

# PRAIRIE DEV CON 2011

PRAIRIE DEVELOPER CONFERENCE

## Microsoft®

ADD-IN EXPRESS™

 **imagnet**

 **paradigm**  
CONNECTED. GROW. LIVE.

 **SOLVERA**  
SAVING RESULTS

 **Online**  
BUSINESS SYSTEMS

 **telerik**

*deliver more than expected*  
(Track Sponsor)

## Google Web Toolkit and the Model View Presenter architecture

Philippe  
Beaudoin

# About me...

Lead developer at **BookedIN.net**

Founder of **arcbees.com** (consulting)

Creator of **gwtplatform.com**

(Starting at Google in two weeks)

philippe.beaudoin@gmail.com

@PhilBeaudoin

# Our tools

(you'll need...)

- Eclipse Helios (for Java Developer, or more)
- Google Plugin for Eclipse
  - <http://dl.google.com/eclipse/plugin/3.6>
  - Take the SDKs too
- GWT Plugin for Eclipse
  - <http://plugin.gwt-platform.googlecode.com/hg/update>

# Our stack

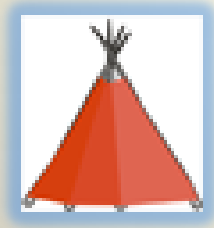
- Java
- HTML / CSS
- GWT
- GWT-Platform (GWTP aka gooteepee)
- GIN (Guice for GWT)
- (AppEngine)



# GWT

## Google Web Toolkit

- Build complex browser-based apps in Java
  - Java → javascript cross compiler
  - All the goodies of Java
    - Rock-solid IDEs and tools
    - Refactoring
    - Unit testing
- Deferred binding
  - Serves a *different* app to different users
    - Chrome + Français? Check!
    - Tablet + Ancient greek? Coming right up!



# GWT-Platform

- Greatly simplifies writing MVP code
- Includes a command pattern
  - Action-based client/server exchanges
- Best thing since sliced bread!
  - Trust the author! ;)

# GIN

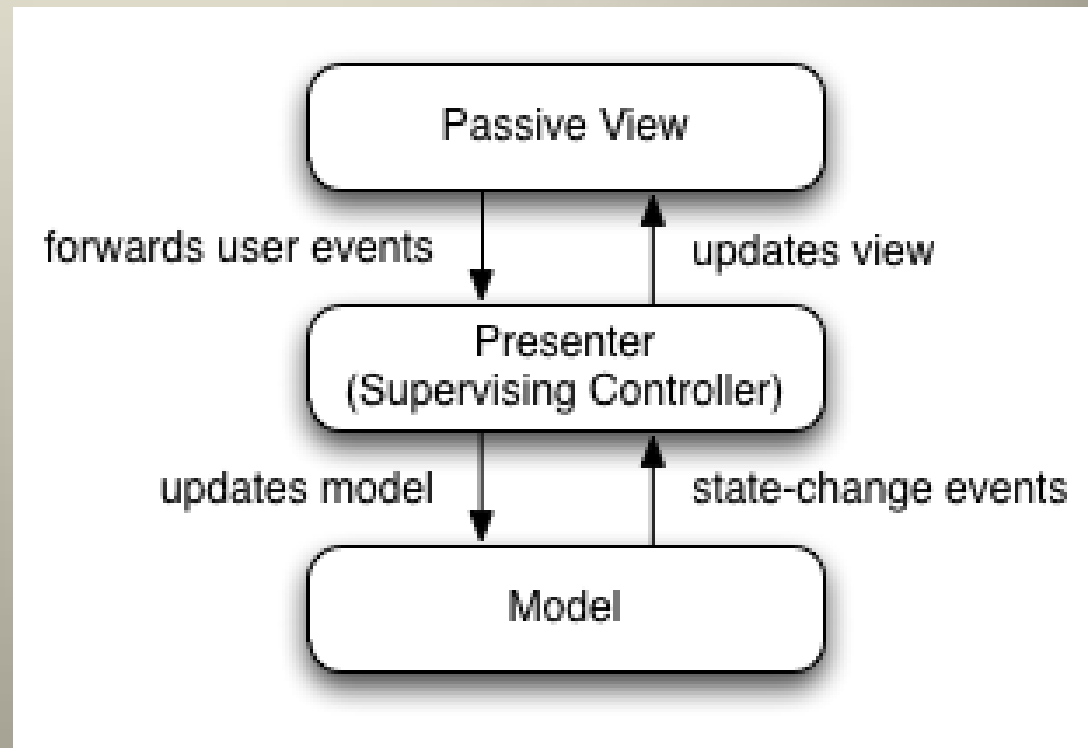
## Guice for GWT

- As-simple-as-possible dependency injection
- Ginjector
  - The “master factory”
- Module
  - Java code to bind interfaces to implementations
- Then: use `@Inject` on your constructors!

# Our architecture

## MVP: Model View Presenter

- Model (entity graph)
- View
  - Dumb, no logic
- Presenter
  - Logic, no UI code
- Advantages
  - Testability
  - Low coupling





# Root vs RootLayout

- Web Page vs Web app
- Browser scrollbar vs Your own scrollbars
- Infinite Page vs Full screen

# Presenter lifecycle (1)

- **onBind()** : right after construction
  - Add handlers to the view, see registerHandler()
  - One time setup (NOT in the constructor)
- **onUnbind()** : when released
  - Has to be called manually (fot now)
  - Undo any setup operations done in onBind()
  - Except for registerHandler()

# Presenter lifecycle (2)

- **onReveal()** : from invisible to visible
- **onHide()** : from visible to invisible
- **onReset()** : when navigation occurs and the presenter is still visible after
  - NOT the same as **onReveal()** as soon as you have a presenter hierarchy
  - The right place for most “update” operations

# Command pattern

- Operations are wrapped in classes
- Send the class to the server with a dispatcher
- Results are also wrapped in classes
- Idea: keep them in a stack for undo/redo!

# URL parameters

- Saving part of the presenter state in the URL
  - Good for bookmarks
  - Or for navigation (back, forward)
- Ex: **`http://phone.com#!search;q=iphone`**
- Override **`prepareFromRequest()`**

# Nested Presenters

Presenter 1  
*not a place*

Presenter 3  
*a place*

Presenter 2  
*not a place*

Logged in as **PhilBeaudoin** | [logout](#)

Blog

***Our services***

Contact Us

## Services

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exercitation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum. Typi non habent claritatem insitam; est usus legentis in iis qui facit eorum claritatem. Investigationes demonstraverunt lectores legere me lius quod ii legunt saepius. Claritas est etiam processus dynamicus, qui sequitur mutationem consuetudium lectorum. Mirum est notare quam littera gothica, quam nunc putamus parum claram, anteposuerit litterarum formas humanitatis per seacula quarta decima et quinta decima. Eodem modo typi, qui nunc nobis videntur parum clari, fiant sollemnes in futurum.

# Event bus

- Keep these presenters decoupled
- Use the EventBus to communicate
- Example:
  - Presenter3 tells Presenter2 which tab to highlight
    - Better: Use GWTP tabbed presenters!
  - System broadcasts that a user has logged in

# Other features of GWTP (1)

- PresenterWidgets
  - Great for reusable components
  - Light-weight way to decouple UI from Logic
- Hierarchical ajax URLs
  - **#!product-list;type=phones/!product;id=323**
  - Ideal for breadcrumbs
- Gatekeepers
  - Dynamically block out presenters



# Other features of GWTP (2)

- Dispatcher goodies
  - Action caching/batching, client-side handling
  - ActionValidator (server-side security)
- Search engine crawler support
  - Make these “hash bang” URLs crawlable
- Google Analytics support
  - Track navigation at the presenter level
- Helpers for internationalization



# Testing?

Check out Jukito, the best (only?) automocking testing framework for Gin/Guice.

Also: the second best thing since sliced bread.

Also: I created it... ;)

<http://jukito.org>

# Need help?



[arcbees.com](http://arcbees.com)

GWT

AppEngine

# Questions?

philippe.beaudoin@gmail.com

@PhilBeaudoin

<http://arcbees.com>

**Slides, GWTP and doc at:**

<http://gwtpplatform.com>

**Code at:**

<http://code.google.com/p/gwtp-demo/>