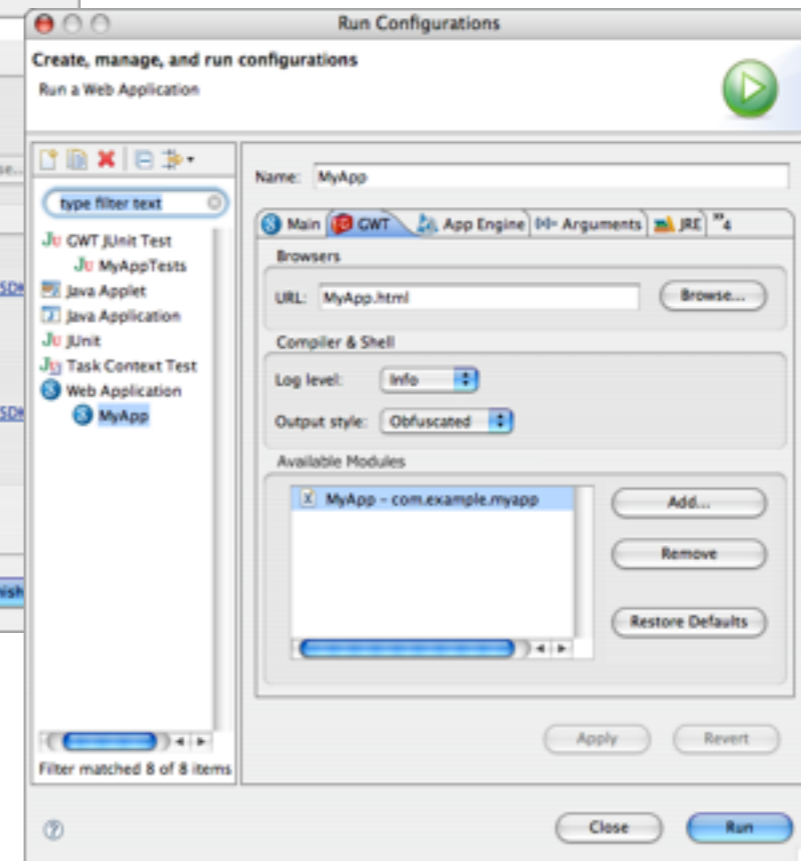
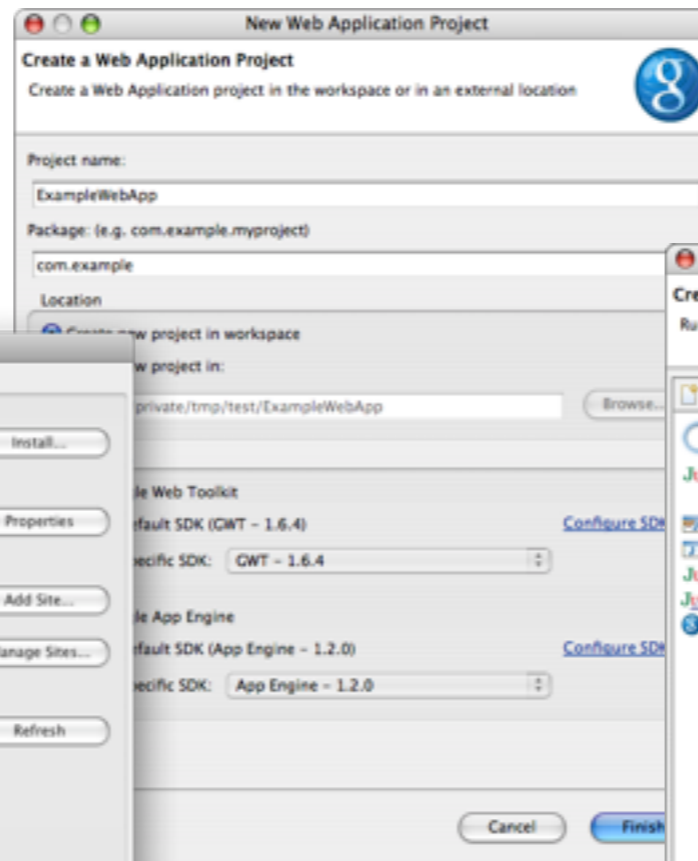
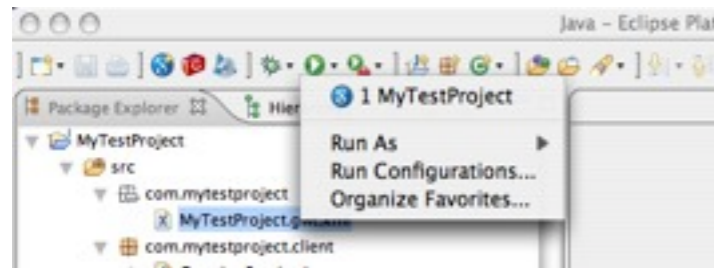
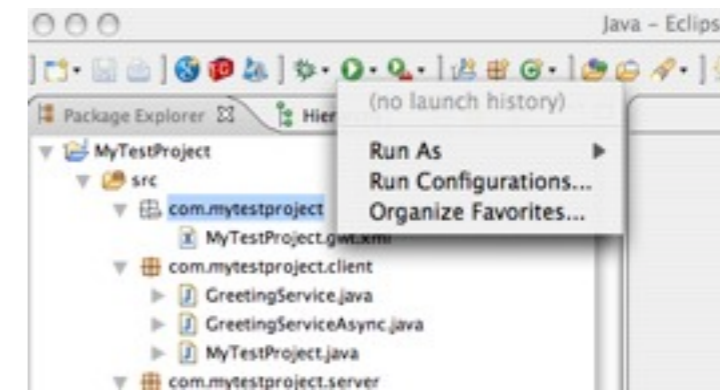
Deploy

- Standard WAR or one-button push to App Engine

Google Plugin for Eclipse



```
private static native void jsniMethod(boolean sayHi)/*- {  
    // Display a pop-up  
    if (sayHi) {  
        var name = this.@com.example.myapp.client.MyApp::name;  
        window.alert('Hello, ' + name);  
    }  
}/*-*/;
```



Google Plugin for Eclipse



Easiest way to get latest GWT, GAE SDKs

Debug / Run / Optimize / Deploy

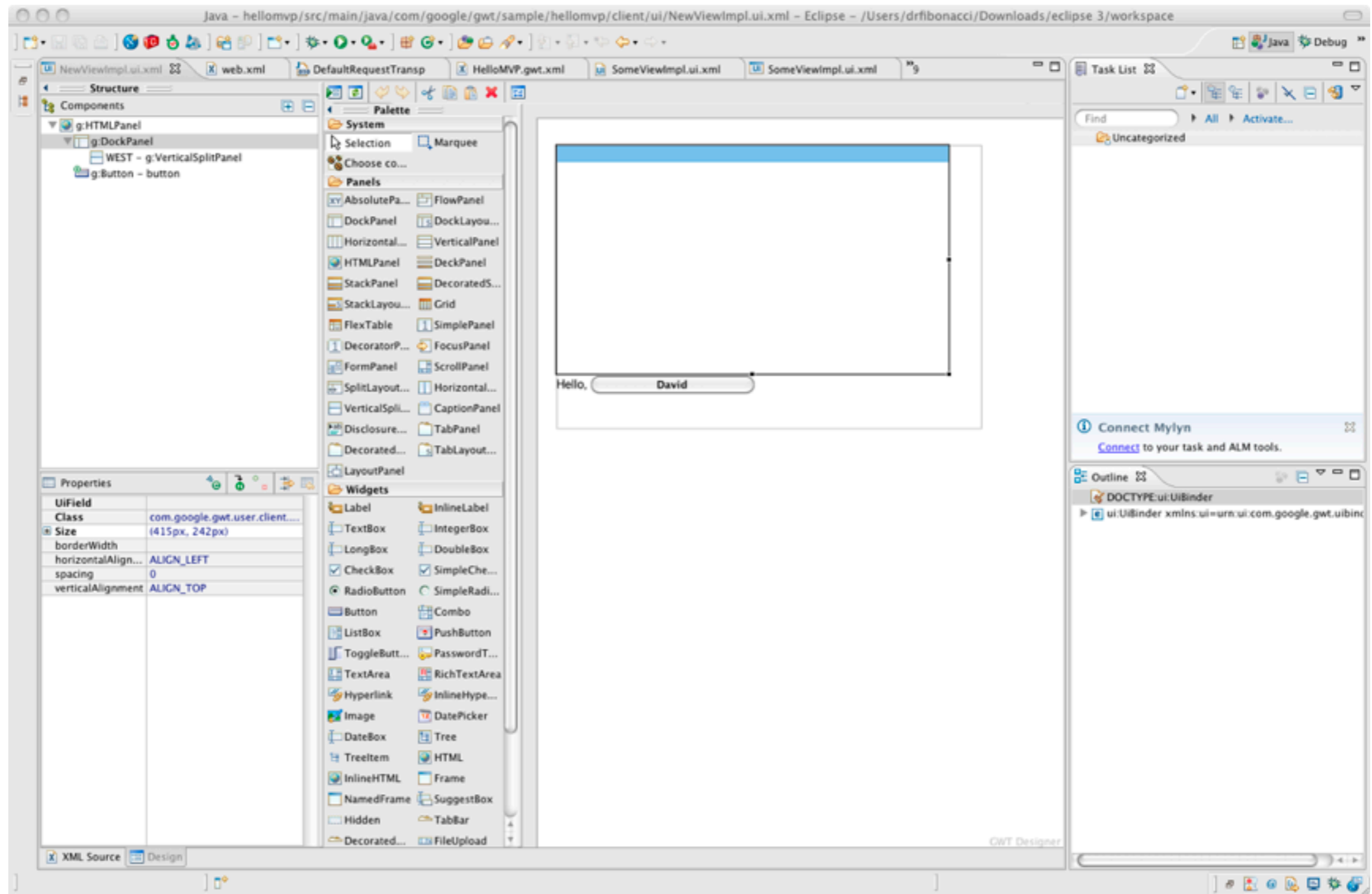
Wizards (new module, entry point, ClientBundle, UiBinder)

Quick fixes (auto-create RPC Async interface)

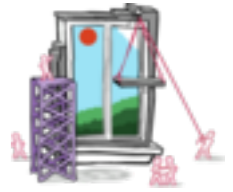
Run As | GWT JUnit Test

Maven integration via m2eclipse plugin

GWT Designer Java, ui.xml

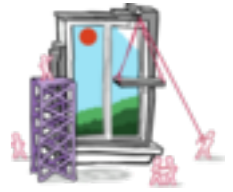


GWT Designer & UiBinder



```
<!DOCTYPE ui:UiBinder SYSTEM "http://dl.google.com/gwt/DTD/xhtml.ent">
<ui:UiBinder xmlns:ui="urn:ui:com.google.gwt.uibinder"
            xmlns:g="urn:import:com.google.gwt.user.client.ui">
  <ui:style>
    .important {
      font-weight: bold;
    }
  </ui:style>
  <g:HTMLPanel>
    <span class="{style.important}" ui:field="nameSpan" />
    <g:VerticalPanel height="203px">
      <g:HTML wordWrap="true">
        <h1>Hello</h1>
      </g:HTML>
      <g:Anchor ui:field="goodbyeLink" text="Say good-bye"></g:Anchor>
    </g:VerticalPanel>
  </g:HTMLPanel>
</ui:UiBinder>
```

GWT Designer CSS Editor



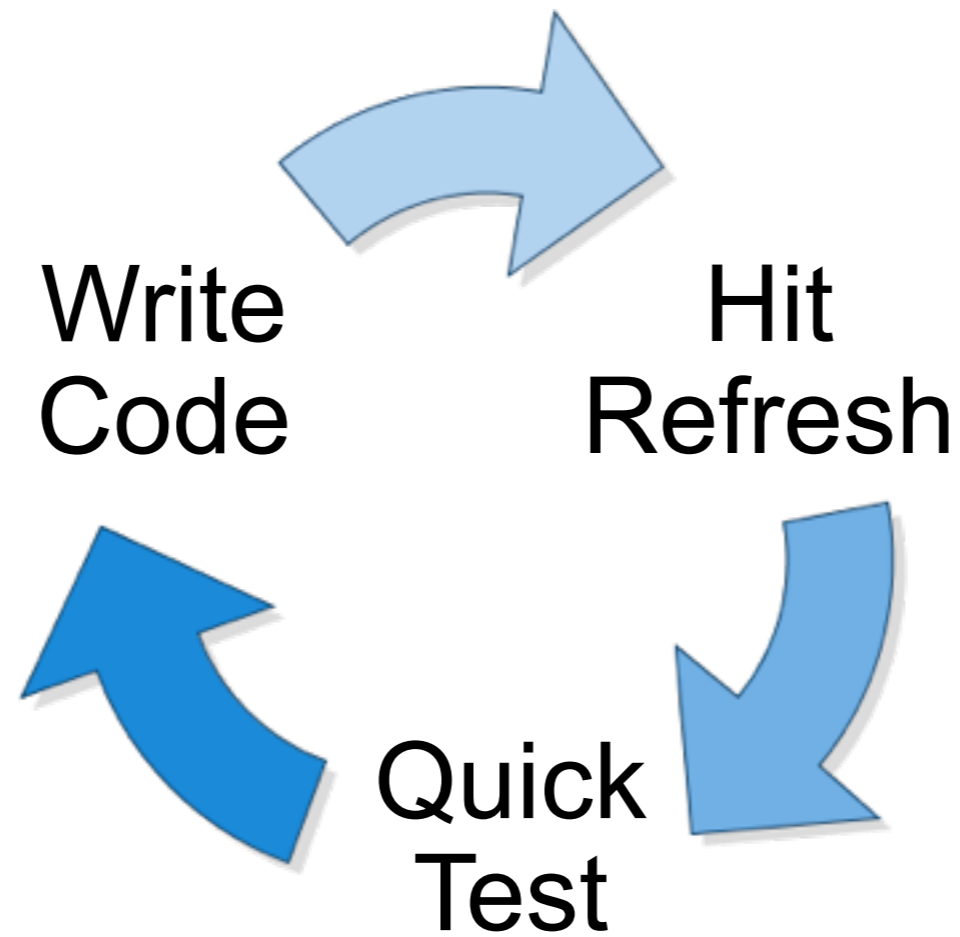
The screenshot displays the GWT Designer CSS Editor interface. On the left, a list titled "CSS rules in file" shows various selectors, with ".sendButton" selected. A "Selector filter" input field is positioned above the list. Below the list are buttons for "Add...", "Rename...", "Edit...", "Remove...", and "Sort".

The main area is divided into two panes. The left pane, titled "CSS Rule Editor: .sendButton", has tabs for "Font", "Background", "Box", "Border", "Text", and "Other". The "Font" tab is active, showing settings for Family (sans-serif), Size (16), Style (normal), Variant, Weight (bold), Stretch, and Color (Blue). A "Decoration" section includes checkboxes for none, underline, overline, line-through, and blink. A "Clear" button is at the bottom of this pane.

The right pane, titled "CSS rule", shows the CSS code for the selected rule: `.sendButton { display: block; font-size: 16pt; }`. Below the code is a preview window showing the text "Nolite mittere margaeritas ante porcas! *Latin proverb*" in blue, bold, sans-serif font.

At the bottom of the main editor area, there are "OK" and "Cancel" buttons. The bottom of the application window shows "Source" and "Design" tabs.

Development cycle

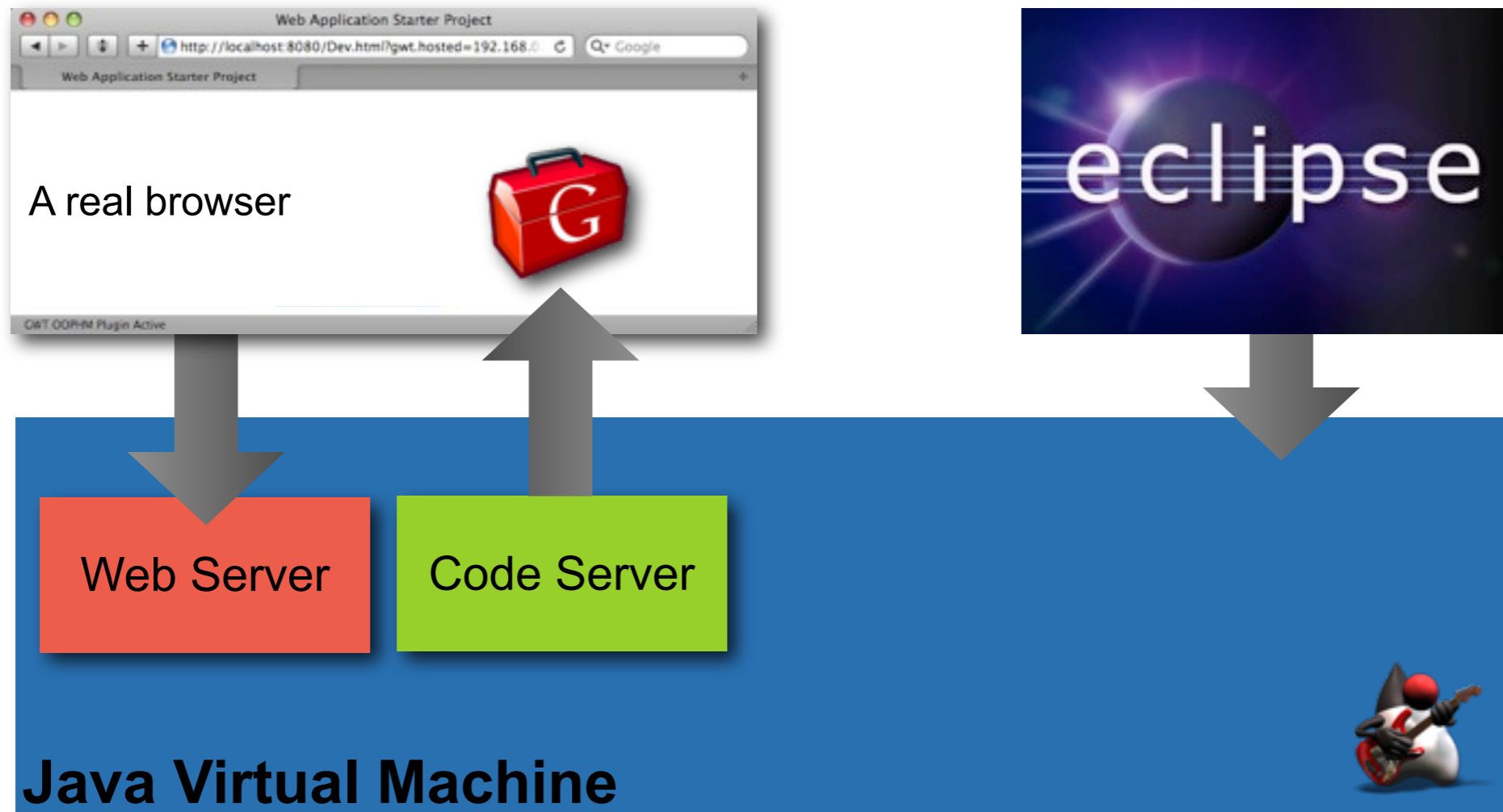


Demo

Dev mode

Set breakpoint

How dev mode works



Debug in dev mode

The screenshot displays the Eclipse IDE interface during a debug session. A browser window in the foreground shows a web application titled "Web Application Starter Project" with the URL `http://localhost:8080/Sample.html`. The page content includes the text "Please enter your name:" followed by a text input field containing "Fred" and a "Send" button. A red arrow points from the "Send" button in the browser to the corresponding code in the IDE.

The Eclipse IDE window shows the following components:

- Editor:** Displays the `Sample.java` file with the following code:

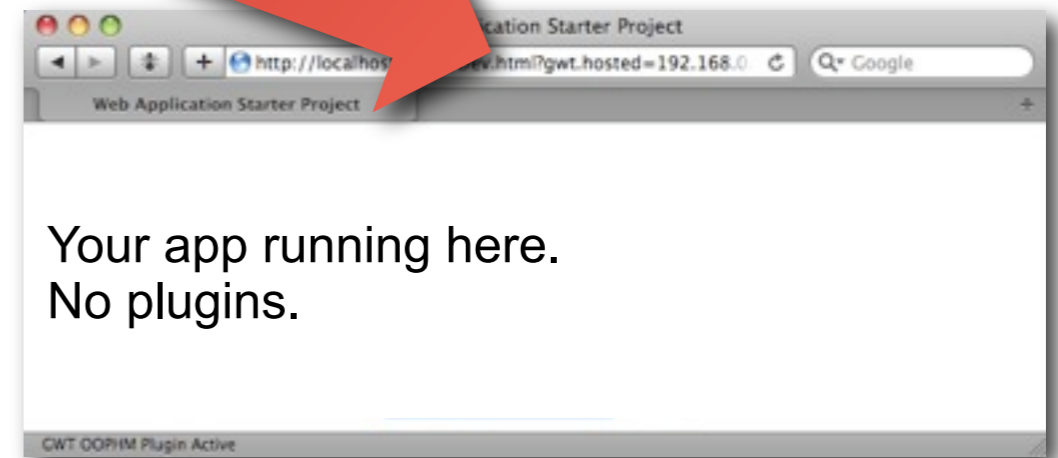
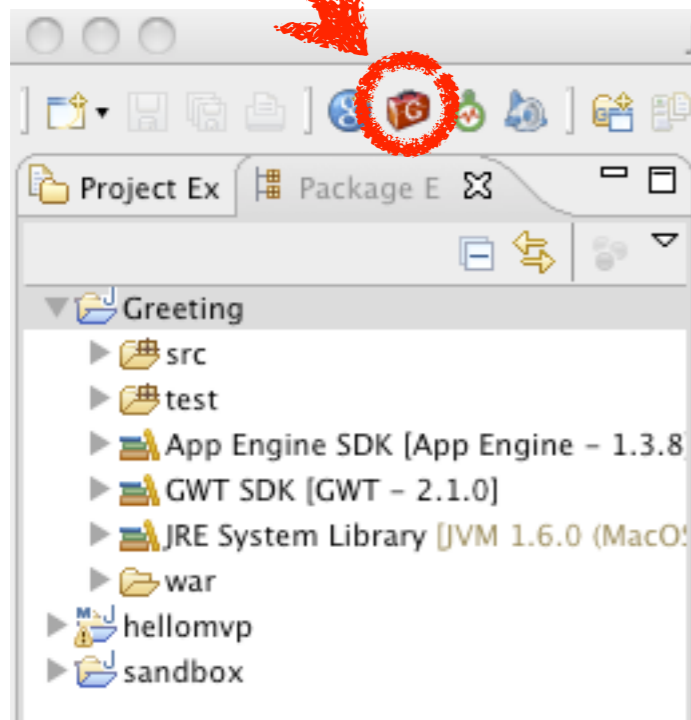
```
73 dialogBox.setWidget(dialogVPanel);
74
75 // Add a handler to close the DialogBox
76 closeButton.addClickHandler(new ClickHandler() {
77     public void onClick(ClickEvent event) {
78         dialogBox.hide();
79         sendButton.setEnabled(true);
80         sendButton.setFocus(true);
81     }
82 });
83
84 // Create a handler for the sendButton and nameField
85 class MyHandler implements ClickHandler, FocusHandler {
```
- Variables View:** Shows the state of variables at the current execution point:

Name	Value
this	Sample\$1 (id=48)
this\$0	Sample (id=65)
greetingService	GreetingService_Proxy (id=3189)
moduleBaseURI	"http://localhost:8080/sample/" (id=3190)
remoteServiceURL	"http://localhost:8080/sample/greet" (id=3191)
serializationPolicy	"29F4EA1240F157649C12466F01F46F60" (id=3192)
serializer	GreetingService_TypeSerializer (id=3193)
val\$dialogBox	DialogBox (id=67)
val\$sendButton	Button (id=77)
- Outline View:** Shows the project structure for `fredsa.sample.client`, including `import declarations`, `Sample`, `SERVER_ERROR : String`, `greetingService : GreetingServiceAsync`, `onModuleLoad()`, `new ClickHandler() {...}`, `onClick(ClickEvent)`, and `MyHandler`.

Compile for Production

Source files:
Code, CSS, Images,
Resources, ...

**GWT
Compiler**



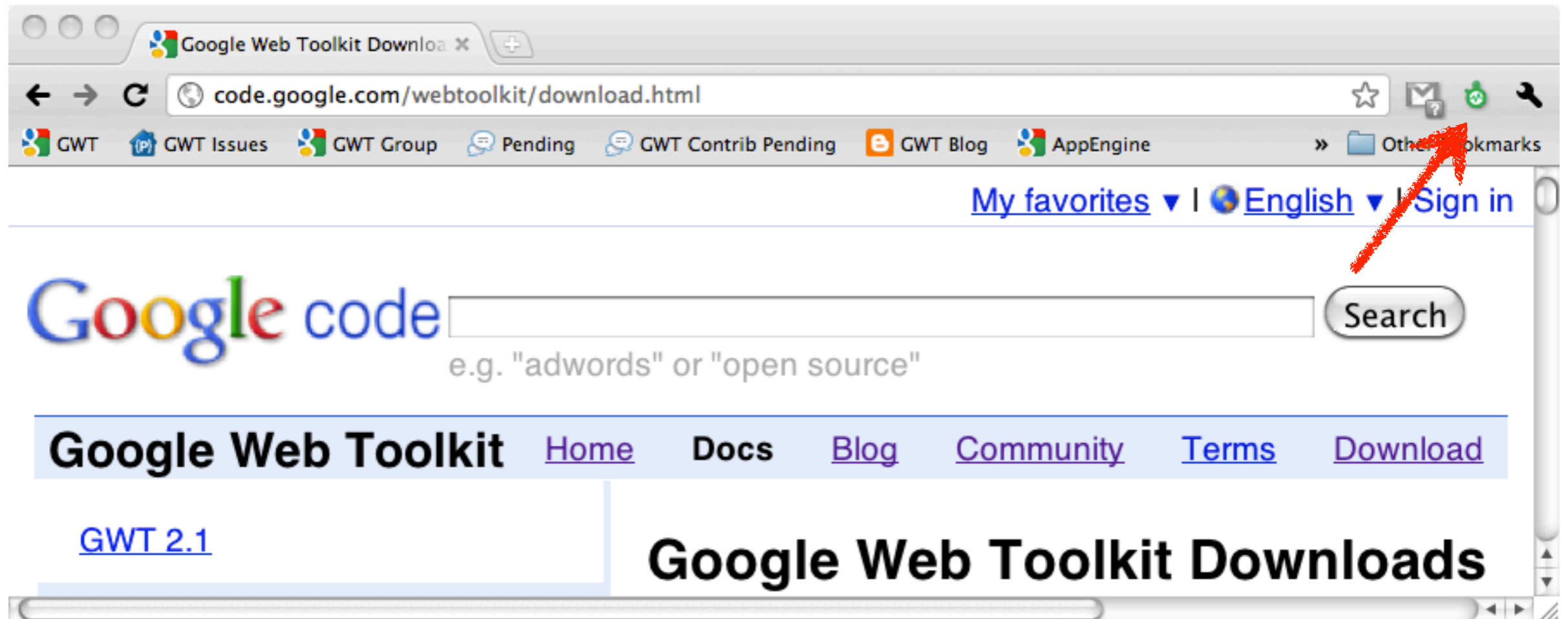
Optimize with Speed Tracer

The screenshot displays the Speed Tracer interface for a browser window. The top navigation bar includes links for GWT, GWT Issues, GWT Group, Pending, GWT Contrib Pending, GWT Blog, and AppEngine. The main area features a timeline graph with 'Sluggishness (events)' and 'Network (resources)' tracks. A zoomed-in view shows 'Resource Data Received @17.34s' with a duration of 19.683ms. A pie chart indicates the breakdown of this resource: 73.0% Script Evaluation, 20.4% Parse HTML, and 6.6% Resource Data Received. The 'Event Trace' section shows a sequence of events, with a detailed view of 'Resource Data Received 19.7ms (self 1.3ms)' and 'Parse HTML 18.4ms (self 4.0ms)'. The 'Details for Resource Data Received' table provides further information:

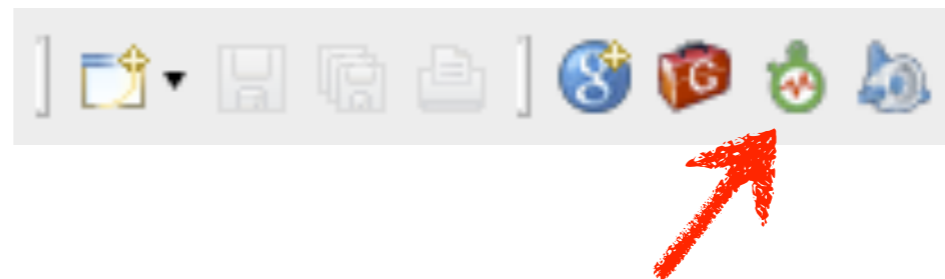
Details for Resource Data Received	
Description	Processing a file received by the resource loader.
@	17339ms
Duration	19.683ms
Processing Resource	6A50BDB1BFDE9EA651D4A1D3F2F71354.cache.html

Launch SpeedTracer

From Chrome...



Or from Eclipse...

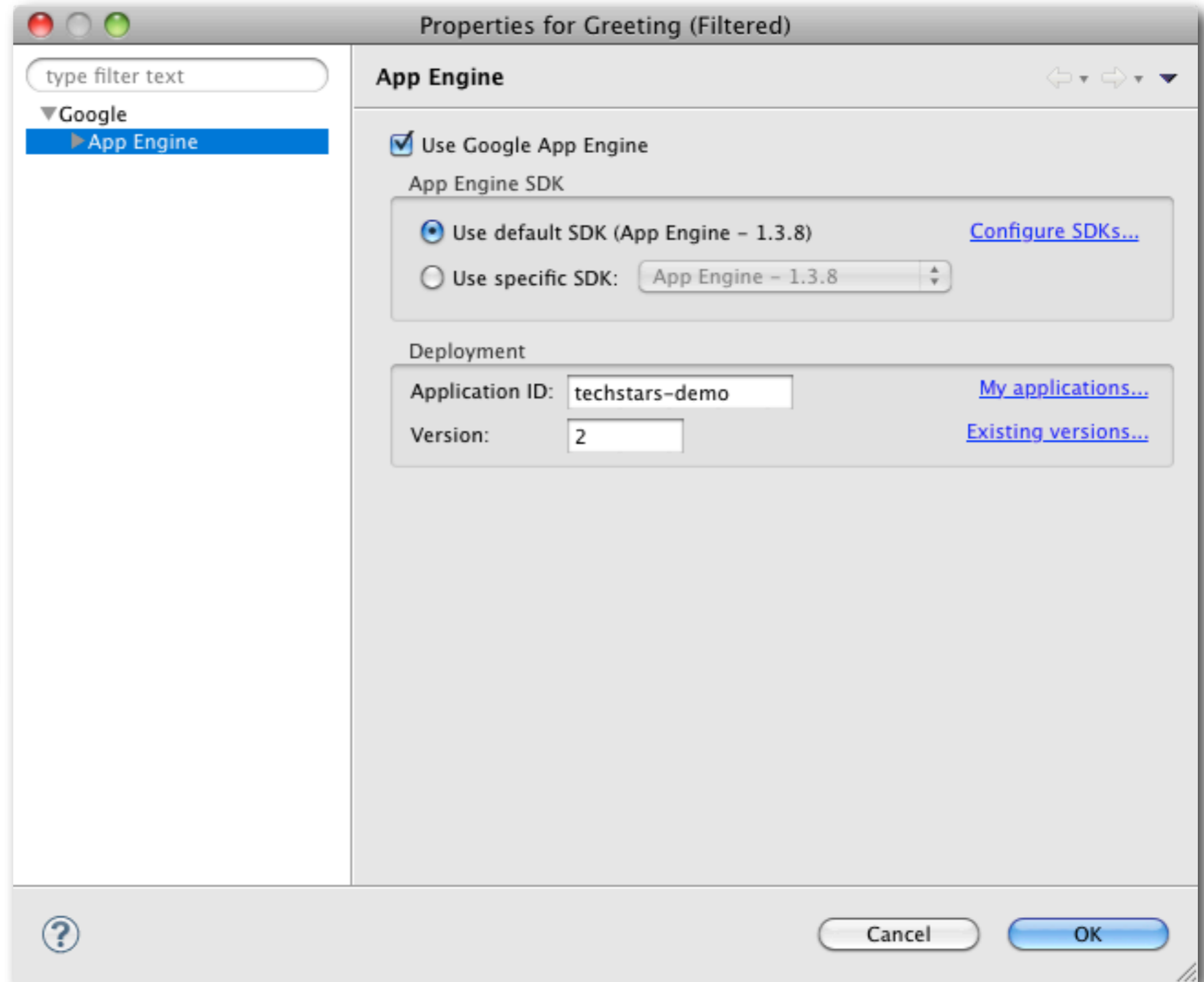
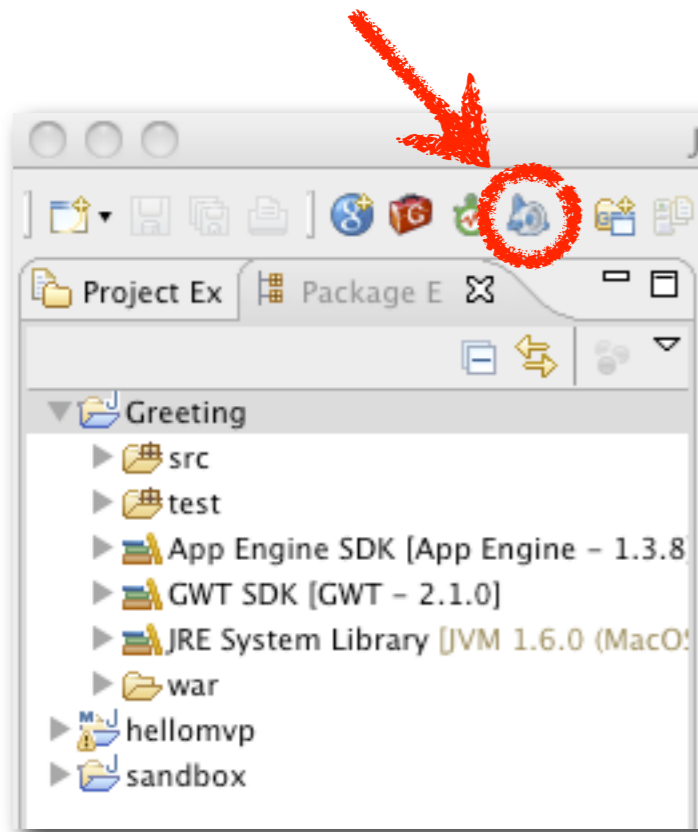


Server traces GAE, tcServer

The screenshot displays the Google Speed Tracer interface. At the top, the browser window shows the URL `http://appstats-test.lates` and a zoom level of `@17.06s for 4.00s` with a total time of `23.45s`. The main area features a timeline with 8 requests, showing sluggishness and network resources. A list of requests is visible on the left, including `clear.cache.gif` and `DailyROARReportAction`. The `Server Trace` section for the `POST /roa/dispatch/DailyROARReportAction` request shows a total duration of `42.0ms (self 10.0ms)` and a list of sub-operations:

- memcache.Get 4.0ms (self 4.0ms)
- memcache.Get 4.0ms (self 4.0ms)
- datastore_v3.RunQuery 13.0ms (self 13.0ms)
- memcache.Get 3.0ms (self 3.0ms)
- datastore_v3.Get 8.0ms (self 8.0ms)

One click deploy to GAE



Performance for your users



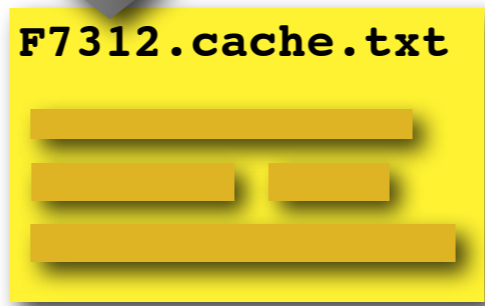
- Perfect Caching
- ClientBundle
- data: URLs & MHTML packaging
- Developer guided code splitting
- Compiler optimizations

Perfect caching

```
index.html
... ..
... .. <script src='foo.nocache.js' .. > ..
```

```
foo.nocache.js
..... .. .. A83B1C.cache.html .. ..
```

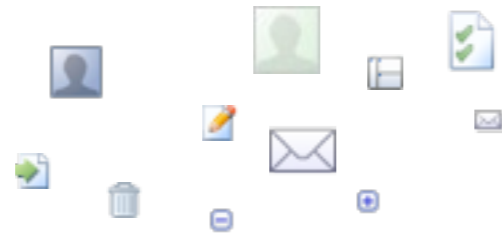
```
A83B1C.cache.html
... .. .. ..
..... .. .. ..
..... .. .. F7312.cache.txt ..
... 6C5BA.cache.png ..... ..
... ..
..... ..
```



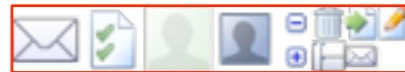
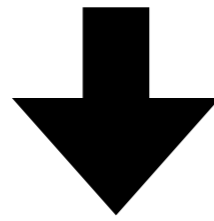
HTTP
If-Modified-Since

FOREVER
Cacheable

Image sprites for free

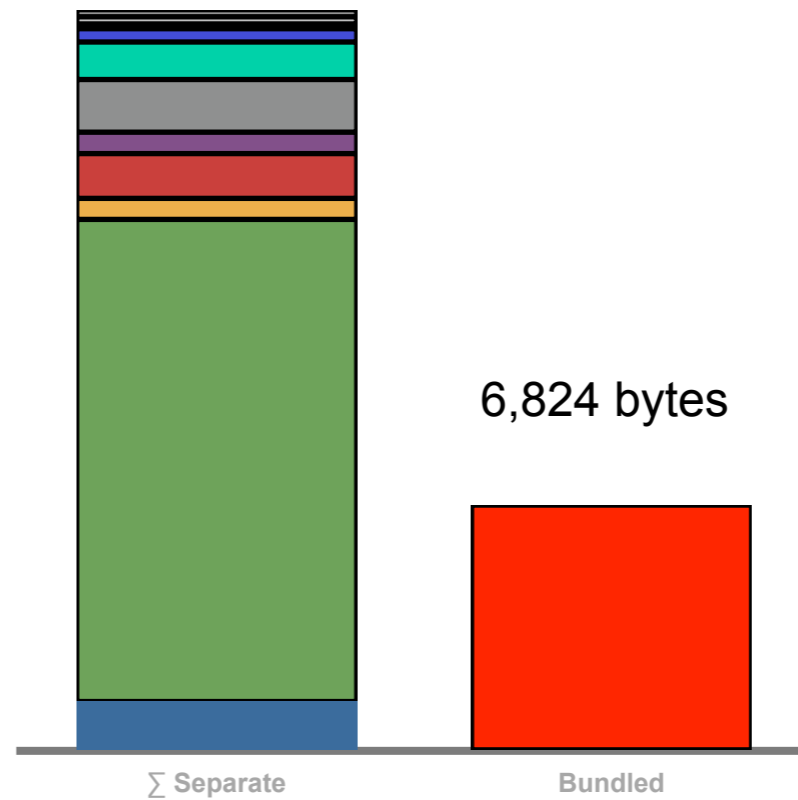


11 separate images



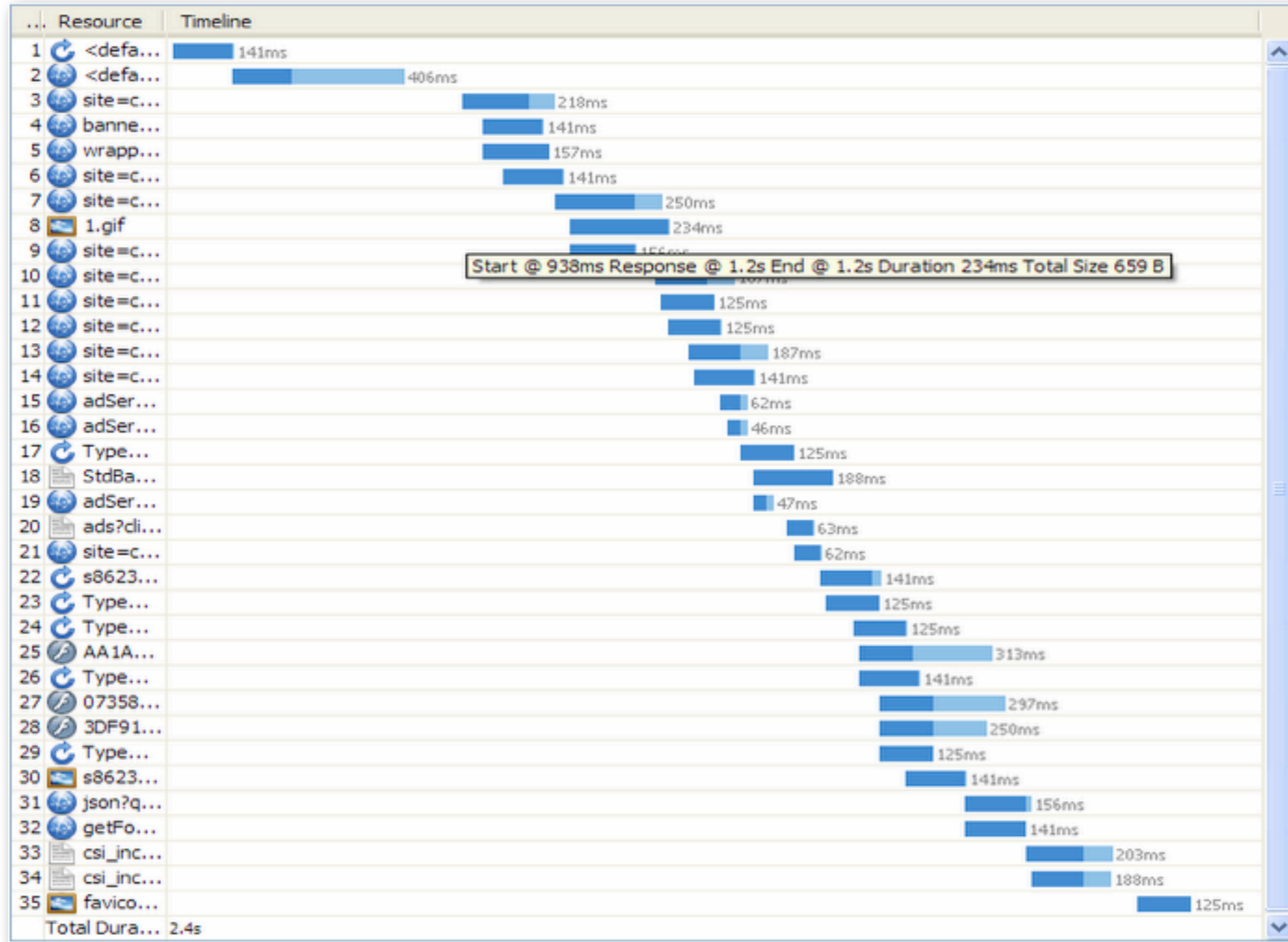
1 bundled image

20,558 bytes



6,824 bytes

Without ClientBundle



Deferred binding

	Firefox	Webkit (Safari)	Opera	IE
Typical portable setInnerText()	2876 ms	1276 ms	2053 ms	4078 ms
textContent=...	-	908 ms	1386 ms	-
innerText=...	2477 ms	918 ms	1520 ms	2469 ms
DOM manipulation	7148 ms	1997 ms	4836 ms	14800 ms

Improvement	14%	29%	32%	39%
-------------	-----	-----	-----	-----

GWT compiler optimization

“The fastest code is that which does not run.”

Joel Webber

GWT co-creator

GWT compiler optimization

Does all the stuff you'd do by hand in a tight JS app

- Shrinking (obfuscation)
- Modularization (pay only for what you use)
- Dead code removal
- Method inlining

...but automated and even tighter

Developer-guided code splitting (runAsync)

What's new in 2.2?

Integration with Spring / Roo

SafeHTML wrappers

GPE: Maven support, integration GWT Designer

Application framework

- RequestFactory, JSR 303 validation
- Activities and Places

Cell Widgets and Editor framework

GWT Canvas + Audio / Video

Four flavors of RPC

1. RequestBuilder + JSONParser (see RESTY-GWT)
2. RequestBuilder + XMLParser
3. GWT-RPC (easiest)
4. RequestFactory (new in 2.1)

GWT-RPC

```
@RemoteServiceRelativePath("greet")
public interface GreetingService extends RemoteService {
    String greetServer(String name) throws IllegalArgumentException;
}
```

```
public interface GreetingServiceAsync {
    void greetServer(String input, AsyncCallback<String> callback)
        throws IllegalArgumentException;
}
```

```
public class GreetingServiceImpl extends RemoteServiceServlet implements
    GreetingService
{
    public String greetServer(String input) throws IllegalArgumentException {
        ...
    }
}
```


Simple, powerful GWT-RPC

Send / receive Plain Old Java Objects (POJO)

- Easy: interface, async, & implementation
- Versioned to help keep client & server in sync
- Even smaller than JSON
- Supports polymorphism
- No JavaScript hijacking risk (JSON attack)
- Easy to find all calls to given service in IDE

GWT 2.1 RequestFactory

Newer alternative to GWT-RPC

Designed for data-oriented services

- Higher level of abstraction than GWT-RPC
- Foundation for future caching / batching

Even faster than GWT-RPC

- JSON-based == very fast (no serialization / deserialization required)
- Tracks changes on the client and sends **only diffs**

RequestFactory

The entity / DTO problem

EntityProxy / ValueProxy

Service stub interfaces extend RequestContext

AppRequestFactory extends RequestFactory

GWT.create(MyAppRequestFactory.class)

EntityProxy

```
@Entity
public class ItemList extends DatastoreObject
{
    private String name;
    private Key<AppUser> owner;
    private ListType listType;
    @Embedded
    private List<ListItem> items; // value type
    ...
}
```

```
@ProxyFor(ItemList.class)
public interface ItemListProxy extends DatastoreObjectProxy
{
    // TODO enums work!
    public enum ListType {NOTES, TODO}
    String getName();
    void setName(String name);
    List<ListItemProxy> getItems();
    AppUserProxy getOwner(); // NOT Key
    ...
}
```

DatastoreObject

```
package com.listwidget.domain;
public class DatastoreObject
{
    @Id
    private Long id;
    private Integer version = 0;

    @PrePersist
    void onPersist()
    {
        this.version++;
    }
    ...
}
```

```
package com.listwidget.shared.proxy;
public interface DatastoreObjectProxy extends EntityProxy
{
    Long getId();
    Integer getVersion();
}
```

ValueProxy

```
public class ListItem // POJO
{
    private String itemText;
    private Date dateCreated;

    public Date getDateCreated()
    {
        return dateCreated;
    }
}
```

```
@ProxyFor(value = ListItem.class)
public interface ListItemProxy extends ValueProxy
{
    String getItemText();
    void setItemText(String itemText);
    Date getDateCreated();
}
```

Making a RequestFactory

```
public interface ListwidgetRequestFactory extends RequestFactory
{
    @Service(value = ItemListDao.class, locator = DaoServiceLocator.class)
    interface ItemListRequestContext extends RequestContext
    {
        Request<List<ItemListProxy>> listAll();
        Request<Void> save(ItemListProxy list);
        Request<ItemListProxy> saveAndReturn(ItemListProxy newList);
    }
    ItemListRequestContext itemListRequest();
}
```

```
private final ListwidgetRequestFactory rf =
    GWT.create(ListwidgetRequestFactory.class);
```

Using RequestFactory

```
@Override
public void persistList(String listName)
{
    final ListwidgetRequestFactory rf = this.clientFactory
        .getRequestFactory();
    ItemListRequestContext reqCtx = rf.itemListRequest();
    final ItemListProxy newList = reqCtx.create(ItemListProxy.class);
    newList.setName(listName);
    newList.setListType(ListType.TODO);
    reqCtx.saveAndReturn(newList).fire(new Receiver<ItemListProxy>()
    {
        @Override
        public void onSuccess(final ItemListProxy savedList)
        {
            // Refresh table
            listDataProvider.getData();
        }
    });
}
```


Using RequestFactory

```
@Override
public void update(int index, ItemListProxy obj, final String newName)
{
    ItemListRequestContext reqCtx = clientFactory.getRequestFactory()
        .itemListRequest();
    ItemListProxy editable = reqCtx.edit(obj);
    editable.setName(newName);
    reqCtx.save(editable).fire(new Receiver<Void>()
    {
        @Override
        public void onSuccess(Void response)
        {
            EventBus.fireEvent(new MessageEvent(newName + " updated",
                MessageType.INFO));
        }
    });
}
```

Using RequestFactory

```
private void getData()
{
    // To retrieve relations and value types, use .with()
    Request<List<ItemListProxy>> findAllReq = rf.itemListRequest()
        .listAll().with("owner");
    // Receiver specifies return type
    findAllReq.fire(new Receiver<List<ItemListProxy>>()
    {
        @Override
        public void onSuccess(List<ItemListProxy> response)
        {
            updateRowData(0, response);
        }
    });
}
```

Using RequestFactory

```
editList = reqCtx.edit(editList);
List<ListItemProxy> items = editList.getItems();
// must initialize collections
if (items == null)
{
    editList.setItems(new ArrayList<ListItemProxy>());
}
editList.getItems().add(newItem);
reqCtx.save(editList).with("items").to(new Receiver<Void>())
{
    @Override
    public void onSuccess(Void response)
    {
        itemsProvider.setList(editList.getItems());
        itemsProvider.refresh();
    }
}).fire();
```

GWT MVP - Concepts

View

- Interface + implementation
- Interface enables testing without GWTTestCase
- Typically expensive to construct so make reusable

Presenter

- No Widgets, just business logic
- Middle man between service layer and views

GWT MVP - Concepts

Activity

- Similar to presenters, but higher level
- “wake up, set up, show up”
- Automatically warn users before leaving
- Instantiate view (or obtain from factory)
- Mapped from Places

GWT MVP - Concepts

Place

- Place represents bookmarkable state of an activity
- PlaceController makes back button / bookmarks work like users expect
- PlaceTokenizers map to / from String tokens on URL

Activity

```
public class EditListActivity extends AbstractActivity implements Presenter
{
    private ClientFactory clientFactory;
    private EventBus eventBus;

    public EditListActivity(ClientFactory cf, EditListPlace editListPlace)
    {
        this.clientFactory = cf;
        this.itemListToken = editListPlace.getToken();
    }

    @Override
    public void start(final AcceptsOneWidget panel, EventBus eventBus)
    {
        this.eventBus = eventBus;
        ...
        panel.setWidget(clientFactory.getEditListView());
    }
    ...
}
```

Place

```
public class EditListPlace extends Place
{
    private String token;
    public EditListPlace(String token)
    {
        this.token = token;
    }
    public String getToken()
    {
        return token;
    }
    public static class Tokenizer implements PlaceTokenizer<EditListPlace>
    {
        public EditListPlace getPlace(String token)
        {
            return new EditListPlace(token);
        }
        public String getToken(EditListPlace place)
        {
            return place.getToken();
        }
    }
}
```


ActivityMapper

```
public class AppActivityMapper implements ActivityMapper {  
  
    private ClientFactory clientFactory;  
  
    public AppActivityMapper(ClientFactory clientFactory) {  
        super();  
        this.clientFactory = clientFactory;  
    }  
  
    @Override  
    public Activity getActivity(Place place) {  
        if (place instanceof EditListPlace) {  
            return new EditListActivity(clientFactory, (EditListPlace) place);  
        }  
        if (place instanceof ListsPlace)  
        {  
            return new ListsActivity(clientFactory);  
        }  
        return null;  
    }  
}
```

PlaceHistoryMapper

```
/**
 * PlaceHistoryMapper interface is used to attach all places which the
 * PlaceHistoryHandler should be aware of. This is done via the @WithTokenizers
 * annotation or by extending PlaceHistoryMapperWithFactory and creating a
 * separate TokenizerFactory.
 */
@WithTokenizers({ ListsPlace.Tokenizer.class, EditListPlace.Tokenizer.class })
public interface AppPlaceHistoryMapper extends PlaceHistoryMapper
{
}
```

EventBus

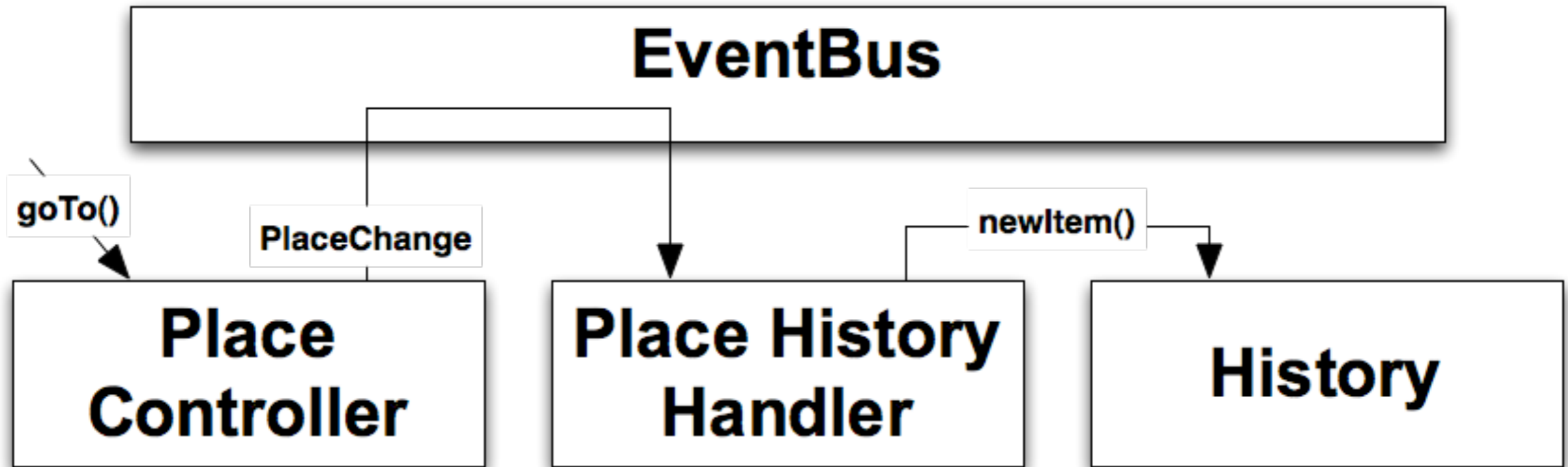
**Place
Controller**

**Place History
Handler**

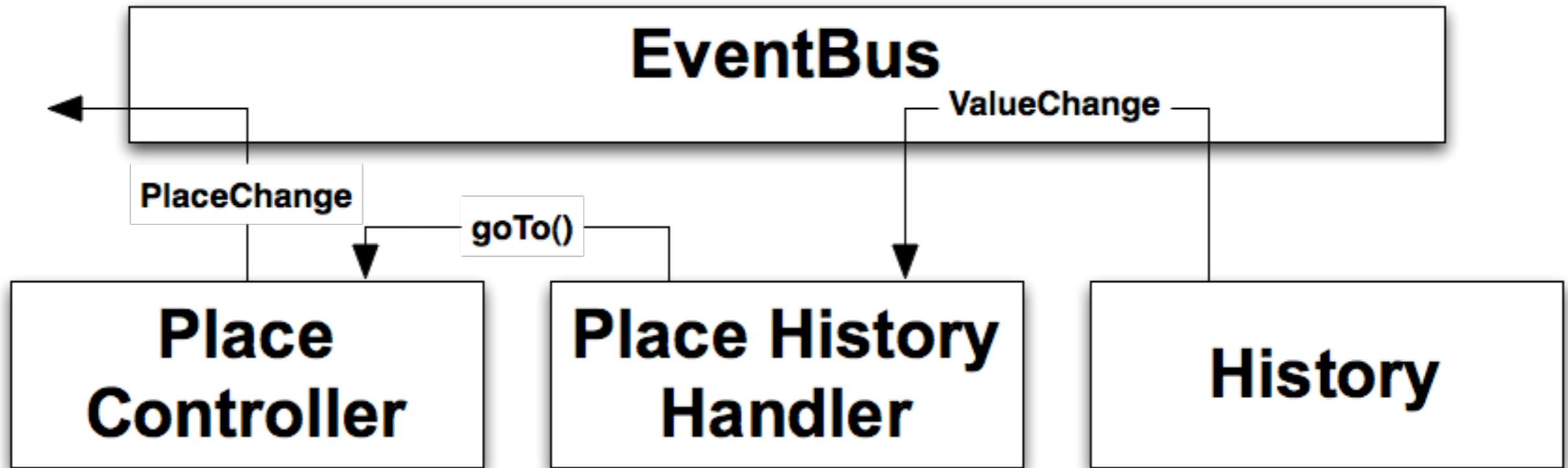
History

Your config here:
list of all place types

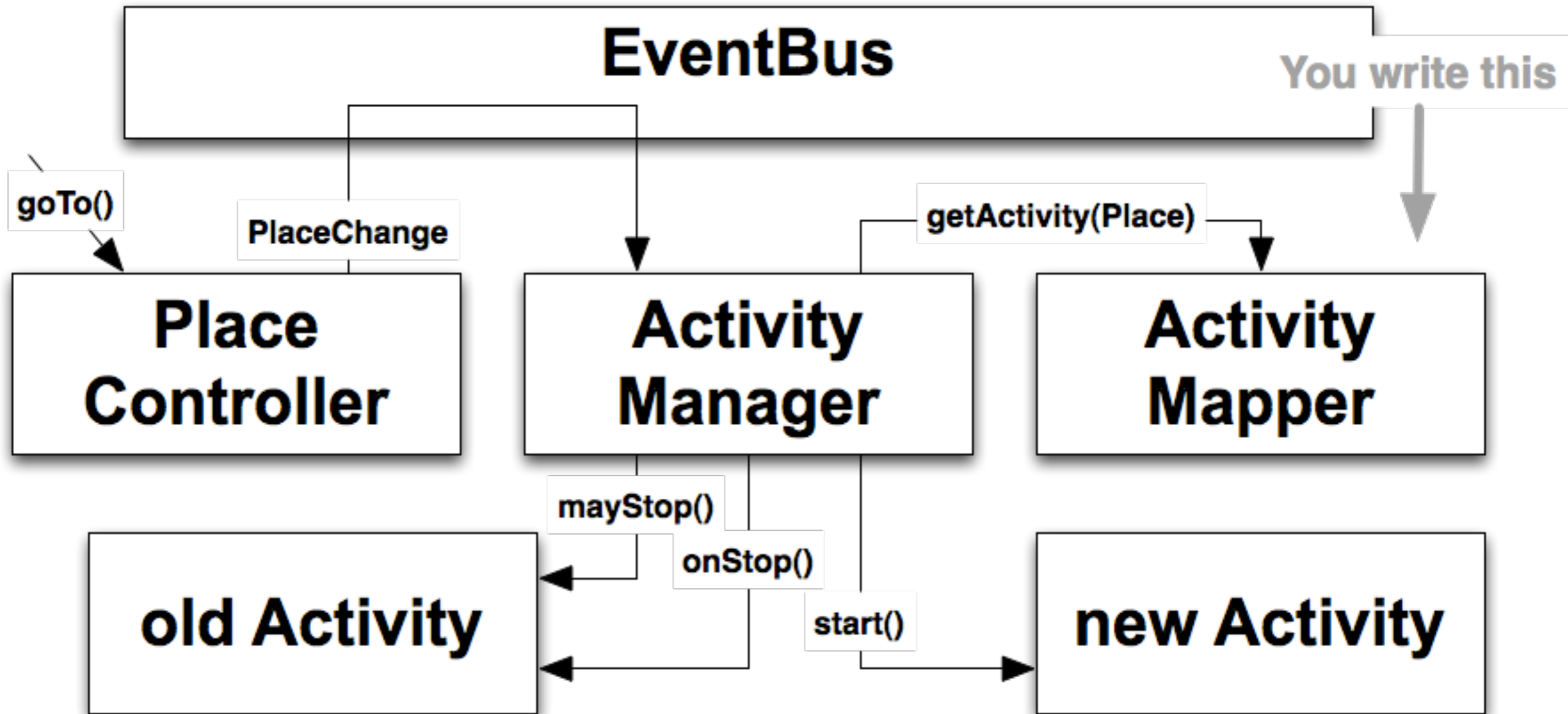
Places: Moving parts



Places: Go to



Places: Back and forth



Get activated

Widgets

Doo Dads

Doo Dad Able

Baker Doo Dad

Thingies

Gizmos

[Home](#) > [Doo Dads](#) > Doo Dad Able

Name	Doo	Dad	Ding	Dong
Doo Dad Able	34	The Goods	Peldi	<input checked="" type="checkbox"/>
Baker Doo Dad 4	18	The Guids	Pongi	<input type="checkbox"/>

Doo Dad Deets

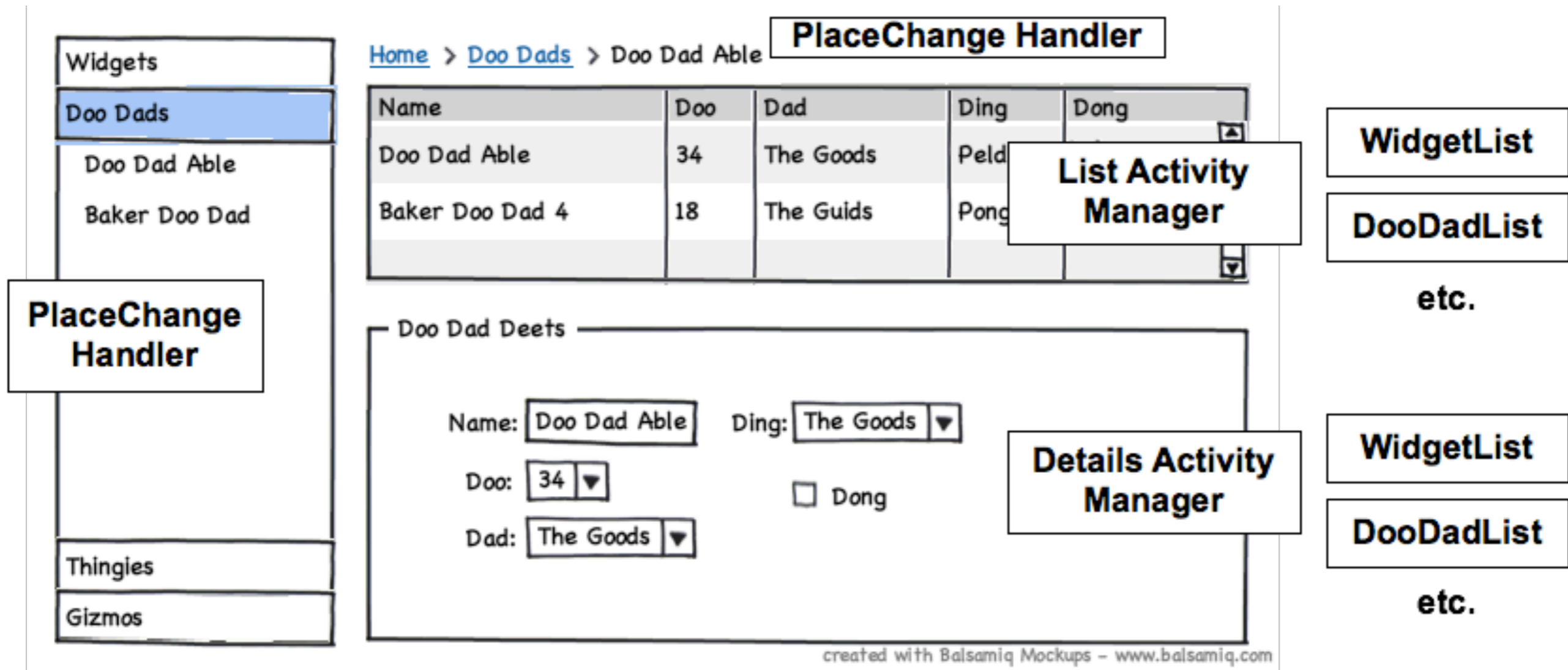
Name: Ding: ▼

Doo: ▼ Dong

Dad: ▼

created with Balsamiq Mockups - www.balsamiq.com

Strategies



Strategies

Activities and Places wiring

```
// Start ActivityManager for the nav (west) panel with our WestActivityMapper
ActivityMapper westActivityMapper = new WestActivityMapper(clientFactory);
ActivityManager westActivityManager =
    new ActivityManager(westActivityMapper, eventBus);
westActivityManager.setDisplay(westPanel);

// Start ActivityManager for the main (center) panel with our CenterActivityMapper
ActivityMapper centerActivityMapper = new CenterActivityMapper(clientFactory);
ActivityManager centerActivityManager =
    new ActivityManager(centerActivityMapper, eventBus);
centerActivityManager.setDisplay(centerPanel);

// Start PlaceHistoryHandler with our PlaceHistoryMapper
PlaceHistoryMapper historyMapper = GWT.create(AppPlaceHistoryMapper.class);
PlaceHistoryHandler historyHandler = new PlaceHistoryHandler(historyMapper);
historyHandler.register(placeController, eventBus, defaultPlace);

RootPanel.get().add(dockLayoutPanel);
// Goes to place represented on URL or default place
historyHandler.handleCurrentHistory();
```

Cell Widgets

Fast, lightweight table rendering

Data binding with
DataProviders, ValueUpdaters

Scrolling and paging controls

Sortable columns, adjustable width

TextBox, IntegerBox, ValueBox<T>

Cell Widgets

```
// Create a CellTable.
CellTable<Contact> table = new CellTable<Contact>();

// Create name column.
@Override
TextColumn<Contact> nameColumn = new TextColumn<Contact>() {
    public String getValue(Contact contact) {
        return contact.name;
    }
};

// Create address column.
@Override
TextColumn<Contact> addressColumn = new TextColumn<Contact>() {
    public String getValue(Contact contact) {
        return contact.address;
    }
};

// Add the columns.
table.addColumn(nameColumn, "Name");
table.addColumn(addressColumn, "Address");
```

Editable Column

```
// Editable column for list name
nameColumn = new Column<ItemListProxy,String>(new EditTextCell())
{
    @Override
    public String getValue(ItemListProxy list)
    {
        return list.getName();
    }
};
```

Custom Column type

```
// Note Flyweight pattern: only one instance of HyperlinkCell passed to the Column
Column<ItemListProxy, Hyperlink> linkColumn =
    new Column<ItemListProxy, Hyperlink>(new HyperlinkCell())
{
    @Override
    public Hyperlink getValue(ItemListProxy list)
    {
        String proxyToken =
            clientFactory.getRequestFactory().getHistoryToken(list.stableId());
        String historyToken =
            clientFactory.getHistoryMapper().getToken(new EditListPlace(proxyToken));
        Hyperlink h = new Hyperlink(list.getName(), historyToken);
        return h;
    }
};
```

Populating a CellTable

```
public static class MyDataProvider extends
    AsyncDataProvider<ItemListProxy>
{
    @Override
    protected void onRangeChanged(HasData<ItemListProxy> display)
    {
        // To retrieve relations and value types, use .with()
        Request<List<ItemListProxy>> findAllReq = rf.itemListRequest()
            .listAll().with("owner");
        // Receiver specifies return type
        findAllReq.fire(new Receiver<List<ItemListProxy>>()
        {
            @Override
            public void onSuccess(List<ItemListProxy> response)
            {
                updateRowData(0, response);
            }
        });
    }
}
```

```
this.myDataProvider = new MyDataProvider(requestFactory);
```

Updating with a CellTable

```
public class NameFieldUpdater implements FieldUpdater<ItemListProxy, String>
{
    @Override
    public void update(int index, ItemListProxy obj, final String newName)
    {
        ItemListRequestContext reqCtx = clientFactory.getRequestFactory()
            .itemListRequest();
        ItemListProxy editable = reqCtx.edit(obj);
        editable.setName(newName);
        reqCtx.save(editable).fire(new Receiver<Void>()
        {
            @Override
            public void onSuccess(Void response)
            {
                EventBus.fireEvent(new MessageEvent(newName + " updated",
                    MessageType.INFO));
            }
        });
    }
};
```

```
// Make list name field editable
display.getNameColumn().setFieldUpdater(new NameFieldUpdater());
```

GWT Logging

GWT now has `java.util.Logging` emulation

```
# In your .gwt.xml file
<inherits name="com.google.gwt.logging.Logging" />

# In your .java file
Logger logger = Logger.getLogger("NameOfYourLogger");
logger.log(Level.SEVERE, "this message should get logged");
```

Configure in your `.gwt.xml`

```
<set-property name="gwt.logging.logLevel" value="SEVERE" />
<set-property name="gwt.logging.enabled" value="FALSE" />
<set-property name="gwt.logging.consoleHandler" value="DISABLED" />
```

Easily enable remote logging, too

```
<set-property name="gwt.logging.simpleRemoteHandler" value="ENABLED" />
```


GWT secrets

-compileReport

Google web toolkit

Application breakdown analysis

Package breakdown

Package	Size (Bytes)
java.util	11216 (21.24%)
com.google.gwt.user.client.ui	9183 (17.39%)
com.google.gwt.layout.client	4814 (9.12%)
com.google.gwt.dom.client	4038 (7.65%)
com.google.gwt.user.client.impl	3512 (6.65%)
com.google.gwt.event.shared	2867 (5.43%)
com.google.gwt.core.client.impl	2579 (4.88%)
java.lang	2159 (4.09%)
com.google.gwt.place.shared	1828 (3.46%)
com.google.gwt.lang	1767 (3.35%)
com.google.gwt.user.client	1531 (2.9%)
com.google.gwt.activity.shared	1413 (2.68%)
com.google.gwt.sample.hellomvp.client.ui	1130 (2.14%)
com.google.gwt.animation.client	843 (1.6%)
com.google.gwt.event.dom.client	676 (1.28%)
com.google.gwt.event.logical.shared	613 (1.16%)
com.google.gwt.sample.hellomvp.client.mvp	548 (1.04%)
com.google.gwt.i18n.client	463 (0.88%)
com.google.gwt.sample.hellomvp.client.activity	460 (0.87%)
com.google.gwt.uibinder.client	360 (0.68%)
com.google.gwt.core.client	331 (0.63%)
com.google.gwt.sample.hellomvp.client.place	301 (0.57%)

GWT secrets: faster compile

- `-draftCompile` skips optimizations
- Set only one user-agent in `gwt.xml` for dev
- Careful with RPC polymorphism

GWT secrets: shrink JS

- `-XdisableClassMetadata`: Disables some `java.lang.Class` methods (e.g. `getName()`)
- `<set-property name="compiler.stackMode" value="strip"/>`
Removes client-side stack trace info (can reduce code size up to 15%)
- `<set-configuration-property name="CssResource.obfuscationPrefix" value="empty"/>`

GWT secrets: shrink JS

- enum obfuscation
- `-XdisableCastChecking`: Disables run-time checking of cast operations (use with caution)
- See GWT FAQ, `CompilerParameters.gwt.xml`

GWT speak

“Very productive environment for Java developers, but there is a learning curve”

“UI layout and styling is a challenge”

“GWT has enabled our team to run much faster than we could otherwise”

“Would be impossible to write our app without GWT”

GWT books

<http://code.google.com/webtoolkit/books.html>



Thank you!

Where to get it:

<http://code.google.com/webtoolkit/>

Wiki, issue tracker, source:

<http://code.google.com/p/google-web-toolkit/>

Official GWT blog:

<http://google-web-toolkit.blogspot.com/>

Twitter:

[@googledevtools](https://twitter.com/googledevtools)

