

веб-разработка на GWT и mvp4g



Котенко Антон, iPark ventures™
2011 ©

Презентация подготовлена для конференции
Application Developer Days 2011
Санкт-Петербург, Россия



Application Developer Days

Котенко Антон

6-летний опыт разработки на **Java EE**

6-летний опыт разработки **UI**

Пол: мужской

Котенко Антон

6-летний опыт разработки на **Java EE**

6-летний опыт разработки **UI**

Пол: мужской



<http://shamansir.madfire.net>



<http://zokotuhaFly.habrahabr.ru>



http://twitter.com/shaman_sir



<http://profiles.google.com/shaman.sir>

План

1. Введение. Краткая история и
примеры использования GWT

2. Краткое описание концепций:

- MVP / Reverse MVP
- EventBus
- Dependency Injection

3. Описание фреймворка mvp4g

- отличие от реализации (R)MVP в GWT
- отличие в реализации EventBus
- многомодульность
- понятие HistoryConverter
- достоинства / недостатки

4. Компоненты в GWT

- UiBinder, стандартные компоненты
- разработка кастомных виджетов

5. Наша разработка Layouting-системы

6. Работа с не-Java Server-Side-API

(в противовес RPC-сервисам)

- создание цепочек вызовов
- callback vs. GwtEvent
- достоинства / недостатки

7. i18n в GWT

8. Заключение. Ссылки на примеры

Вопросы можно (и **нужно**) задавать в течение доклада:
для извещения докладчика о наличии у вас вопроса по теме или нет
поднимите руку (любую).

Вот так:



Дискуссия — это важно.

1. Кратко о GWT

GWT

Google
Web
Toolkit



/'gwɪt/

code.google.com/webtoolkit/

GWT

/'gwɪt/

/ˈɡwɪt/

Используется в проектах:

Google Wave wave.google.com

Google Checkout checkout.google.com

Google Moderator google.com/moderator

Whirled whirled.com

Lombardi Blueprint blueprint.lombardi.com

ContactOffice beta.contactoffice.com

Используется в проектах:

GoGrid gogrid.com

Curriki curriki.org

OpenKM openkm.com

Kdice kdice.com

SeeMap seemap.ru

Одноклассники odnoklassniki.ru

Набор полезных инструментов

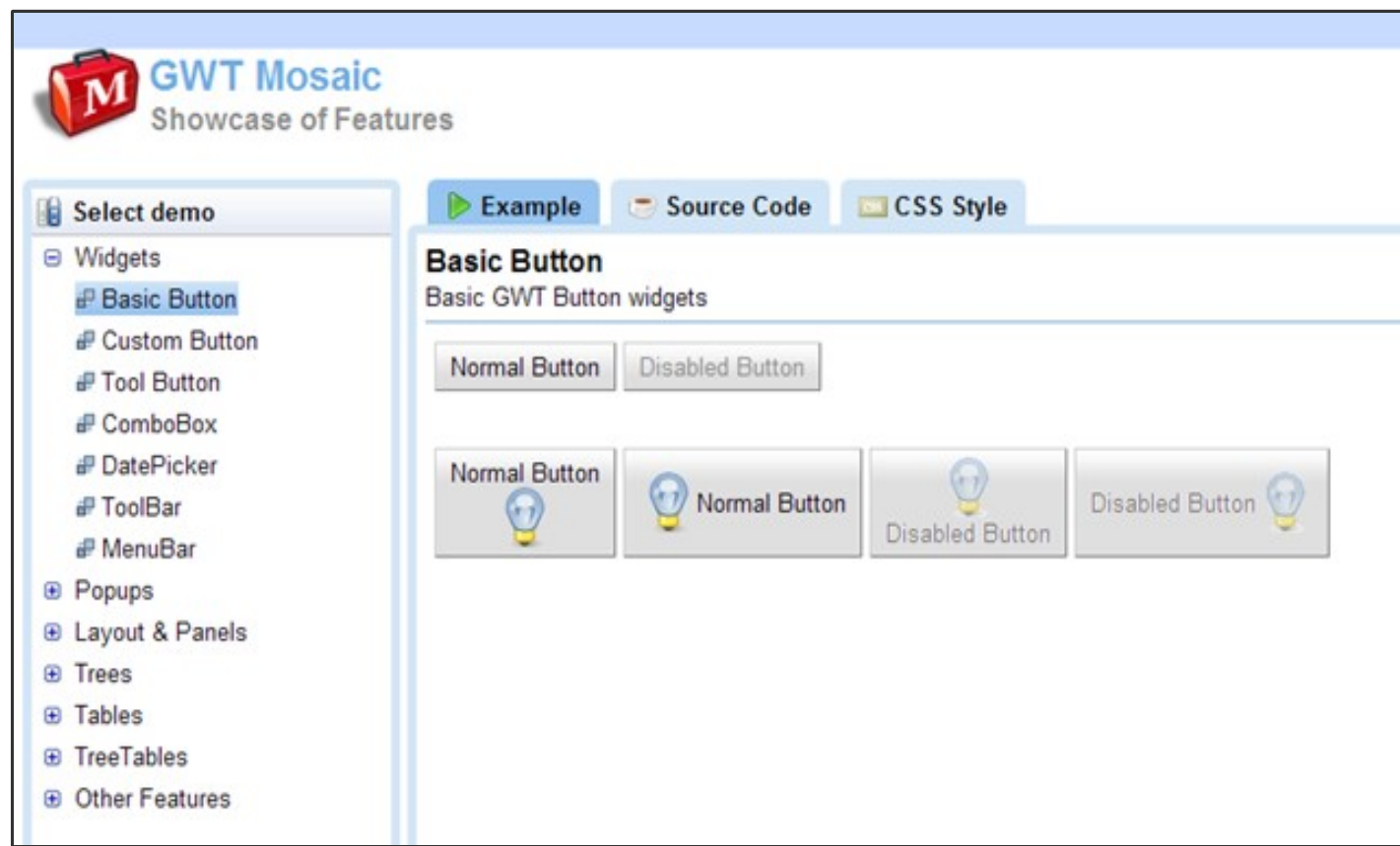


Наиболее полный Набор полезных инструментов



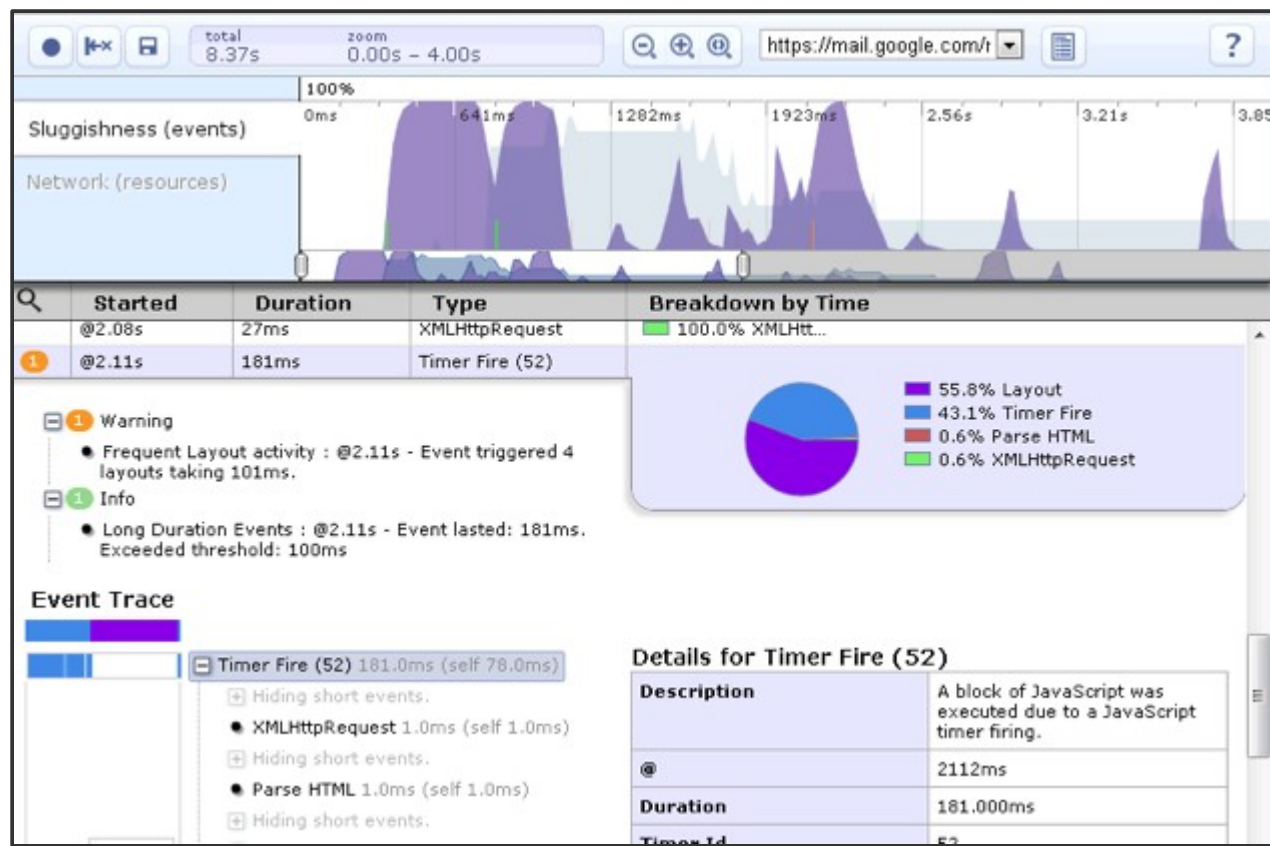
веб-разработчика

ВИДЖЕТЫ



ВИДЖЕТЫ

ОПТИМИЗАЦИЯ



ВИДЖЕТЫ

ОПТИМИЗАЦИЯ

кросс-браузерно!

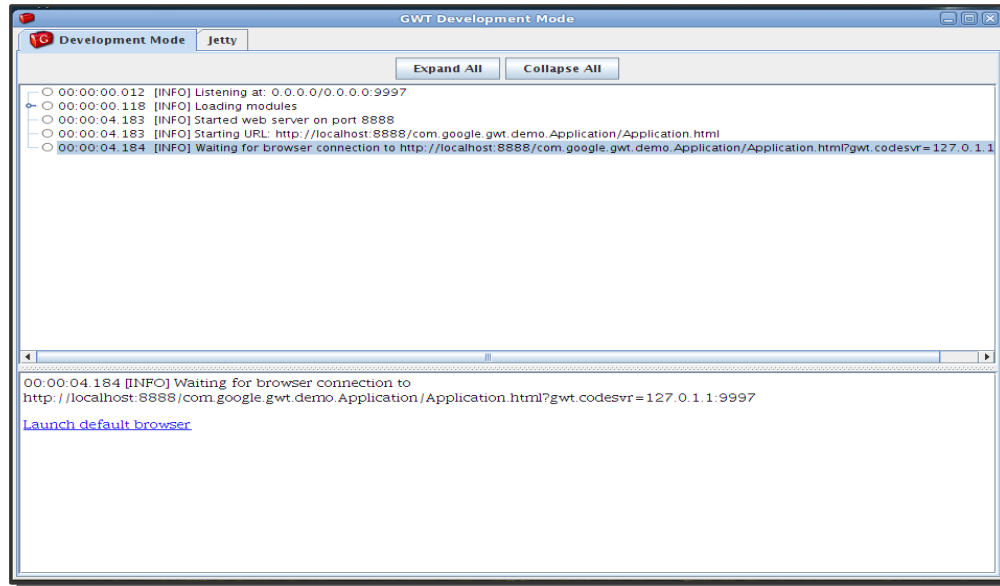


ВИДЖЕТЫ

ОПТИМИЗАЦИЯ

КРОСС-БРАУЗЕРНО!

РАЗРАБОТКА on-the-fly



ВИДЖЕТЫ

ОПТИМИЗАЦИЯ

кросс-браузерно!

разработка on-the-fly

преимущества ООП



ВИДЖЕТЫ

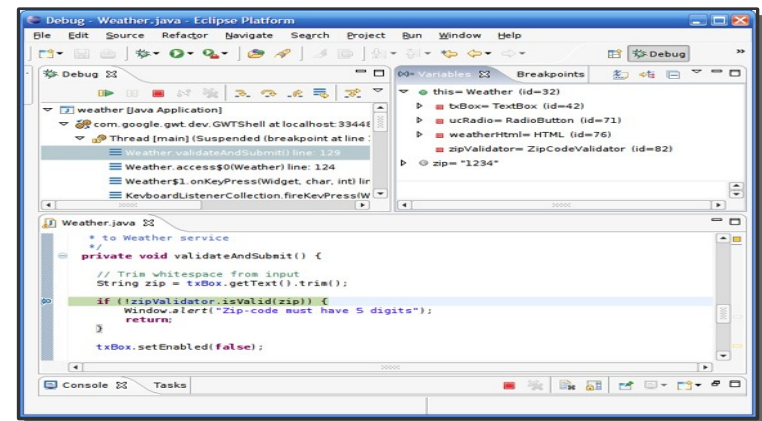
ОПТИМИЗАЦИЯ

КРОСС-БРАУЗЕРНО!

РАЗРАБОТКА ON-THE-FLY

ПРЕИМУЩЕСТВА ООП

debug



ВИДЖЕТЫ

ОПТИМИЗАЦИЯ

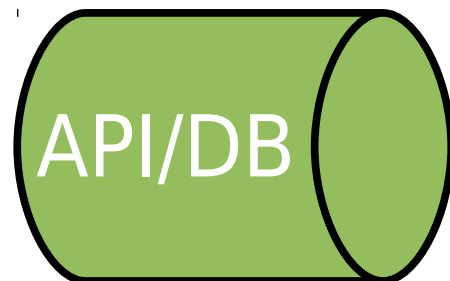
кросс-браузерно!

разработка on-the-fly

преимущества ООП

debug

RPC



ВИДЖЕТЫ

ОПТИМИЗАЦИЯ

кросс-браузерно!

разработка on-the-fly

преимущества ООП

debug

RPC



простой i18n

ВИДЖЕТЫ

ОПТИМИЗАЦИЯ

кросс-браузерно!

разработка on-the-fly

преимущества ООП

debug

RPC

простой i18n

Code Splitting

ВИДЖЕТЫ

ОПТИМИЗАЦИЯ

кросс-браузерно!

разработка on-the-fly

преимущества ООП

debug

RPC

простой i18n

Code Splitting





/ˈɡwɪt/
GWT — набор полезных инструментов



История

Версия 1.0
вышла в мае 2006-го года

Версия 1.0
вышла в мае 2006-го года

Версия 1.6
Google Eclipse Plugin
структура проекта соответствует
спецификации Web Application

Версия 2.0

- Development Mode
- Code Splitting
- Декларативный UI
- Client Bundle

Версия 2.1

MVP-концепция

RequestFactory / Editors

Версия 2.1

MVP-концепция

RequestFactory / Editors

Версия 2.2

UI Designer

поддержка HTML5 Canvas,

остаётся только Java 1.6

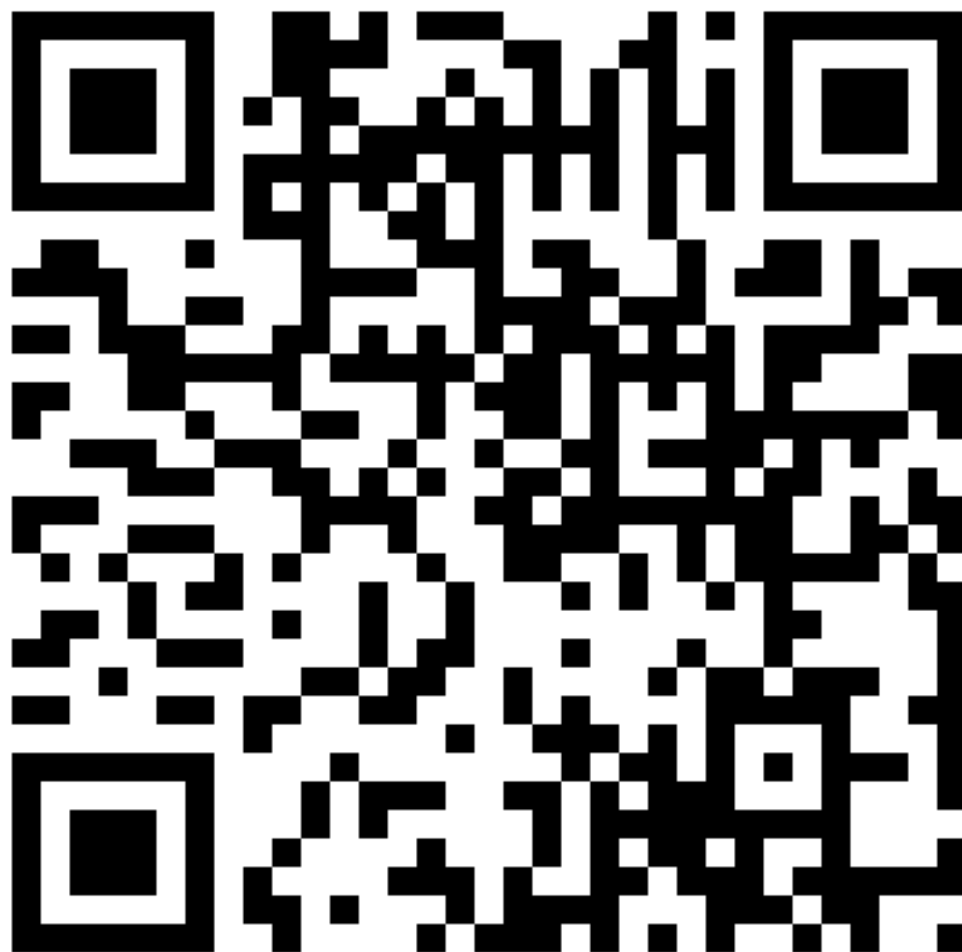
quake2-gwt-port.appspot.com

Quake 2 в браузере



quake2-gwt-port.appspot.com

Quake 2 в браузере



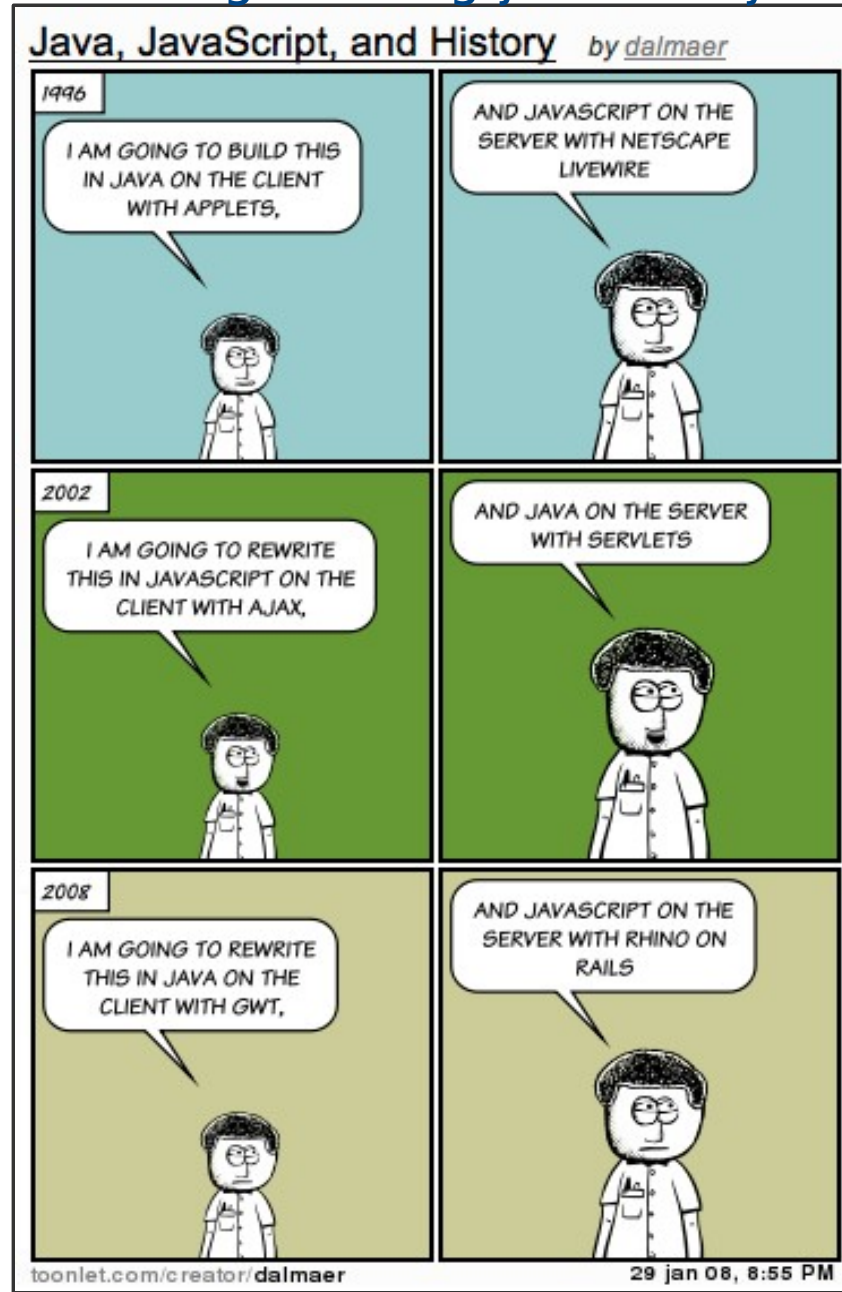
Модно
Молодёжно
Современно

/'gwɪt/

GWT активно развивается,
при том что уже содержит в себе
всё необходимое



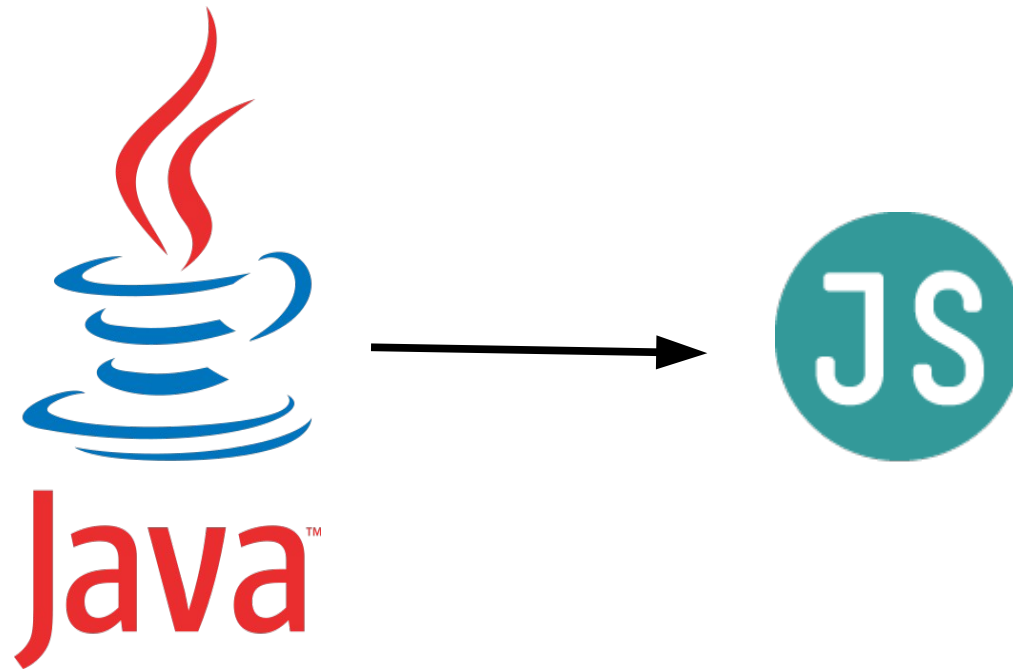
2. Концепции GWT



EntryPoint

Любой **GWT**-проект начинается с **точки входа**

- Java → JavaScript, JSNI
- Development Mode
- Code Splitting
- `<Module>.gwt.xml`
- MVC, MVP, RMVP, EventBus
- Deferred Binding
- Dependency Injection
- Remote Service
- JUnit
- Недостатки и замечания



JavaScript Native Interface

JSNI

```
public native static void <functionName>(<parameters>) /*-{  
    . . .  
    @<path.to.the.package>.<ClassName>::<methodName>(  
        /L<param-type>;/L<param-type>;...)(<arguments>);  
    . . .  
}-*/;
```

JSNI

```

public native static void getJson(int requestId, String url,
                                   StockWatcher handler) /*-{
    var callback = "callback" + requestId;

    var script = document.createElement("script");
    script.setAttribute("src", url+callback);
    script.setAttribute("type", "text/javascript");

    window[callback] = function(jsonObj) {
        handler.@com.google.gwt.sample.stockwatcher
            .client.StockWatcher::handleJsonResponse(
                Lcom/google/gwt/core/client/JavaScriptObject;)(jsonObj);
        window[callback + "done"] = true;
    }

    setTimeout(function() {
        if (!window[callback + "done"]) {
            handler.@com.google.gwt.sample.stockwatcher
                .client.StockWatcher::handleJsonResponse(
                    Lcom/google/gwt/core/client/JavaScriptObject;)(null);
        }

        document.body.removeChild(script);
        delete window[callback];
        delete window[callback + "done"];
    }, 1000);

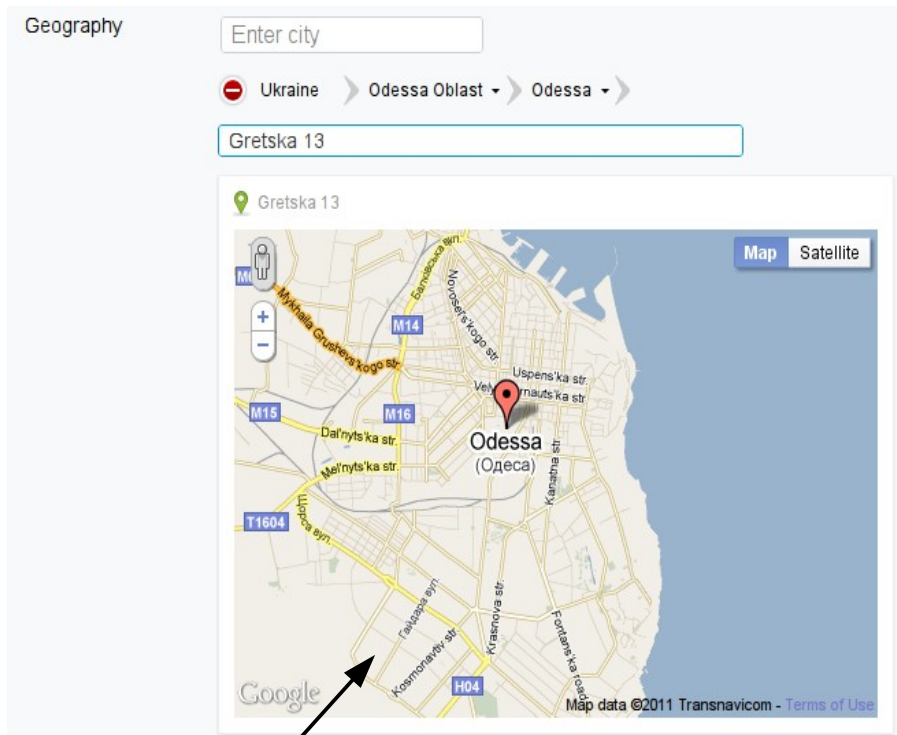
    document.body.appendChild(script);
}-*/;

```

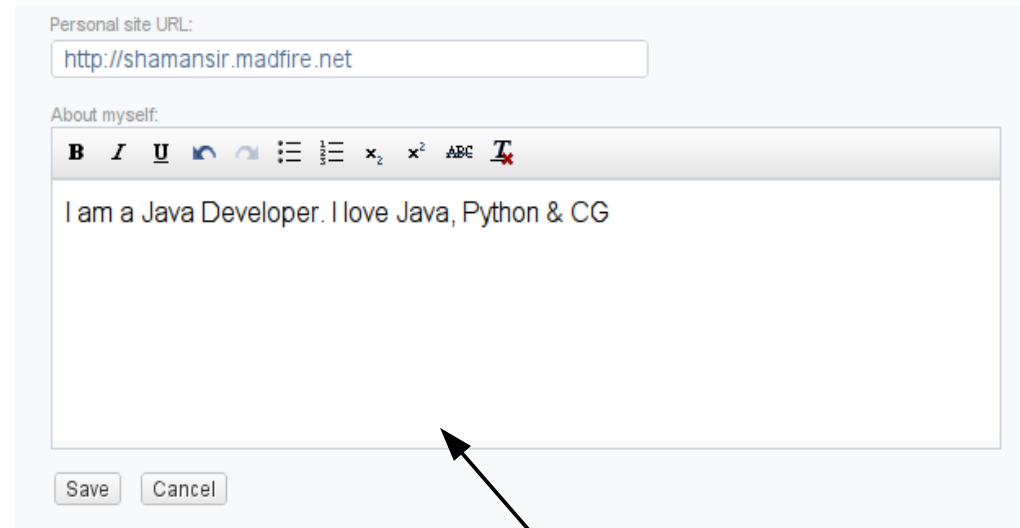
JSNI

Можно делать обёртки для нативных JavaScript виджетов.
Например, для Google Maps или любого WYSIWYG-редактора

JSNI

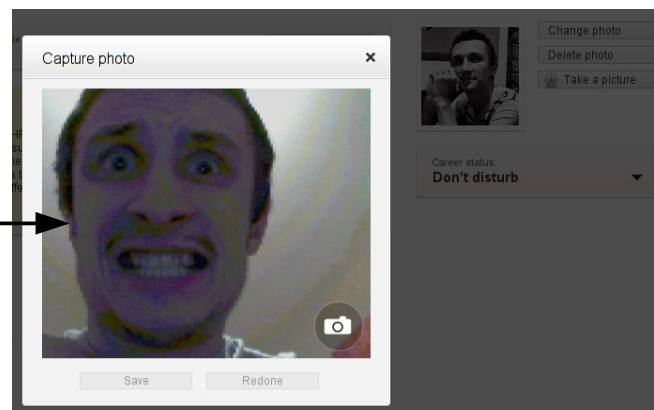


Виджет Google Maps,
встроенный с помощью JSNI



WYSIWYG-виджет,
написанный на Closure
и встроенный с помощью JSNI

Flash-объект
с JavaScript-коллбэками,
вызывающимися через JSNI



JSNI

JSNI можно использовать для
оборачивания сторонних компонентов,
написанных на **JavaScript**, в **GWT**-виджеты



Development Mode



Development Mode

<http://lc:8080/ui/?gwt.codesvr=lc:9997#!job/start>



Development Mode

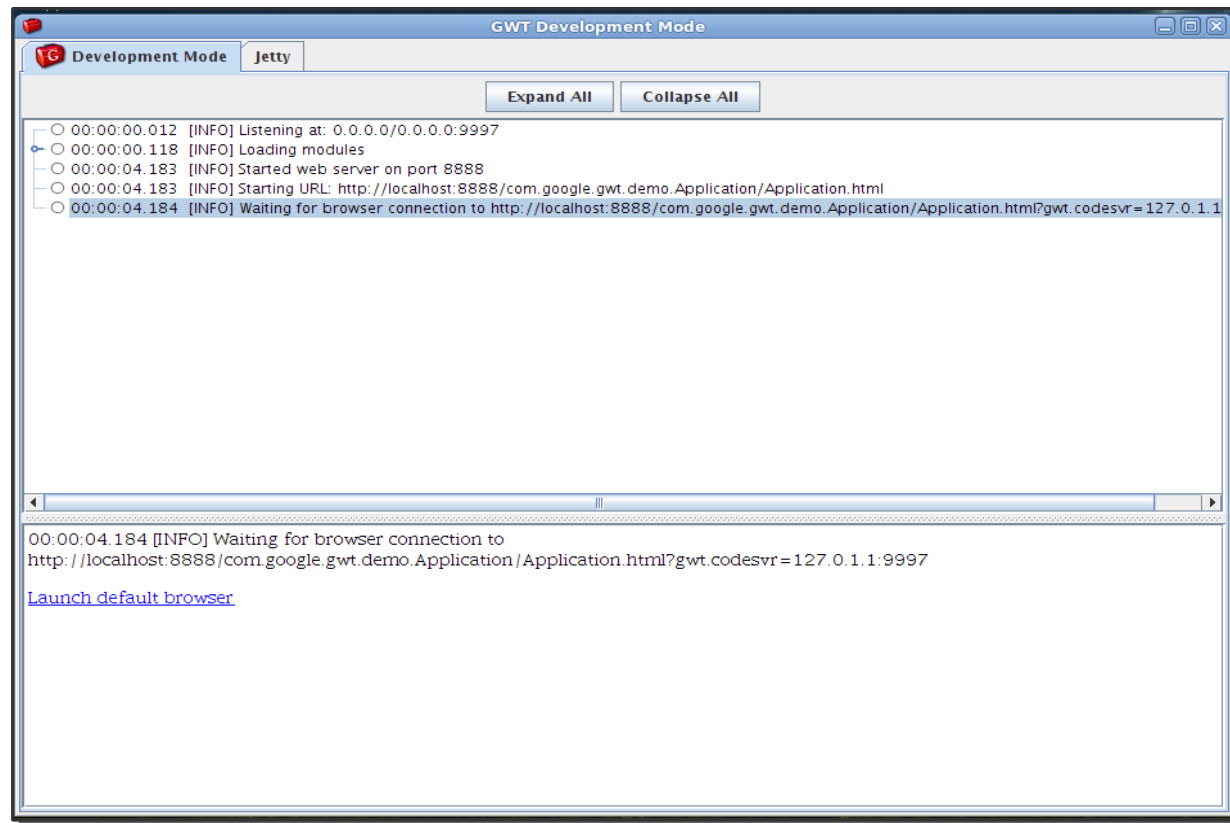
<http://lc:8080/ui/?gwt.codesvr=lc:9997#!job/start>

плагин для вашего любимого IDE
плагин для вашего любимого браузера
(но нет плагина для Opera)



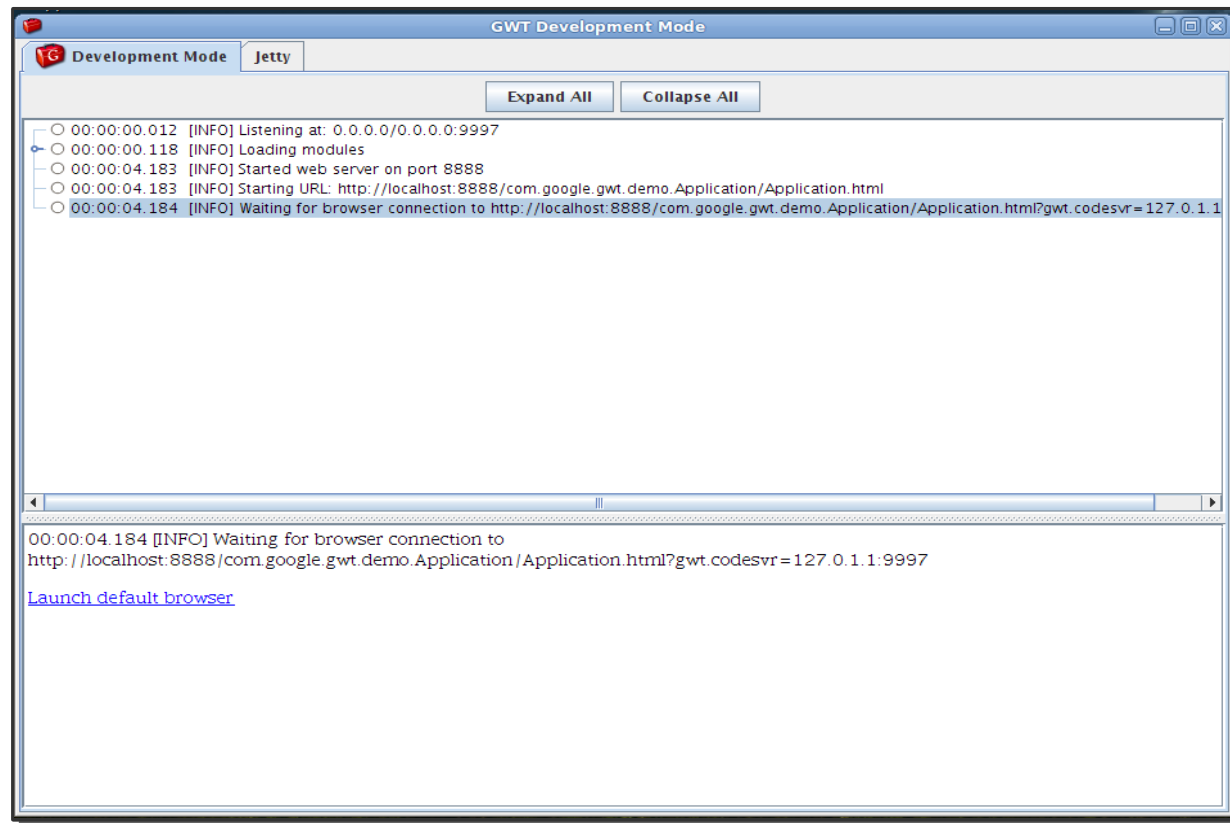
Development Mode

`http://lc:8080/ui/?gwt.codesvr=lc:9997#!job/start`

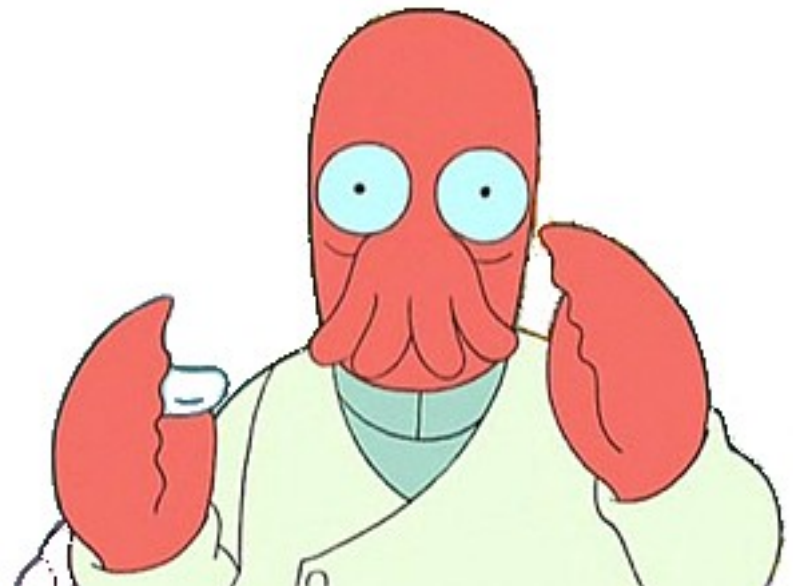


Development Mode

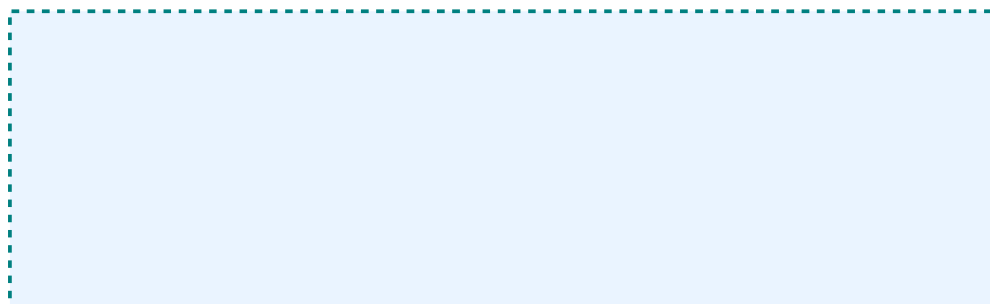
`http://lc:8080/ui/?gwt.codesvr=lc:9997#!job/start`



Development Mode помогает отлаживать проект:
при изменении **Java**-кода
достаточно нажать **Ctrl+F5** в вашем браузере —
и изменения придут в силу!



Code Splitting



когда необходимый код загружен

Code Splitting

Заклинание

```
GWT.runAsync(...)
```



когда необходимый код загружен

Code Splitting

Заклинание

```
GWT.runAsync(new RunAsyncCallback() {  
    public void onFailure(Throwable err) {  
        . . .  
    }
```

```
    public void onSuccess() {
```

```
        . . .  
        . . .  
    }
```

```
});
```

```
}
```

когда необходимый код загружен

Code Splitting

```
public static void createAsync(final MClient client) {  
  
    GWT.runAsync(new RunAsyncCallback() {  
        public void onFailure(Throwable err) {  
            client.onUnavailable();  
        }  
  
        public void onSuccess() {  
            if (instance == null) {  
                instance = new Module();  
            }  
            client.onSuccess(instance);  
        }  
    });  
}
```

когда необходимый код загружен

Code Splitting позволяет загружать части проекта *по отдельности*: за счёт этого вы можете разбить проект на *крупные модули* — и пользователи будут счастливы! Бррлыуарр!



<Module>.gwt.xml

<Module>.gwt.xml

- * Компоненты

список компонентов, которые вы используете

<Module>.gwt.xml

- * Компоненты

список компонентов, которые вы используете

- * Браузеры

список браузеров, для которых будет осуществляться компиляция

<Module>.gwt.xml

- * Компоненты

список компонентов, которые вы используете

- * Браузеры

список браузеров, для которых будет осуществляться компиляция

- * Локали

список локалей, поддерживаемых в вашем проекте

<Module>.gwt.xml

- * Компоненты

список компонентов, которые вы используете

- * Браузеры

список браузеров, для которых будет осуществляться компиляция

- * Локали

список локалей, поддерживаемых в вашем проекте

- * Отладка

включение/отключение отладочной информации

<Module>.gwt.xml

- * Компоненты

```
<inherits name="com.google.gwt.user.User"/>
```

- * Браузеры

- * Локали

- * Отладка

<Module>.gwt.xml

- * Компоненты

```
<inherits name="com.google.gwt.user.User"/>
```

- * Браузеры

```
<set-property name="user.agent"  
              value="ie6,gecko1_8,safari" />
```

- * Локали

- * Отладка

<Module>.gwt.xml

- * Компоненты

```
<inherits name="com.google.gwt.user.User"/>
```

- * Браузеры

```
<set-property name="user.agent"  
              value="ie6,gecko1_8,safari" />
```

- * Локали

```
<extend-property name="locale" values="fr_CA,de" />  
<set-property-fallback name="locale" value="fr_CA" />
```

- * Отладка

<Module>.gwt.xml

* Компоненты

```
<inherits name="com.google.gwt.user.User"/>
```

* Браузеры

```
<set-property name="user.agent"  
              value="ie6,gecko1_8,safari" />
```

* Локали

```
<extend-property name="locale" values="fr_CA,de" />  
<set-property-fallback name="locale" value="fr_CA" />
```

* Отладка [**<Module>Debug.gwt.xml**]

<Module>.gwt.xml

* Компоненты

```
<inherits name="com.google.gwt.user.User"/>
```

* Браузеры

```
<set-property name="user.agent"  
              value="ie6,gecko1_8,safari" />
```

* Локали

```
<extend-property name="locale" values="fr_CA,de" />  
<set-property-fallback name="locale" value="fr_CA" />
```

* Отладка [**<Module>Debug.gwt.xml**]

```
<inherits name="com.example.MainModule" />  
<set-property name="log_level" value="DEBUG" />
```

<Module>.gwt.xml

* Компоненты

```
<inherits name="com.google.gwt.user.User" />
```

* Браузеры

```
<set-property name="user.agent"  
              value="ie6,gecko1_8,safari" />
```

* Локали

```
<extend-property name="locale" values="fr_CA,de" />  
<set-property-fallback name="locale" value="fr_CA" />
```

* Отладка [`<Module>Debug.gwt.xml`]

```
<inherits name="com.example.MainModule" />  
<set-property name="log_level" value="DEBUG" />
```

`.gwt.xml`-файлы — то же,
что и `web.xml` для веб-приложения:
здесь хранится вся конфигурация проекта



R

M

V

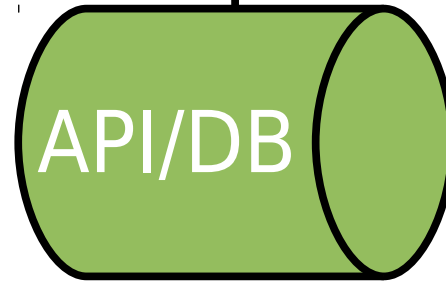
P

Model

View

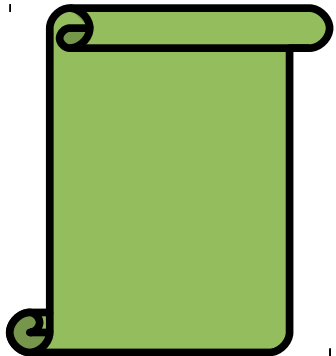
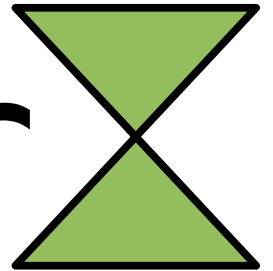
Controller

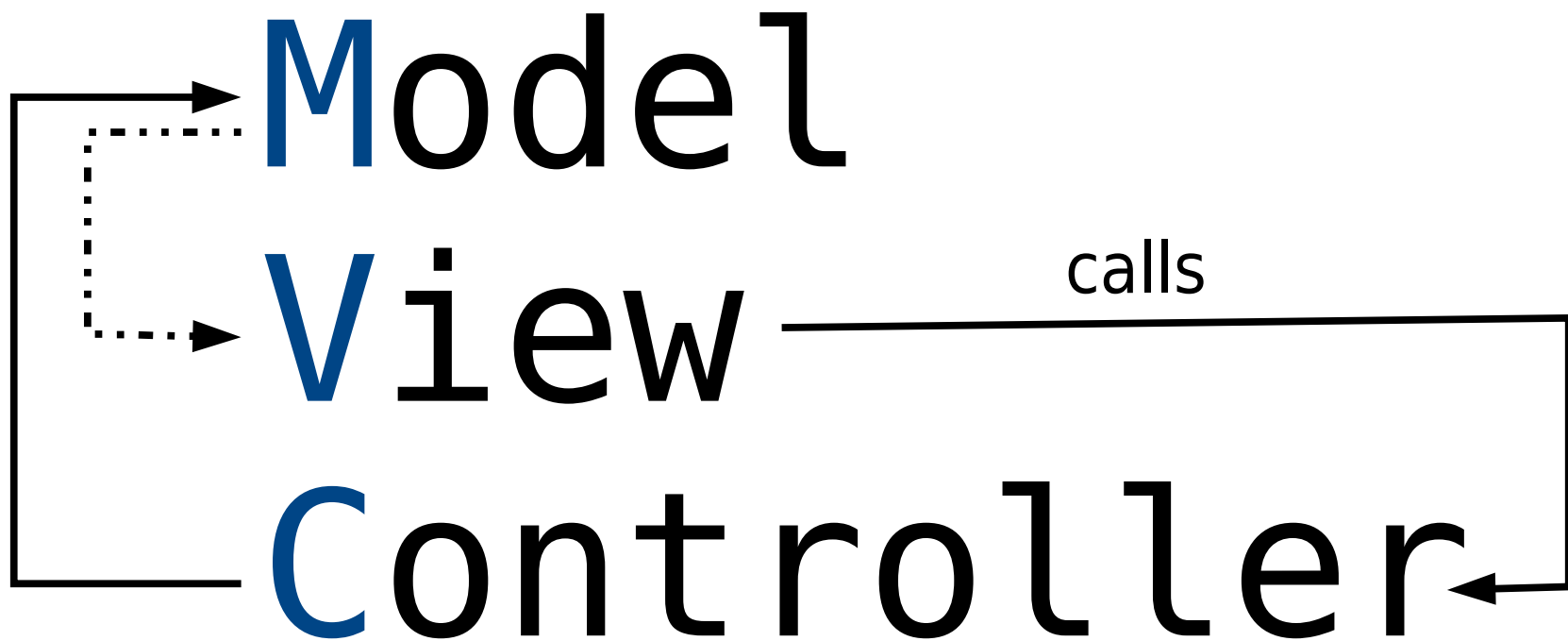
Model



View

Controller

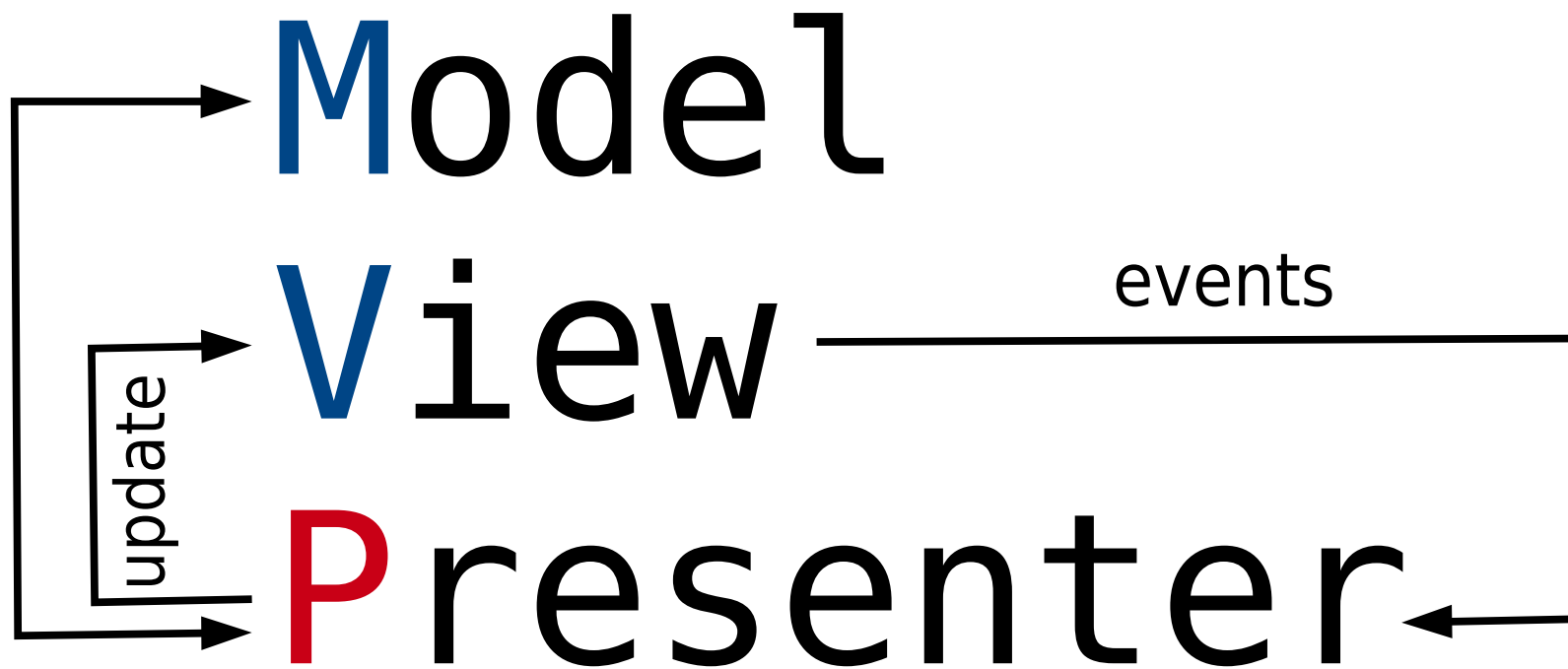




Model

View

Presenter



Reverse

Model

View

Presenter

Reverse

Model

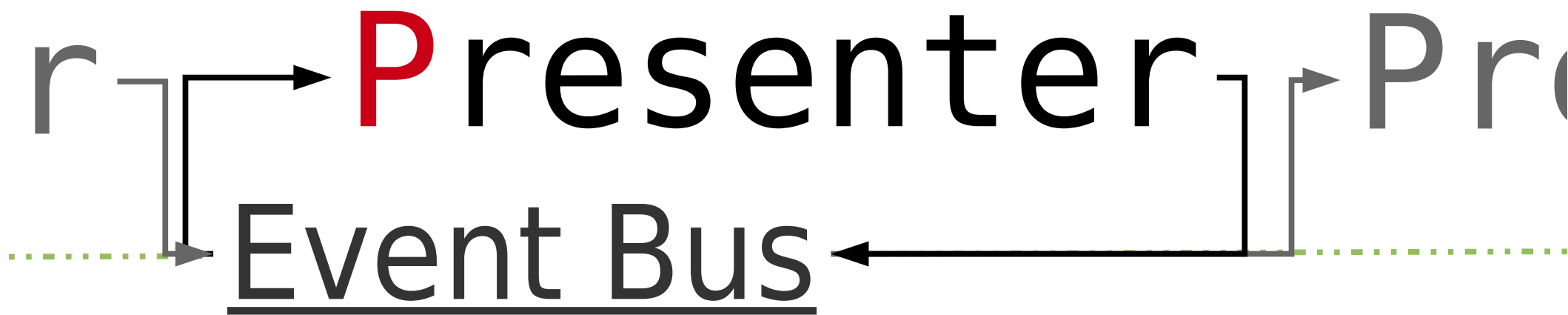
View

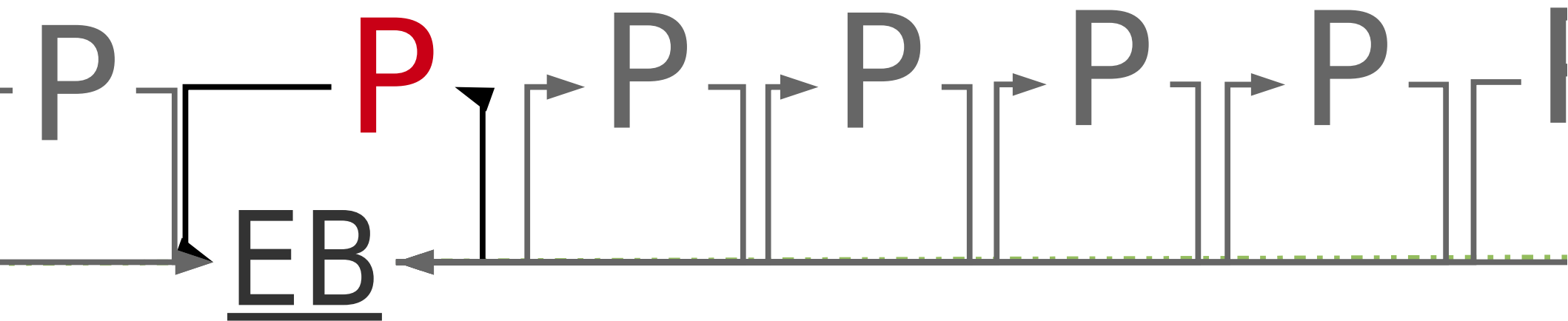
Presenter



A curved line with arrowheads at both ends connects the 'View' and 'Presenter' text elements, indicating a bidirectional relationship or interaction between them.



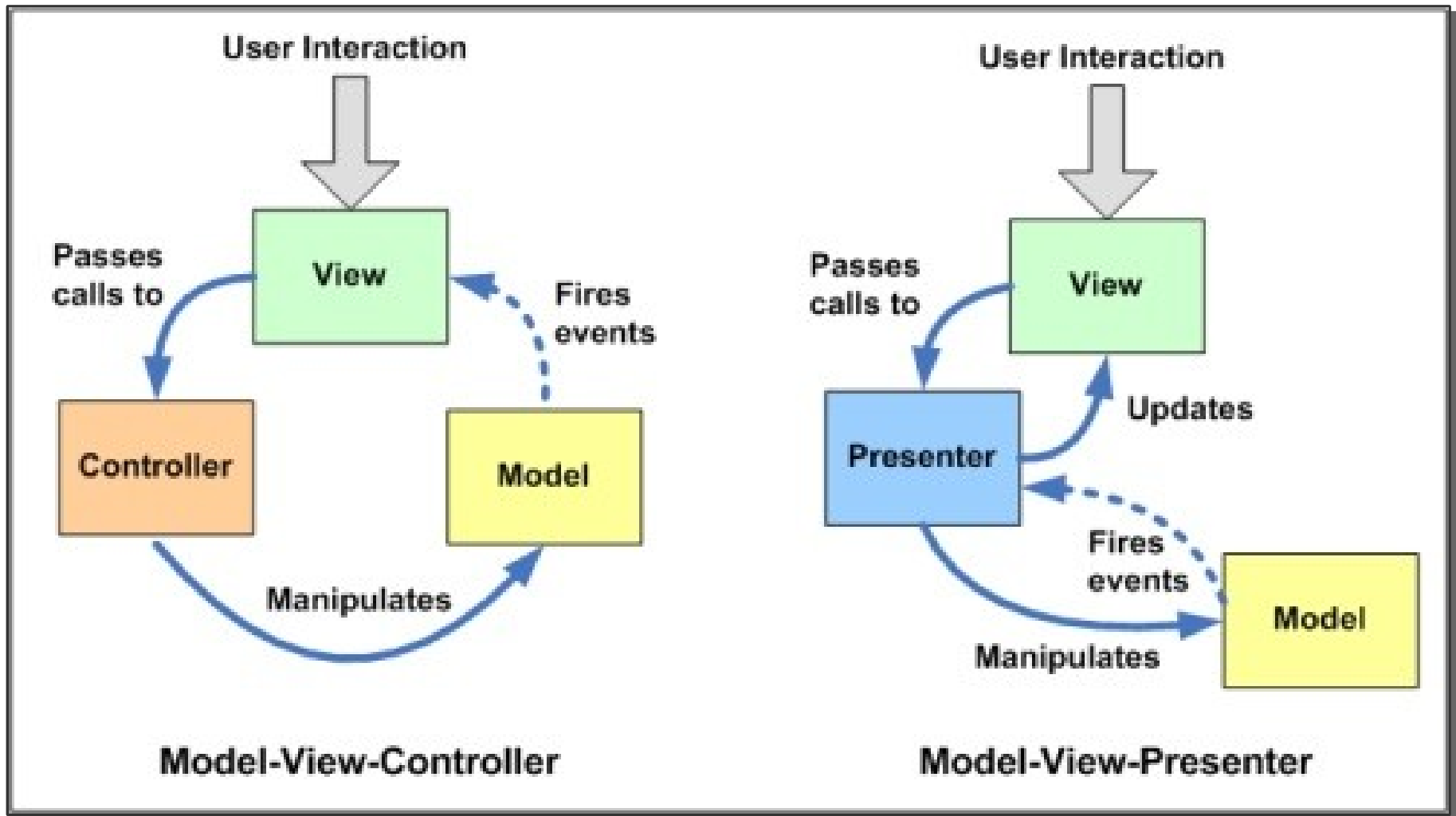




Хммм... Я, честно говоря, не понял
разницы во всех этих мэ-хрэ-эм-вэ-пэ



Различия между MVC и MVP



geekswithblogs.net/kobush/archive/2006/01/09/65305.aspx

Статья о различиях [MVC/MVP](#)



geekswithblogs.net/kobush/archive/2006/01/09/65305.aspx

Видео с примером работы EventBus



tv.jetbrains.net/videocontent/gwt-event-bus-basics

Видео с примером работы EventBus



tv.jetbrains.net/videocontent/gwt-event-bus-basics

EventBus — центральный канал сообщения



Deferred Binding

Deferred Binding

Ответ на отсутствие **Reflection**

Deferred Binding

Ответ на отсутствие Reflection

Динамическое создание имплементации на основе интерфейса (и только при необходимости).

Deferred Binding

Ответ на отсутствие Reflection

Динамическое создание имплементации
на основе интерфейса (и только при
необходимости)

Заклинание `GWTC.create(...class)`

Deferred Binding

Ответ на отсутствие Reflection

Динамическое создание имплементации на основе интерфейса (и только при необходимости).

Заклинание `GWT.create(...class)`

CompileTime-связывание

Deferred Binding

Ответ на отсутствие **Reflection**

Динамическое создание имплементации на основе интерфейса (и только при необходимости).

Заклинание `GWT.create(...class)`

CompileTime-связывание

Deferred Binding

```
PopupImpl: public void setVisible(boolean visible) {  
    // ... common code for all implementations of PopupPanel ...  
  
    // If the PopupImpl creates an iframe shim, it's also  
    // necessary to hide it as well.  
    impl.setVisible(getElement(), visible);  
}
```

Deferred Binding

```
PopupImpl: public void setVisible(boolean visible) {  
    // ... common code for all implementations of PopupPanel ...  
  
    // If the PopupImpl creates an iframe shim, it's also  
    // necessary to hide it as well.  
    impl.setVisible(getElement(), visible);  
}
```

```
PopupImplIE6: public native void setVisible(Element popup,  
                                             boolean visible) /*-{  
    if (popup.__frame) {  
        popup.__frame.style.visibility = visible ? 'visible' : 'hidden';  
    }  
}-*/;
```

Deferred Binding

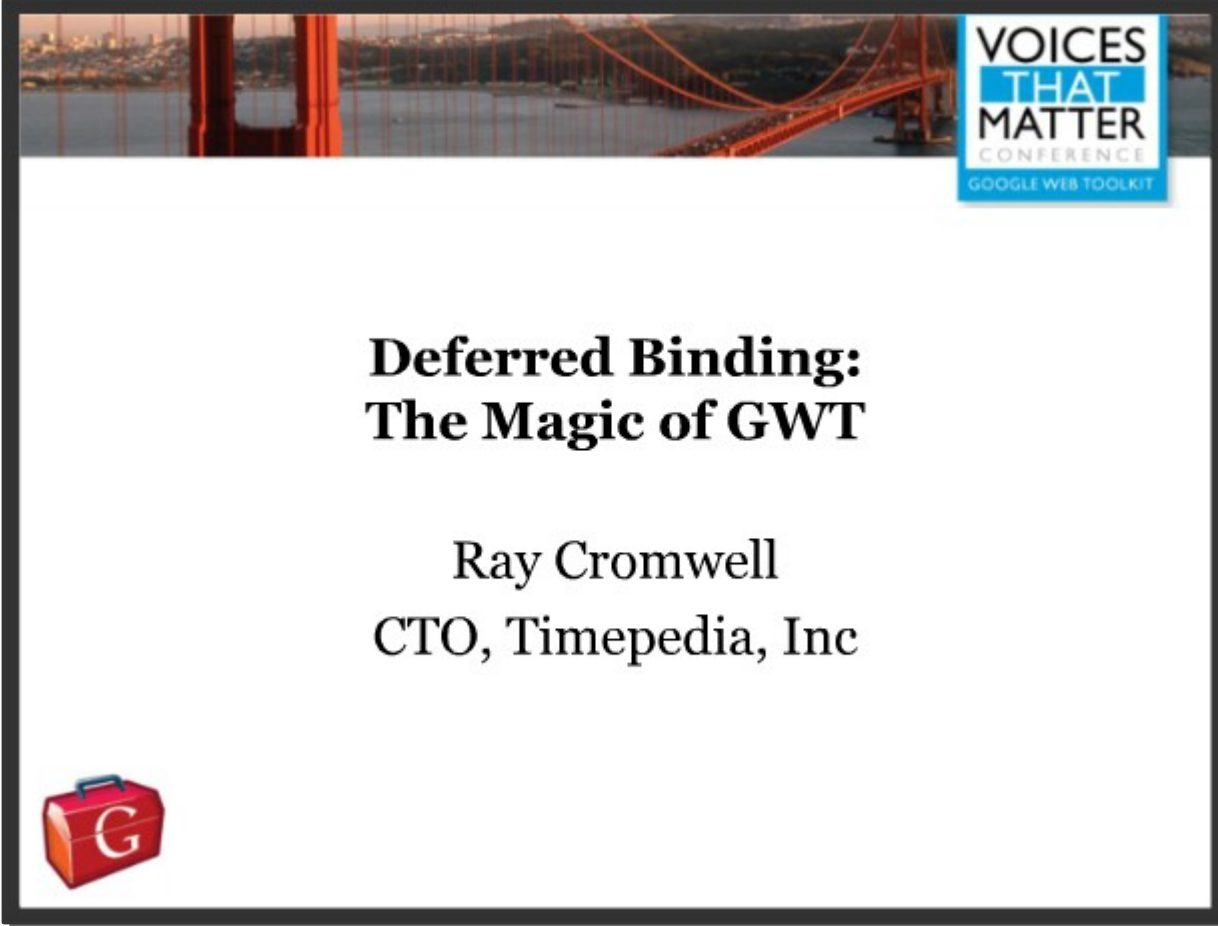
```
<replace-with class="com.google.gwt...PopupImplIE6">  
  <when-type-is class="com.google.gwt...PopupImpl" />  
  <any>  
    <when-property-is name="user.agent" value="ie6" />  
    <when-property-is name="user.agent" value="ie6_1" />  
  </any>  
</replace-with>
```

Deferred Binding

```
<replace-with class="com.google.gwt...PopupImplIE6">  
  <when-type-is class="com.google.gwt...PopupImpl" />  
  <any>  
    <when-property-is name="user.agent" value="ie6" />  
    <when-property-is name="user.agent" value="ie6_1" />  
  </any>  
</replace-with>
```

```
private static final PopupImpl impl = GWT.create(PopupImpl.class);
```


Презентация по Deferred Binding



**Deferred Binding:
The Magic of GWT**

Ray Cromwell
CTO, Timepedia, Inc

VOICES
THAT
MATTER
CONFERENCE
GOOGLE WEB TOOLKIT



www.docstoc.com/docs/53396874/Deferred-Binding-The-Magic-of-GWT

Презентация по Deferred Binding



www.docstoc.com/docs/53396874/Deferred-Binding-The-Magic-of-GWT

Deferred Binding — инструмент для создания кроссбраузерных и межязыковых реализаций. То есть для приёмов, которые будут различаться между контекстами использования проекта.



★ Внимание ★

⚠ Внимание!

Dependency Injection

Dependency Injection

Через фреймворки GWT INjection / Guice

Dependency Injection

Через фреймворки GWT INjection / Guice

Привязка экземпляров к интерфейсам
в одной точке (отделение поведения
от решения об имплементации)

Dependency Injection

Через фреймворки GWT INjection / Guice

Привязка экземпляров к интерфейсам в одной точке (отделение поведения от решения об имплементации)

Позволяет забыть об **XML**-настройках и фабриках

Dependency Injection

Через фреймворки GWT INjection / Guice

Привязка экземпляров к интерфейсам в одной точке (отделение поведения от решения об имплементации)

Позволяет забыть об XML-настройках и фабриках

Runtime-связывание

Dependency Injection

Через фреймворки **GWT IN**jection / Guice

Привязка экземпляров к интерфейсам в одной точке (отделение поведения от решения об имплементации)

Позволяет забыть об **XML**-настройках и фабриках

Runtime-связывание

Dependency Injection

```
class MyModule extends AbstractGinModule {  
  @Override  
  protected void configure() {  
    bind(Something.class).toProvider(SomethingProvider.class);  
    bind(Any.class).in(Singleton.class);  
    bind(Foo.class).to(SomeFooImpl.class);  
  }  
}  
  
class Bar {  
  @Inject private Any any;  
  private final Something something;  
  private final Foo foo;  
  
  @Inject  
  public Bar(Something something, Foo foo) {  
  }  
}
```

Dependency Injection

```
@GinModules(MyModule.class)
interface class MyGinjector extends Ginjector {

    public Something getSomething();
    public Foo getFoo();

}
```

Вики-страницы Guice



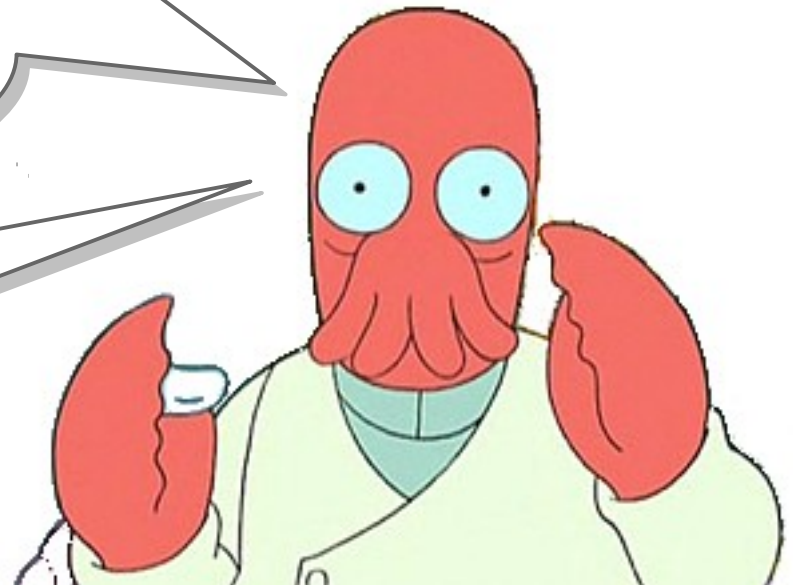
code.google.com/p/google-guice/wiki/Motivation?tm=6

Dependency Injection позволяет вам с лёгкостью управлять имплементациями логических частей проекта во время его работы.

Например, подменять драйвера баз данных или имплементации сервисов.

То, что вы делали в `application-context.xml` в Spring, но намного лучше.

Аннотации просто чудесны!



Remote Service

Remote Service

```
public interface StringReverserService extends RemoteService {  
    public String reverseString(String stringToReverse);  
}
```

Remote Service

```
public interface StringReverserService extends RemoteService {  
    public String reverseString(String stringToReverse);  
}
```

```
public interface StringReverserServiceAsync {  
    void reverseString(String stringToReverse, AsyncCallback async);  
}
```


Remote Service

```
public interface StringReverserService extends RemoteService {  
    public String reverseString(String stringToReverse);  
}
```

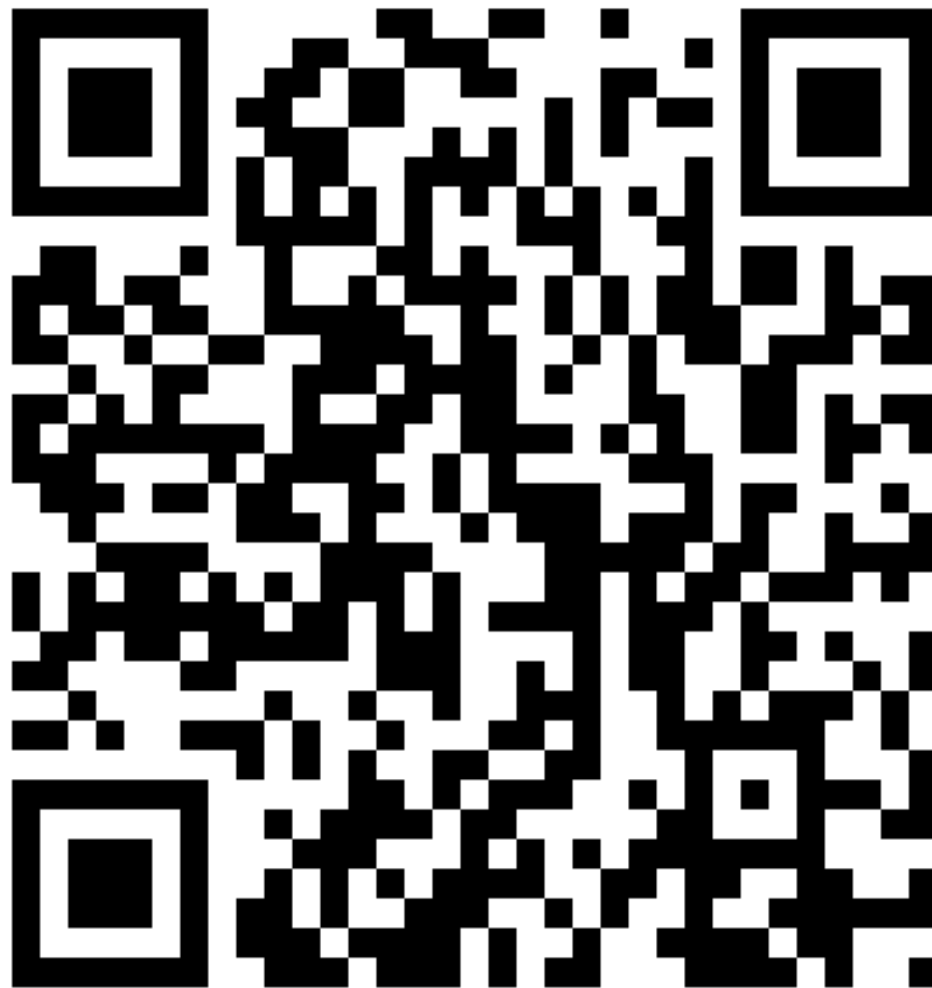
```
public interface StringReverserServiceAsync {  
    void reverseString(String stringToReverse, AsyncCallback async);  
}
```

```
public class StringReverserServiceImpl extends RemoteServiceServlet  
    implements StringReverserService { . . . }
```

Remote Service

```
public interface StringReverserService extends RemoteService {  
    public String reverseString(String stringToReverse);  
}  
  
public interface StringReverserServiceAsync {  
    void reverseString(String stringToReverse, AsyncCallback async);  
}  
  
public class StringReverserServiceImpl extends RemoteServiceServlet  
    implements StringReverserService { . . . }  
  
StringReverserServiceAsync reverserService =  
(StringReverserServiceAsync) GWT.create(StringReverserService.class);
```

Руководство по созданию Remote Services



developerlife.com/tutorials/?p=125

Remote Services — серверное API,
построенное на Java-интерфейсах



JUnit

JUnit

```
public class StockWatcherTest extends GWTTestCase {  
    public String getModuleName() {  
        return "com.google.gwt.sample.stockwatcher.StockWatcher";  
    }  
    . . .  
}
```

GWT-код легко тестировать
благодаря поддержке JUnit



Недостатки и замечания

Всё равно требуется **хорошее** знание **JavaScript**

Недостатки и замечания

Всё равно требуется **хорошее** знание JavaScript
Особенно при использовании внешних JS-библиотек

Недостатки и замечания

Всё равно требуется **хорошее** знание JavaScript
Особенно при использовании внешних JS-библиотек

«Ручная» вёрстка (**HTMLPanel**) становится **не** поддерживаемой
и **не** кроссбраузерной

Недостатки и замечания

Всё равно требуется **хорошее** знание JavaScript
Особенно при использовании внешних JS-библиотек

«Ручная» вёрстка (**HTMLPanel**) становится **не** поддерживаемой
и **не** кроссбраузерной

GWT — для *веб-приложений*, а не для *вычурных* порталов

Недостатки и замечания

Всё равно требуется **хорошее** знание JavaScript
Особенно при использовании внешних JS-библиотек

«Ручная» вёрстка (**HTMLPanel**) становится **не** поддерживаемой
и **не** кроссбраузерной

GWT — для *веб-приложений*, а не для *вычурных* порталов

Писать свои или расширять существующие компоненты *можно*,
но тогда забота о *кроссбраузерности* — на вас

Недостатки и замечания

Всё равно требуется **хорошее** знание JavaScript
Особенно при использовании внешних JS-библиотек

«Ручная» вёрстка (**HTMLPanel**) становится **не** поддерживаемой
и **не** кроссбраузерной

GWT — для *веб-приложений*, а не для *вычурных* порталов

Писать свои или расширять существующие компоненты *можно*,
но тогда забота о *кроссбраузерности* — на вас

И, кстати, придётся делать верстальщикам *подробный* экскурс
в систему компонентов **GWT**

Недостатки и замечания

Всё равно требуется **хорошее** знание JavaScript
Особенно при использовании внешних JS-библиотек

«Ручная» вёрстка (HTMLPanel) становится **не** поддерживаемой
и **не** кроссбраузерной

GWT — для *веб-приложений*, а не для *вычурных* порталов

Писать свои или расширять существующие компоненты *можно*,
но тогда забота о *кроссбраузерности* — на вас

И, кстати, придётся делать верстальщикам *подробный* экскурс
в систему компонентов GWT

Нет ориентировки на не-**Java** Server-Side (**Python** & **GAE**, например)

Недостатки и замечания

Всё равно требуется **хорошее** знание JavaScript
Особенно при использовании внешних JS-библиотек

«Ручная» вёрстка (HTMLPanel) становится **не** поддерживаемой
и **не** кроссбраузерной

GWT — для *веб-приложений*, а не для *вычурных* порталов

Писать свои или расширять существующие компоненты *можно*,
но тогда забота о *кроссбраузерности* — на вас

И, кстати, придётся делать верстальщикам *подробный* экскурс
в систему компонентов GWT

Нет ориентировки на не-**Java** Server-Side (**Python** & **GAE**, например)

Ну что ж, построим всё на **RequestBuilder**

Недостатки и замечания

Всё равно требуется **хорошее** знание JavaScript
Особенно при использовании внешних JS-библиотек

«Ручная» вёрстка (HTMLPanel) становится **не** поддерживаемой
и **не** кроссбраузерной

GWT — для *веб-приложений*, а не для *вычурных* порталов
Писать свои или расширять существующие компоненты *можно*,
но тогда забота о *кроссбраузерности* — на вас
И, кстати, придётся делать верстальщикам *подробный* экскурс
в систему компонентов GWT

Нет ориентировки на не-Java Server-Side (Python & GAE, например)
Ну что ж, построим всё на RequestBuilder

Одна **JS**-ошибка валит **всё** приложение

Недостатки и замечания

Всё равно требуется **хорошее** знание JavaScript
Особенно при использовании внешних JS-библиотек

«Ручная» вёрстка (HTMLPanel) становится **не** поддерживаемой
и **не** кроссбраузерной

GWT — для *веб-приложений*, а не для *вычурных* порталов
Писать свои или расширять существующие компоненты *можно*,
но тогда забота о *кроссбраузерности* — на вас
И, кстати, придётся делать верстальщикам *подробный* экскурс
в систему компонентов GWT

Нет ориентировки на не-Java Server-Side (Python & GAE, например)
Ну что ж, построим всё на RequestBuilder

Одна **JS**-ошибка валит **всё** приложение
Впрочем, есть же `GWT.setUncaughtExceptionHandler`

Недостатки и замечания

Всё равно требуется **хорошее** знание JavaScript
Особенно при использовании внешних JS-библиотек

«Ручная» вёрстка (HTMLPanel) становится **не** поддерживаемой
и **не** кроссбраузерной

GWT — для *веб-приложений*, а не для *вычурных* порталов
Писать свои или расширять существующие компоненты *можно*,
но тогда забота о *кроссбраузерности* — на вас
И, кстати, придётся делать верстальщикам *подробный* экскурс
в систему компонентов GWT

Нет ориентировки на не-Java Server-Side (Python & GAE, например)
Ну что ж, построим всё на RequestBuilder

Одна JS-ошибка валит **всё** приложение
Впрочем, есть же `GWT.setUncaughtExceptionHandler`

JavaScript-ошибки *малоинформативны*

Недостатки и замечания

Всё равно требуется **хорошее** знание JavaScript
Особенно при использовании внешних JS-библиотек

«Ручная» вёрстка (HTMLPanel) становится **не** поддерживаемой
и **не** кроссбраузерной

GWT — для *веб-приложений*, а не для *вычурных* порталов
Писать свои или расширять существующие компоненты *можно*,
но тогда забота о *кроссбраузерности* — на вас
И, кстати, придётся делать верстальщикам *подробный* экскурс
в систему компонентов GWT

Нет ориентировки на не-Java Server-Side (Python & GAE, например)
Ну что ж, построим всё на RequestBuilder

Одна JS-ошибка валит **всё** приложение
Впрочем, есть же `GWT.setUncaughtExceptionHandler`

JavaScript-ошибки *малоинформативны*

Хотя с включённой отладочной информацией разобраться, конечно, легче

Недостатки и замечания

Всё равно требуется **хорошее** знание JavaScript
Особенно при использовании внешних JS-библиотек

«Ручная» вёрстка (HTMLPanel) становится **не** поддерживаемой
и **не** кроссбраузерной

GWT — для *веб-приложений*, а не для *вычурных* порталов
Писать свои или расширять существующие компоненты *можно*,
но тогда забота о *кроссбраузерности* — на вас
И, кстати, придётся делать верстальщикам *подробный* экскурс
в систему компонентов GWT

Нет ориентировки на не-Java Server-Side (Python & GAE, например)
Ну что ж, построим всё на RequestBuilder

Одна JS-ошибка валит **всё** приложение
Впрочем, есть же `GWT.setUncaughtExceptionHandler`

JavaScript-ошибки *малоинформативны*
Хотя с включённой отладочной информацией разобраться, конечно, легче

Development Mode работает медленнее, чем реальный код:
проблемы с контролем событий

Недостатки и замечания

Всё равно требуется **хорошее** знание JavaScript
Особенно при использовании внешних JS-библиотек

«Ручная» вёрстка (HTMLPanel) становится **не** поддерживаемой
и **не** кроссбраузерной

GWT — для *веб-приложений*, а не для *вычурных* порталов
Писать свои или расширять существующие компоненты *можно*,
но тогда забота о *кроссбраузерности* — на вас
И, кстати, придётся делать верстальщикам *подробный* экскурс
в систему компонентов GWT

Нет ориентировки на не-Java Server-Side (Python & GAE, например)
Ну что ж, построим всё на RequestBuilder

Одна JS-ошибка валит **всё** приложение
Впрочем, есть же `GWT.setUncaughtExceptionHandler`

JavaScript-ошибки *малоинформативны*
Хотя с включённой отладочной информацией разобраться, конечно, легче

Development Mode работает медленнее, чем реальный код:
проблемы с контролем событий

Ну-с, придётся поотлаживать **alert'**ами

Недостатки и замечания

Всё равно требуется **хорошее** знание JavaScript
Особенно при использовании внешних JS-библиотек

«Ручная» вёрстка (HTMLPanel) становится **не** поддерживаемой
и **не** кроссбраузерной

GWT — для *веб-приложений*, а не для *вычурных* порталов
Писать свои или расширять существующие компоненты *можно*,
но тогда забота о *кроссбраузерности* — на вас
И, кстати, придётся делать верстальщикам *подробный* экскурс
в систему компонентов GWT

Нет ориентировки на не-Java Server-Side (Python & GAE, например)
Ну что ж, построим всё на RequestBuilder

Одна JS-ошибка валит **всё** приложение
Впрочем, есть же `GWT.setUncaughtExceptionHandler`

JavaScript-ошибки *малоинформативны*
Хотя с включённой отладочной информацией разобраться, конечно, легче

Development Mode работает медленнее, чем реальный код:
проблемы с контролем событий
Ну-с, придётся поотлаживать `alert`'ами

Недостатки и замечания

Всё равно требуется **хорошее** знание **JavaScript**

«Ручная» вёрстка (**HTMLPanel**) становится **не** поддерживаемой
и **не** кроссбраузерной

Нет ориентировки на не-**Java** Server-Side (**Python** & **GAE**, например)

Одна **JS**-ошибка валит **всё** приложение

JavaScript-ошибки *малоинформативны*

Development Mode работает медленнее, чем реальный код:
проблемы с контролем событий

Недостатки и замечания

Всё равно требуется **хорошее** знание **JavaScript**
Особенно при использовании внешних JS-библиотек

«Ручная» вёрстка (**HTMLPanel**) становится **не** поддерживаемой
и **не** кроссбраузерной

GWT — для *веб-приложений*, а не для *вычурных* порталов
Писать свои или расширять существующие компоненты *можно*,
но тогда забота о *кроссбраузерности* — на вас
И, кстати, придётся делать верстальщикам *подробный* экскурс
в систему компонентов GWT

Нет ориентировки на не-**Java** Server-Side (**Python** & **GAE**, например)
Ну что ж, построим всё на RequestBuilder

Одна **JS**-ошибка валит **всё** приложение
Впрочем, есть же `GWT.setUncaughtExceptionHandler`

JavaScript-ошибки *малоинформативны*
Хотя с включённой отладочной информацией разобраться, конечно, легче

Development Mode работает медленнее, чем реальный код:
проблемы с контролем событий
Ну-с, придётся поотлаживать `alert`'ами

Здесь обсуждают [недостатки GWT](#)



www.linux.org.ru/forum/talks/4497412

Каждый обоснованный недостаток
GWT имеет разумное решение.



Резюме по оптимизации GWT-кода

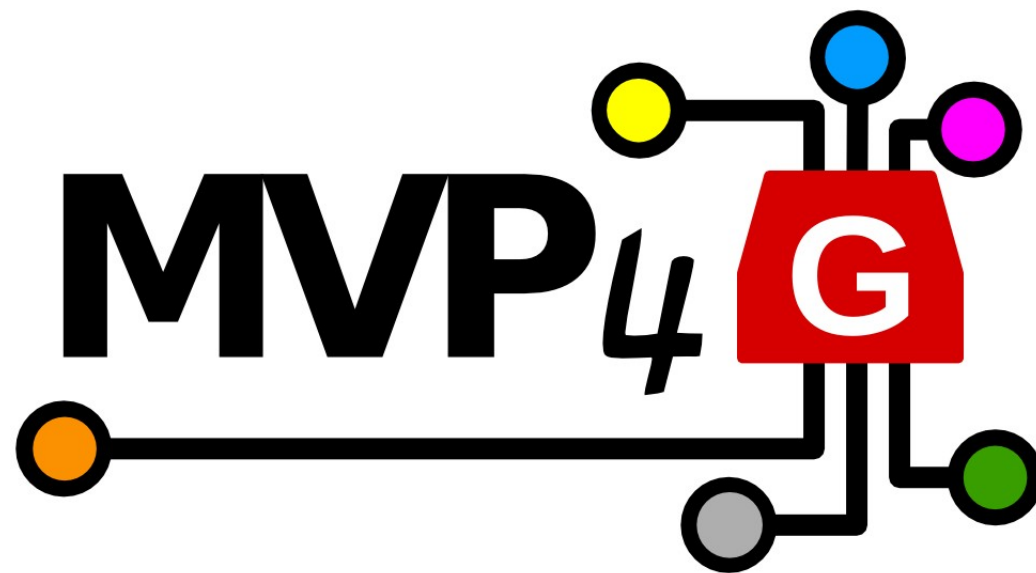


galak-sandbox.blogspot.com/2010/10/gwt.html

А я бы уже хлопнул **пивка**!



3..mvp4g



Страница фреймворка [mvp4g](https://code.google.com/p/mvp4g/)



code.google.com/p/mvp4g/

- Чем помогает?
- Система аннотаций
- Реализация RMVP
- Реализация EventBus
- URL, HistoryConverters, #!
- Мультимодульность
- PlaceService
- Замечания

Фреймворк `mvn4g` облегчает

Фреймворк `mvp4g` облегчает

Работу с `(R)MVP`

Фреймворк **mvpr4g** облегчает

Работу с (R)MVP

Организацию **мульти-модульного** приложения

Фреймворк **mvr4g** облегчает

Работу с (R)MVP

Организацию *мульти-модульного* приложения

Проектирование и разработку *шин событий*

Фреймворк `mvr4g` облегчает

Работу с (R)MVP

Организацию *мульти-модульного* приложения

Проектирование и разработку *шин событий*

Работу с *историей* (в т.ч. *хэшбэнги* **#!**)

Фреймворк **mvr4g** облегчает

Работу с (R)MVP

Организацию *мульти-модульного* приложения

Проектирование и разработку *шин событий*

Работу с историей (в т.ч. *хэшбэнги* #!)

...тоже постоянно улучшается

Фреймворк `mvr4g` облегчает

Работу с (*R*)*MVP*

Организацию *мульти-модульного* приложения

Проектирование и разработку *шин событий*

Работу с *историей* (в т.ч. *хэшбэнги* *#!*)

...тоже постоянно улучшается

Showcase фреймворка mvp4g


Mvp4g Showcase

- TestMvp4g
- EmployeeAdmin
- EmployeeAdminWithGXT
- Mvp4gHistory
- Mvp4gModules
- MailWithMvp4g**

MailWithMvp4g

This example shows the GWT Mail example converted with Mvp4g. It illustrates how Mvp4g can be used with UIBinder.

Demo Event Bus Interface



Mailboxes

- foo@example.com
- Inbox
- Drafts
- Templates
- Sent
- Trash

Tasks

Contacts

Welcome back, **foo@example.com**
[Sign Out](#) [About](#)

Sender	Email	Subject	1 - 20 of 64	older >
markboland05	mark@example.com	URGENT -[Mon, 24 Apr 2006 02:17:27 +0000]		
Hollie Voss	hollie@example.com	URGENT TRANSACTION -[Sun, 23 Apr 2006 13:10:03 +0000]		
boticario	boticario@example.com	fw: Here it comes		
Emerson Milton	emerson@example.com	voce ganho um vale presente Boticario		
Healy Colette	healy@example.com	Read this ASAP		
Brigitte Cobb	brigitte@example.com	Hot Stock Talk		
Elba Lockhart	elba@example.com	New Breed of Equity Trader		

URGENT -[Mon, 24 Apr 2006 02:17:27 +0000]

From: markboland05
To: foo@example.com

Dear Friend,

I am Mr. Mark Boland the Bank Manager of ABN AMRO BANK 101 Moorgate, London, EC2M 6SB.

I have an urgent and very confidential business proposition for you. On July 20, 2001; Mr. Zemenu Gente, a National of France, who used to be a private contractor with the Shell Petroleum Development Company in Saudi Arabia. Mr. Zemenu Gente Made a Numbered time (Fixed deposit) for 36 calendar months, valued at

mvp4gshowcase.appspot.com

Showcase фреймворка [mvp4g](#)



mvp4gshowcase.appspot.com

Сравнение GWT и фреймворка mvp4g

Mvp4g	GWTP
<p>Event bus: creating an event and registering an handler</p> <pre>//this method is defined inside your event bus interface, //you need one interface for your whole application @Event(handlers = OnePresenter.class) void oneEvent(BeanoOne one, BeanTwo two);</pre> <p>Presenter: defining the handling method</p> <pre>public class OnePresenter ... { public void onOneEvent(BeanoOne one, BeanTwo two){ //do sthg } }</pre> <p>Firing an event</p> <pre>eventBus.oneEvent(beanOne, beanTwo);</pre>	<p>Creating an event (regular way with a GWT Handler Manager)</p> <pre>public class OneEvent extends GwtEvent<OneEvent.OneHandler> { public interface OneHandler extends EventHandler { void onOneEvent(OneEvent event); } private static Type<OneHandler> TYPE = new Type<OneHandler>(); private BeanoOne one; private BeanTwo two; public OneEvent(BeanoOne one, BeanTwo two){ this.one = one; this.two = two; } @Override public Type<ShowMessageHandler> getAssociatedType() { return TYPE; } @Override protected void dispatch(OneHandler handler) { handler.onOneEvent(this); } }</pre> <p>OR Creating an event with the new tool provided by GWTP (it generates the code above thanks to the following code)</p> <pre>@GenEvent public class OneEvent { @Order(1) BeanoOne one; @Order(2) BeanTwo two; }</pre> <p>Presenter: registering the handler and defining the handling method</p> <pre>public class OnePresenter ... implements OneHandler { public void onOneEvent(BeanoOne one, BeanTwo two){ //do sthg } public void onBind(){ addRegisteredHandler(OneEvent.Type, this); } }</pre> <p>firing an event</p> <pre>eventBus.fire(new OneEvent(bean1, bean2));</pre>

code.google.com/p/mvp4g/wiki/Mvp4g_vs_GWTP

Сравнение GWT и фреймворка mvp4g



code.google.com/p/mvp4g/wiki/Mvp4g_vs_GWTP

Pierre-Laurent Coirier
plcoirier@gmail.com

Pierre-Laurent Coirier

plcoirier@gmail.com

(встречайте его на Google I/O '11)

Код с использованием фреймворка **mvp4g** значительно проще **GWT**-кода без его использования.

Это достигается за счёт грамотного использования аннотаций у Пьера.

Код пишет один человек.
Иногда лучше ему помочь.



Аннотации

Аннотации

@Debug
@Event
@EventHandler
@Events
@Filters
@Forward
@History
@InitHistory
@InjectService
@NotFoundHistory
@PlaceService
@Presenter
@Service
@Start

Есть **Annotation Processor Factory**
(валидатор аннотаций в режиме
редактирования исходного кода)

Аннотации

```
// ===== navigation events =====  
  
@Event(name = "start", navigationEvent = true,  
        handlers = AgentsListPresenter.class, calledMethod = "agentus",  
        historyConverter = JobHistoryConverter.class)  
public void job();  
  
@Event(name
```

✖ JobHistoryConverter is missing public method agentus(): String

Аннотации — сила `mvp4g`



*R*MVP

RMVP

аннотации

@Debug
@Event
@EventHandler
@Events
@Filters
@Forward
@History
@InitHistory
@InjectService
@NotFoundHistory
@PlaceService
@Presenter
@Service
@Start

RMVP
presenter

```
@Presenter(view=OneView.class)
public class OnePresenter extends
    BasePresenter<OneView, OneEventBus> {

    @Inject
    private ServiceAsync service;
```

RMVP

view

```
@Presenter(view=OneView.class)
public class OnePresenter extends
    BasePresenter<OneView, OneEventBus> {

    @Inject
    private ServiceAsync service;
}

class OneView extends Composite
    implements OneView {
    . . .
}
```

RMVP

reverse

```
@Presenter(view=OneView.class)
public class OnePresenter extends
    BasePresenter<OneView, OneEventBus> {

    @Inject
    private ServiceAsync service;
}

class OneView extends Composite
    implements OneView,
    ReverseViewInterface<OnePresenter> {
    . . .
}
```


EventBus

EventBus

аннотации

@Debug
@Event
@EventHandler
@Events
@Filters
@Forward
@History
@InitHistory
@InjectService
@NotFoundHistory
@PlaceService
@Presenter
@Service
@Start

EventBus

СОБЫТИЯ

```
@Events(startView = StartView.class)
public interface OneEventBus
    extends EventBus {

    @Event
    public void fooEvent(...);

    @Event
    public void barEvent(...);

}
```


EventBus

ХЭНДЛЕРЫ

```
@Events(startView = StartView.class)
public interface OneEventBus
    extends EventBus {

    @Event(handlers={FooPresenter.class,
                     AcmePresenter.class})
    public void fooEvent(...);

    @Event(handlers=BarPresenter.class)
    public void barEvent(...);

}

FooPresenter::onFooEvent(...) {...}
FooPresenter::onFooEvent(...) {...}
BarPresenter::onBarEvent(...) {...}
```

EventBus

активация

```
@Events(startView = StartView.class)
public interface OneEventBus
    extends EventBus {

    @Event(handlers={FooPresenter.class,
                    AcmePresenter.class},
          activate={FooPresenter.class,
                    AcmePresenter.class},
          deactivate={BarPresenter.class})
    public void fooEvent(...);

    @Event(handlers=BarPresenter.class,
          activate={BarPresenter.class},
          deactivate={FooPresenter.class,
                    AcmePresenter.class})
    public void barEvent(...);

}
```

EventBus

вещание

```
@Events(startView = StartView.class)
public interface OneEventBus
    extends EventBus {

    @Event(broadcastTo=IBroadcast.class,
        calledMethod="boo")
    public void broadcastEvent(...);

}

public class Foo implements IBroadcast {
    public void boo(...) {...};
}
```

```

@Filters(filterClasses={FilterOne.class,
                        FilterTwo.class})
@Debug(logger=CustomLogger.class)
@Events(startView = StartView.class)
public interface OneEventBus
    extends EventBus {

    . . .

}

```

EventBus

фильтры и пр.

```

class FilterOne implements EventFilter<OneEventBus> {

    @Override
    public boolean filterEvent(...) { return ...; }

}

```


History

History

аннотации

@Debug
@Event
@EventHandler
@Events
@Filters
@Forward
@History
@InitHistory
@InjectService
@NotFoundHistory
@PlaceService
@Presenter
@Service
@Start

History

передача

```
@Events(..., historyOnStart = true)
public interface OneEventBus
    extends EventBus {

    @Start
    @InitHistory
    public void start();

    @Event(handlers=...,
            navigationEvent=true,
            historyName="foo",
            historyConverter=OneHC.class)
    public void fooEvent(...);

    @NotFoundHistory
    public void show404();

}
```


History

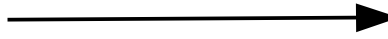
передача

/start



```
@Start
@InitHistory
public void start();
```

/foo



```
@Event(handlers=...,
        navigationEvent=true,
        historyName="foo",
        historyConverter=OneHC.class)
public void fooEvent(...);
```

```
@NotFoundHistory
public void show404();
```

```
}
```

```
@Events(..., historyOnStart = true)
public interface OneEventBus
        extends EventBus {
```

History

параметры

```
@Events(...)
public interface OneEventBus ... {

    @Event(..., navigationEvent=true,
            historyName="foo",
            historyConverter=OneHC.class)
    /foo?26;all → public void fooEvent(int id,
                                    Filter filter);

}

@History public class OneHC implements
    HistoryConverter<OneEventBus> {

    @Inject private TokenGenerator tokens;

    public void convertFromToken(...) {
        if ("foo".equals(event))
            eventBus.fooEvent(
                Integer.parseInt(params[0]),
                Filter.parse(params[1]));
    }

    public String fooEvent(...) {
        return tokens.fooEvent(id, filter);
    }

}
```

```

@Events(...)
public interface OneEventBus ... {

    @Event(...)
    /#!foo?26;all → public void fooEvent(int id,
                                           Filter filter);

}

```

History

ХЭШБЭНГИ

```

@History public class OneHC implements
    HistoryConverter<OneEventBus> {

    ...

    public boolean isCrawable() {
        return true;
    }

}

```

История и шины событий — каркас
навигационной системы вашего сайта



Мульти-
модульность

Мульти- модульность *аннотации*

@AfterLoadChildModule
@BeforeLoadChildModule
@ChildModule
@ChildModules
@DisplayChildModuleView
@HistoryName
@LoadChildModuleErrors

Мульти- модульность *URLы*

REST

объект/действие[?параметры]

company/list
company/add
company/edit?123
user/list
user/add
user/edit?39


```
public interface UserModule  
    extends Mvp4gModule {}
```

```
public interface CompanyModule  
    extends Mvp4gModule {}
```

Мульти- МОДУЛЬНОСТЬ

МОДУЛИ

```
@Events(..., module=CompanyModule.class)  
public interface CompanyEventBus  
    extends EventBus {...}
```

```
@Events(..., module=UserModule.class)  
public interface UserEventBus  
    extends EventBus {...}
```

```

@Events(...)
@ChildModules(
    @ChildModule(moduleClass=UserModule.class)
    @ChildModule(moduleClass=CompanyModule.class)
)
public interface ParentEventBus extends EventBus{

    @Event(modulesToLoad=UserModule.class)
    public void usersList();

    @Event(modulesToLoad=CompanyModule.class)
    public void companiesList();
}

```

Мульти- МОДУЛЬНОСТЬ *ШИНЫ СОБЫТИЙ*

```

@Events(..., module=UserModule.class)
public interface UserEventBus extends EventBus {

    @Event(..., handlers=UserListPresenter.class,
        historyName="list")
    public void usersList();
    . . .
}

```

Мульти- модульность . . . *асинхронная загрузка*

```
@Events(...)
@ChildModules(
    @ChildModule(moduleClass=
        UserModule.class,
        runAsync=true)
    @ChildModule(moduleClass=
        CompanyModule.class,
        runAsync=true))
public interface ParentEventBus
    extends EventBus {
    . . .
}
```

История, шины событий и модули — каркас
навигационной системы вашего сайта

Модули можно загружать асинхронно!
Значит пользователь не получит
ненужных ему килобайтов, если не будет
посещать какие-то разделы.



PlaceService

PlaceService

аннотации

@Debug
@Event
@EventHandler
@Events
@Filters
@Forward
@History
@InitHistory
@InjectService
@NotFoundHistory
@PlaceService
@Presenter
@Service
@Start

```
@PlaceService(CustomPlaceService.class)
@Events(...)
public interface MainEventBus
    extends EventBusWithLookup {
    . . .
}
```

PlaceService

переопределение

```
public class CustomPlaceService
    extends PlaceService {
    protected void convertToken(String token) { ... }
    protected String[] parseToken(String token) { ... }
    public String tokenize(String eventName,
        String param) { ... }
    . . .
}
```


Замечания

Пока что нет **layouting**-системы

Замечания

Пока что нет **layouting**-системы

Мультимодульность движется
к принципу **объект** → **действие**

Замечания

Пока что нет **layouting**-системы

Мультимодульность движется
к принципу **объект** → **действие**

Поддерживается **GIN**/Guice

Замечания

Пока что нет **layouting**-системы

Мультимодульность движется
к принципу **объект** → **действие**

Поддерживается **GIN**/Guice

Замечания

При использовании **GwtEvent**, нужно
внимательно следить за активацией
презентеров. Коллбэки — проще.

Пока что нет **layouting**-системы

Мультимодульность движется
к принципу **объект** → **действие**

Поддерживается **GIN**/Guice

Замечания


При использовании **GwtEvent**, нужно
внимательно следить за активацией
презентеров. Коллбэки — проще.

Есть **LazyView** и **LazyPresenter**

mvp4g — то, что Зойдберг прописал!



4. UI компоненты

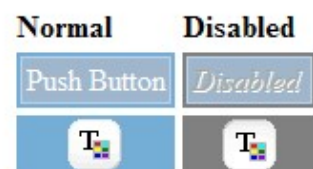
Button
PushButton
RadioButton
CheckBox
DatePicker
ToggleButton
TextBox
PasswordTextBox
TextArea
Hyperlink / Anchor
ListBox
CellList
MenuBar
Tree, CellTree
SuggestBox
RichTextArea
FlexTable, Grid, CellTable
 CellBrowser
TabBar
DialogBox

PopupPanel
StackPanel, StackLayoutPanel
HorizontalPanel
VerticalPanel
FlowPanel
VerticalSplitPanel
HorizontalSplitPanel
SplitLayoutPanel
DockPanel, DockLayoutPanel
TabPanel, TabLayoutPanel
DisclosurePanel

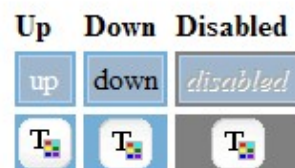
Button



PushButton



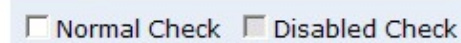
ToggleButton



RadioButton



CheckBox



DatePicker



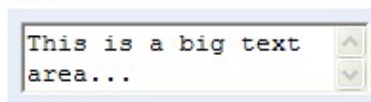
TextBox



PasswordTextBox



TextArea



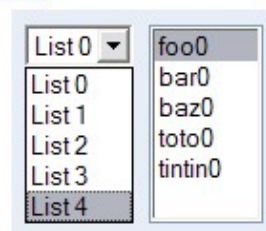
HyperLink



TabBar



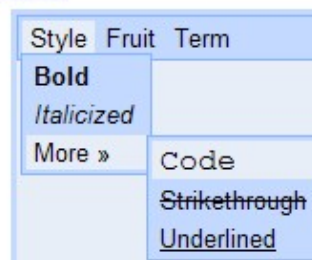
ListBox



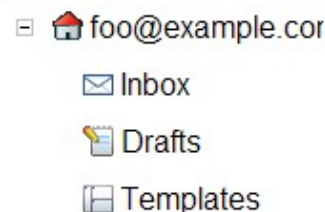
CellList



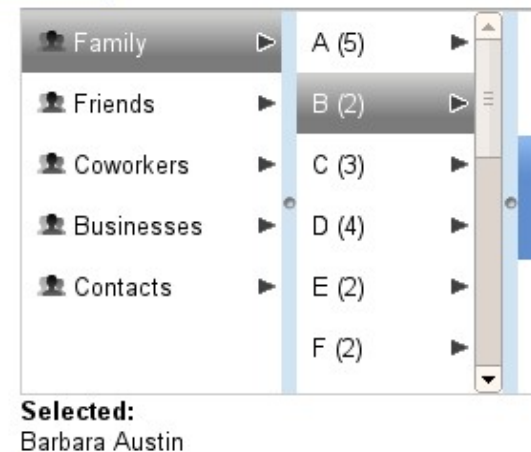
MenuBar



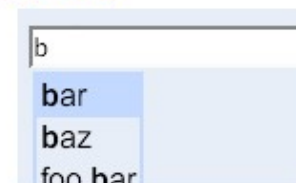
Tree, CellTree



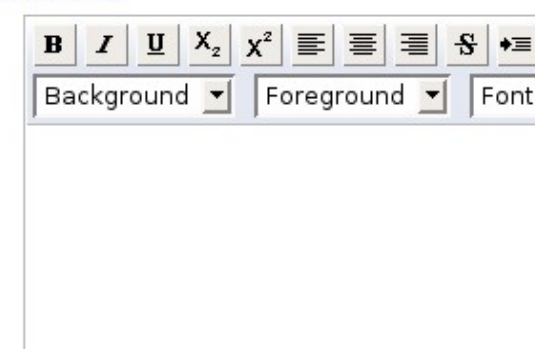
CellBrowser



SuggestBox



RichTextArea



FlexTable, Grid, and CellTable

sender	email
markboland05	mark@example.com
Hollie Voss	hollie@example.com
boticario	boticario@example.com
Emerson Milton	emerson@example.com
Healy Colette	healy@example.com
Brigitte Cobb	brigitte@example.com
Elba Lockhart	elba@example.com

PopupPanel

[Dob Sage](#)

[Ludwig von Beethoven](#)

[Richard Feynman](#)

[Alan](#)

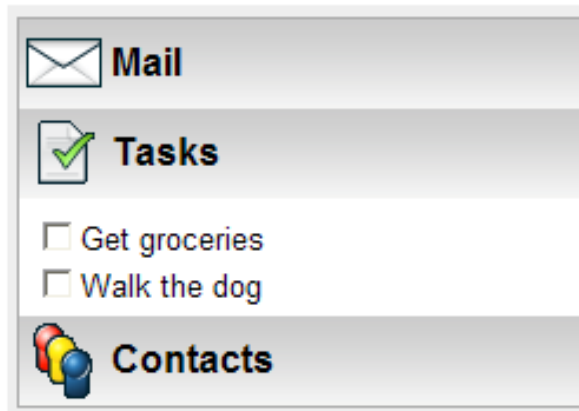
[John](#)



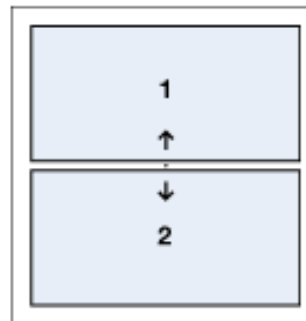
Richard Feynman

richard@example.com

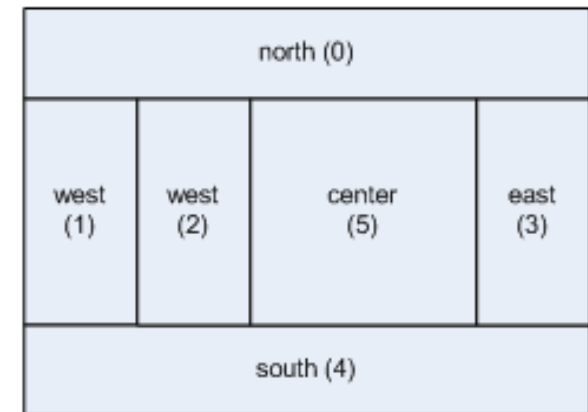
StackPanel and StackLayoutPanel*



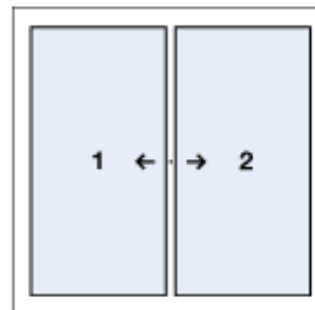
VerticalSplitPanel



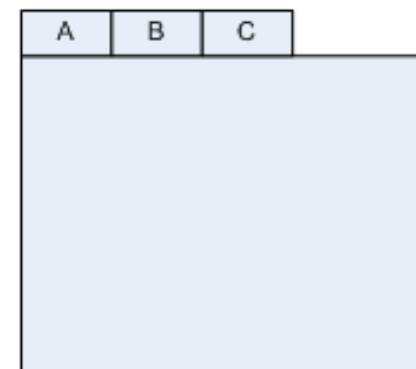
DockPanel and DockLayoutPanel*



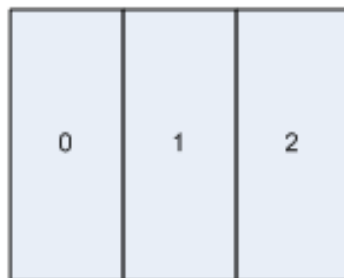
HorizontalSplitPanel



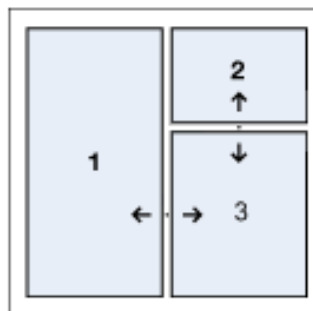
TabPanel and TabLayoutPanel*



HorizontalPanel



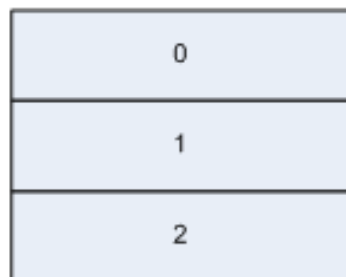
SplitLayoutPanel*



DisclosurePanel

▼ Click to dis
This widget is is :
by the disclosure

VerticalPanel



FlowPanel



Библиотека компонентов GWT



code.google.com/webtoolkit/doc/latest/RefWidgetGallery.html

UiBinder: .ui.xml

```
<ui:UiBinder xmlns:ui="urn:ui:com.google.gwt.uibinder"
              xmlns:g="urn:import:com.google.gwt.user.client.ui">
  <g:VerticalPanel styleName="my-css-style">
    <g:HorizontalPanel>
      <g:Label>Name</g:Label>
      <g:TextBox ui:field="nameBox">Babylon</g:TextBox>
    </g:HorizontalPanel>

    <g:HorizontalPanel>
      <g:Label>Family name</g:Label>
      <g:TextBox ui:field="fnameBox">Tatarsky</g:TextBox>
    </g:HorizontalPanel>

    <g:ListBox ui:field="namesLst" visibleItemCount="1" />

    <g:Button ui:field="submit">Submit</g:Button>
  </g:VerticalPanel>
</ui:UiBinder>
```

UiBinder: .java

```
public class SettingsForm extends Composite {
    interface SFormBinder extends UiBinder<Widget,
                                           SettingsForm> {}

    private static FormBuilder uiBinder =
        GWT.create(SFormBinder.class);

    @UiField TextBox nameBox;
    @UiField TextBox fnameBox;
    @UiField ListBox namesLst;

    public HelloWorld(String... names) {
        initWidget(uiBinder.createAndBindUi(this));
        for (String name : names) { namesLst.addItem(name); }
    }

    @UiHandler("submit")
    public onSubmit(ClickEvent e) { ... }
}
```


HTMLPanel

```
<g:HTMLPanel>
  <div>Some div</div>
  <div>
    <ul>
      <li>Item 1</li>
      <li>Item 2</li>
      <li>Item 3</li>
    </ul>
  </div>
  <p>
    <span>Some span</span>
  </p>
</g:HTMLPanel>
```

 Внимание!

```
.b-popup { position: absolute; }
```

Вёрстка вручную **vs.** Кроссбраузерность

```
.gwt-Button { font-size: 150%; }
```

В **GWT** есть обширнейшая библиотека
КОМПОНЕНТОВ и арсенал **лэйаутов**.

Однако, разработка сайта с собственным
уникальным стилем чревата
проблемами с кроссбраузерностью.



Кастомные компоненты

Кастомные компоненты

Не наследовать, а делегировать

Кастомные компоненты

Не наследовать, а делегировать

```
@UiConstructor  
public MyCustomWidget(String defaultText) {  
    . . .  
}  
  
public void setMaxLength(int maxLength) { ... }
```

Кастомные компоненты

Не наследовать, а делегировать

```
@UiConstructor
public MyCustomWidget(String defaultText) {
    initWidget(uiBinder.createAndBindUi(this));
}

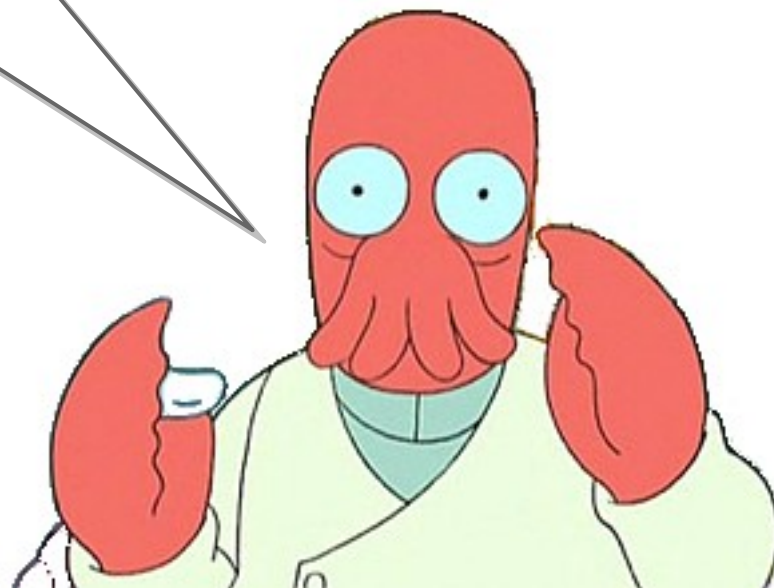
public void setMaxLength(int maxLength) { ... }
```


 Внимание!

Кастомизация **vs.** Кроссбраузерность

То есть лучше либо делать сайт
как можно проще в плане дизайна,...

...либо следует выделить
значительное количество
времени проектировщиков UI,
дизайнеров и программистов
на проработку собственной
библиотеки виджетов



Д

недостаток

Для кого-то
недостатком *было*
отсутствие **UI Designer**

5. Layouting

Зачем?

Зачем?

Если ваш сайт содержит логические блоки, которые выстраиваются в разном порядке, в зависимости от контекста.

Зачем?

Если ваш сайт содержит логические блоки, которые выстраиваются в разном порядке, в зависимости от контекста.

Например, виджеты или «портлеты»

Зачем?

Если ваш сайт содержит логические блоки, которые выстраиваются в разном порядке, в зависимости от контекста.

Например, виджеты или «портлеты»

В общем, когда это *гибкий* сайт.

Зачем?

Если ваш сайт содержит логические блоки, которые выстраиваются в разном порядке, в зависимости от контекста.

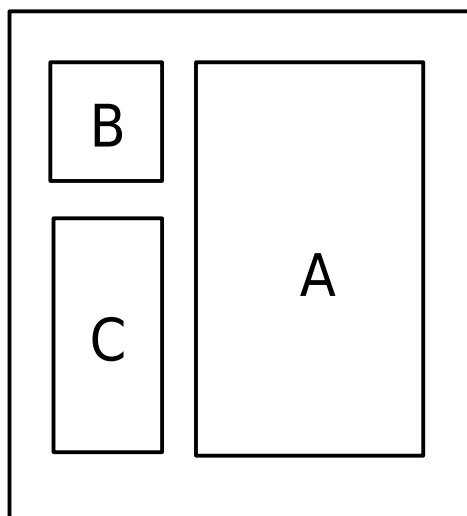
Например, виджеты или «портлеты»

В общем, когда это *гибкий* сайт.

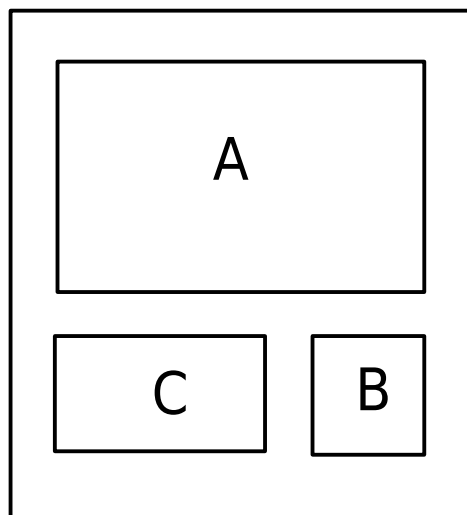
Такого механизма *пока* нет в **mvp4g**

Лэяуты

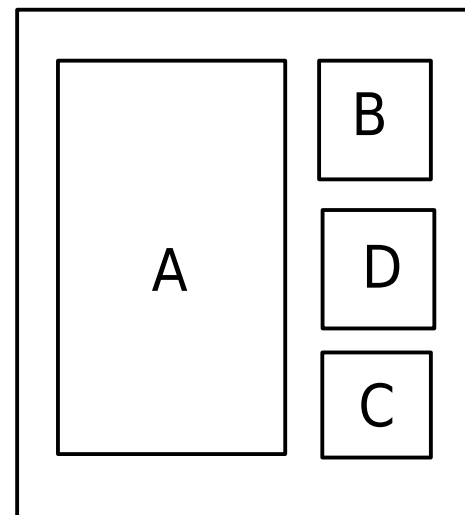
LIST



ITEM



EDIT



Лэйауты

```
public interface Layout {  
    public LayoutId id();  
    public HasWidgets place(Place place);  
    Map<Place, HasWidgets> places();  
}
```

LIST

```
public class LayoutList  
implements Layout { }
```

LayoutList.ui.xml:

```
<FlowPanel ui:field="a"/>  
<FlowPanel ui:field="b"/>  
<FlowPanel ui:field="c"/>
```

ITEM

```
public class LayoutItem  
implements Layout { }
```

LayoutItem.ui.xml:

```
<FlowPanel ui:field="a"/>  
<FlowPanel ui:field="b"/>  
<FlowPanel ui:field="c"/>
```

EDIT

```
public class LayoutEdit  
implements Layout { }
```

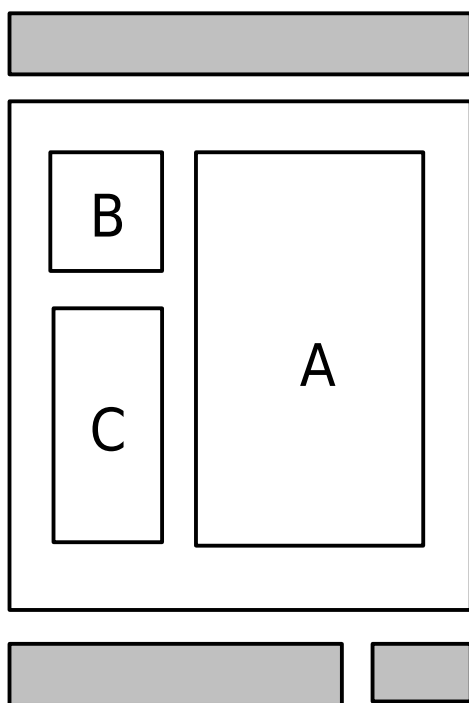
LayoutEdit.ui.xml:

```
<FlowPanel ui:field="a"/>  
<FlowPanel ui:field="b"/>  
<FlowPanel ui:field="c"/>  
<FlowPanel ui:field="d"/>
```

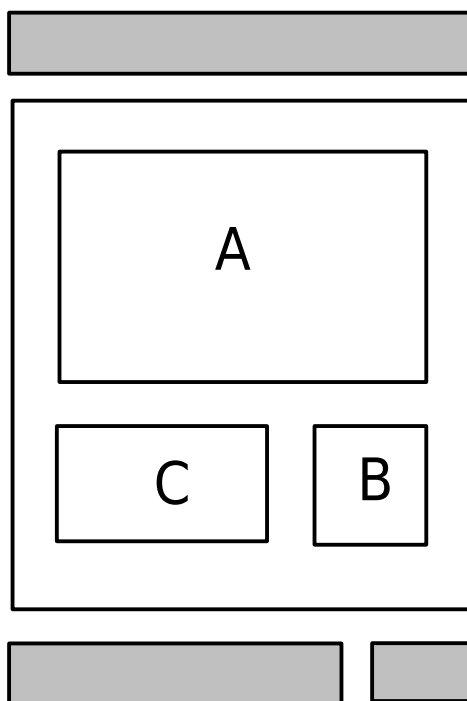
```
public enum Place { A, B, C, D};
```

Лэяуты

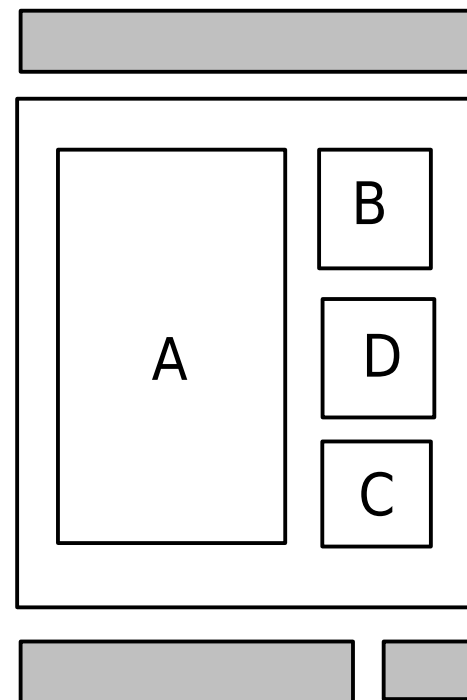
LIST



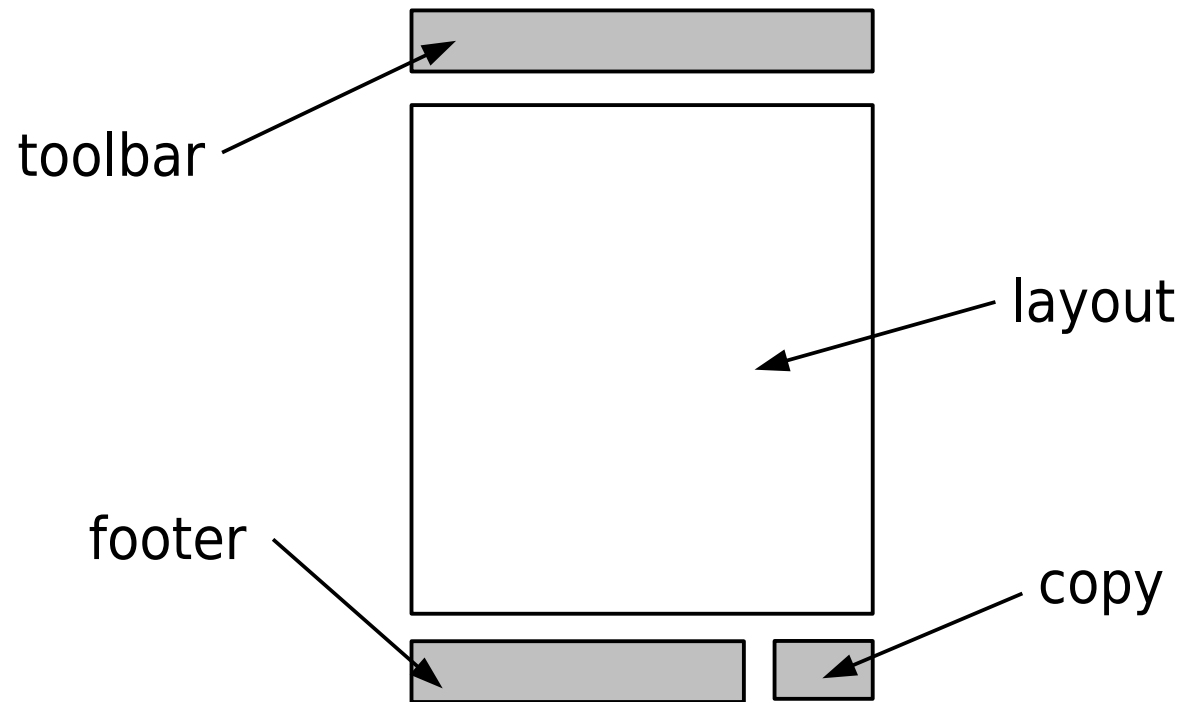
ITEM



EDIT



Базовая страница



```
BasePage.ui.xml:  
<FlowPanel ui:field="toolbar"/>  
<FlowPanel ui:field="layout"/>  
<FlowPanel ui:field="footer"/>  
<FlowPanel ui:field="copy"/>
```


Страница / Портал

```
public enum Portal implements MakesLink {

    NEWS_LIST(LayoutId.LIST, <event-spec>, <options>),
    NEWS_EDIT(LayoutId.EDIT, <event-spec>, <options>),
    NEWS_VIEW(LayoutId.ITEM, <event-spec>, <options>),
    NEWS_DELETE(LayoutId.ITEM, <event-spec>, <options>),
    USER_LIST(LayoutId.LIST, <event-spec>, <options>),
    USER_EDIT(LayoutId.EDIT, <event-spec>, <options>),
    USER_VIEW(LayoutId.ITEM, <event-spec>, <options>),
    USER_DELETE(LayoutId.ITEM, <event-spec>, <options>),
    . . .

    @Override
    public String makeLink() { . . . }

}
```

Ссылка

```
public enum Portal implements MakesLink {  
    . . .  
  
    public class PortalUrl implements MakesLink {  
        PortalUrl(Portal portal[, <params>]) { ... }  
  
        public PortalUrl addParam(...) { ... }  
        public PortalUrl fromEvent(String module,  
                                    String event,  
                                    String params) { ... }  
        @Override public String makeLink() { ... }  
    }  
}
```

Ссылка

```
public enum Portal implements MakesLink {  
    . . .  
  
    public class PortalUrl implements MakesLink {  
        PortalUrl(Portal portal[, <params>]) { ... }  
  
        public PortalUrl addParam(...) { ... }  
        public PortalUrl fromEvent(String module,  
                                   String event,  
                                   String params) { ... }  
        @Override public String makeLink() { ... }  
    }  
}  
  
History.newItem(Portal.USER_LIST.makeLink());  
History.newItem(new PortalUrl(Portal.USER_EDIT, uid).makeLink());  
History.newItem(userTokenGenerator.edit(uid));
```

Построитель лэйаутов

```
public abstract class
    LayoutBuilder<E extends ChildEventBus> {

    public CanBuildLayout prepareFor(final Portal page) {
        return new CanBuildLayout {
            public Layout build(State state) {
                layout(page, state,
                    LayoutFactory.get(page.layout).places());
            }
        }
    }

    public abstract void layout(Portal page,
                                State state,
                                Map<Place, HasWidgets> places);
}
```

Смена лэйаута

```
public class UserHistoryConverter
    implements HistoryConverter<UserEventBus> {

    public void convertFromToken(String evt, String param) {

        // можно использовать tokenGenerator
        PortalUrl curUrl =
            PortalUrl.fromToken("user", evt, param);
        Portal portal = curUrl.portal;

        eventBus.newPage(portal,
            layoutBuilder.prepareFor(portal));

        eventBus.dispatch(curUrl);
    }
}
```

Имплементация построителя

```
public class UserLayoutBuilder
    implements LayoutBuilder<UserEventBus> {

    public void layout(Portal page, State state,
        Map<Place, HasWidgets> places)

        switch (page) {
            case USER_ITEM: {
                eventBus.projectItem(places.get(Place.A));
                eventBus.projectCalendar(places.get(Place.B));
                eventBus.projectPreview(places.get(Place.C));
            } break;
            case USER_LIST: switch (state) { ... } break;
            case USER_EDIT: ... break;
        }

    }

}
```


Шины событий

```
public interface ChildEventBus {  
  
    @Event(forwardToParent=true)  
    public void newPage(Portal page, CanBuildLayout builder);  
  
    @Event(forwardToParent=true)  
    public void project(Widget what, HasWidgets where);  
  
    @Event(forwardToParent=true)  
    public void updateState(State state)  
  
}
```

Уходят в **BaseEventBus**

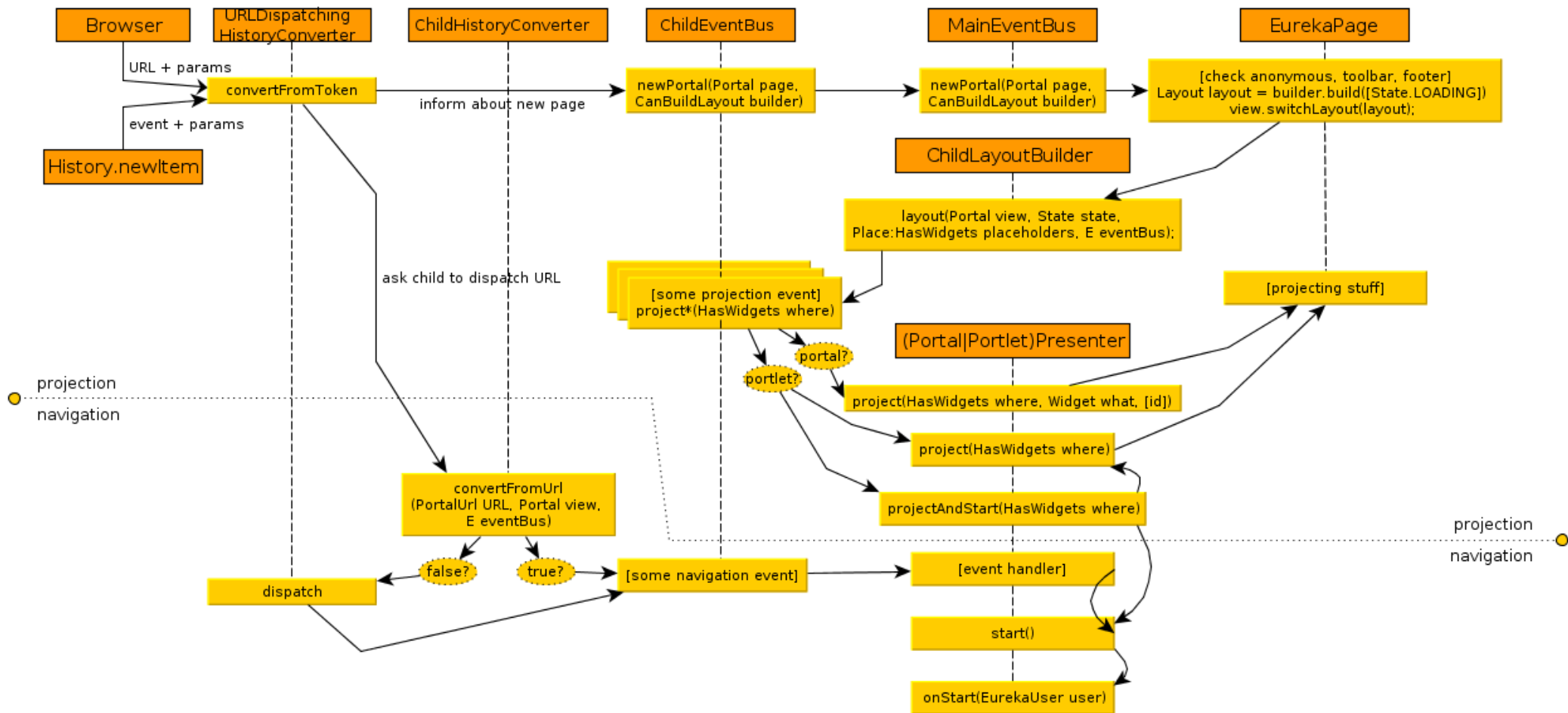
Шины событий

```
public interface UserEventBus extends ChildEventBus {

    // navigation
    @Event(navigationEvent=true, ...)
    public void list();
    @Event(navigationEvent=true,
           handlers=UserShowPresenter.class,
           historyConverter=UserHistoryConverter.class)
    public void show(String uid);
    @Event(navigationEvent=true, ...)
    public void edit(String uid);
    . . .

    // projection
    @Event(handlers=UserShowPresenter.class, calledMethod="prjItem")
    public void projectItem(HasWidgets where);
    @Event(handlers=UserShowPresenter.class, calledMethod="prjPrvw")
    public void projectPreview(HasWidgets where);
    @Event(handlers=CalendarPresenter.class, calledMethod="project")
    public void projectCalendar(HasWidgets where)

}
```



Лэяаутиг нам строить и жить помогает!



Хотел сделать вам демо, но не успел ;(

github.com/shamansir/gwt-mvp4g-layouting-demo

Хотел сделать вам демо, но не успел ;(

Следите за



github.com/shamansir/gwt-mvp4g-layouting-demo

Хотел сделать вам **демо**, но не успел ;(



github.com/shamansir/gwt-mvp4g-layouting-demo

Фото gwt-mvp4g-layouting-demo.appspot.com

QRCode gwt-mvp4g-layouting-demo.appspot.com

6. Non-Java API

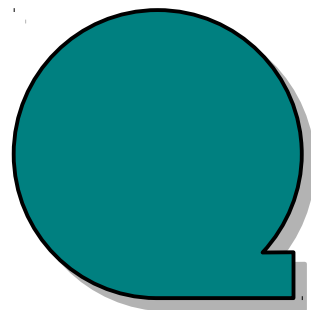
Совершенно случайно, здесь основная суть



code.google.com/p/google-web-toolkit-doc-1-5/wiki/GettingStartedJSON

Возможность — есть

RequestBuilder



Общий контекст

Цепочки вызовов

Вставка JS-объектов в HTML-страницу

(можно распарсить в JSNI)

Достаток : Независимость от сериализации

Недостаток : Нестандартный подход со всеми вытекающими

А весь **ИСХОДНЫЙ КОД** — здесь



shamansir-ru.tumblr.com/post/1728720550/deferred-api-gwt-rpc

7. i18n

Messages / Constants

```
public interface LoginMessages extends Messages {  
    public String enterName();  
    public String emailExists(String email);  
    public String emailInvalid(String email);  
    public String loginFailed(String username);  
    public String youFailedNTimes(@PluralCount int times);  
}
```

```
public interface MenuConstants extends Constants {  
    public String login();  
    public String logout();  
    public String contacts();  
    public String settings();  
}
```

Messages / Constants

LoginMessages_ru.properties

enterName = Введите имя
emailExists = E-mail {0} зарегистрирован в системе
emailInvalid = Некорректный e-mail {0}
loginFailed = Не удалось зайти пользователем {0}
youFailedNTimes = Кол-во неудачных попыток: {0,number}
youFailedNTimes[one] = {0,number} неудачная попытка
youFailedNTimes[few] = {0,number} неудачных попыток

MenuConstants_ru.properties

login= Войти
logout = Выйти
contacts = Контакты
settings = Настройки

Messages / Constants

```
LoginMessages messages = GWT.create(LoginMessages.class);  
MenuConstants constants = GWT.create(MenuConstants.class);
```

```
<ui:with type="....LoginMessages" field="messages" />  
<e:MyTextBox ui:field="box" defaultText="{messages.enterText}"
```

Messages / Constants

```
<ui:UiBinder ... xmlns:txt="ui:with:...MyMessages" />  
<txt:msg key="messageKey">Message</txt:msg>
```

Messages / Constants

```
public interface ErrorsConstants extends ConstantsWithLookup {  
    Map<String, String> errors();  
}
```

ErrorsConstants_ru.properties

ERR_101 = Ошибка авторизации

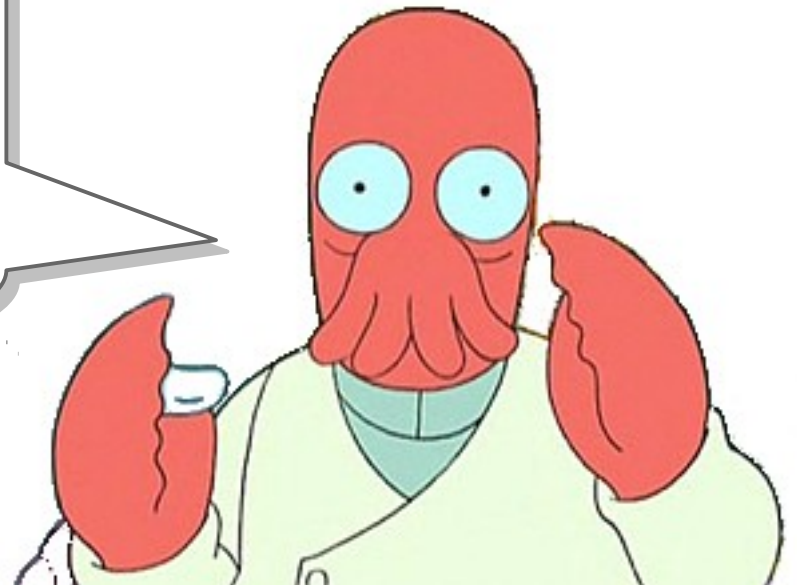
ERR_102 = Неизвестная ошибка

ERR_103 = Ресурс не найден

errors = ERR_101, ERR_102, ERR_103

Mae'n hawdd iawn i
gyfieithu prosiectau **GWT**...

...если вы научите
переводчиков пользоваться
.properties-файлами или дадите
им **PoEdit** или **Pootle**

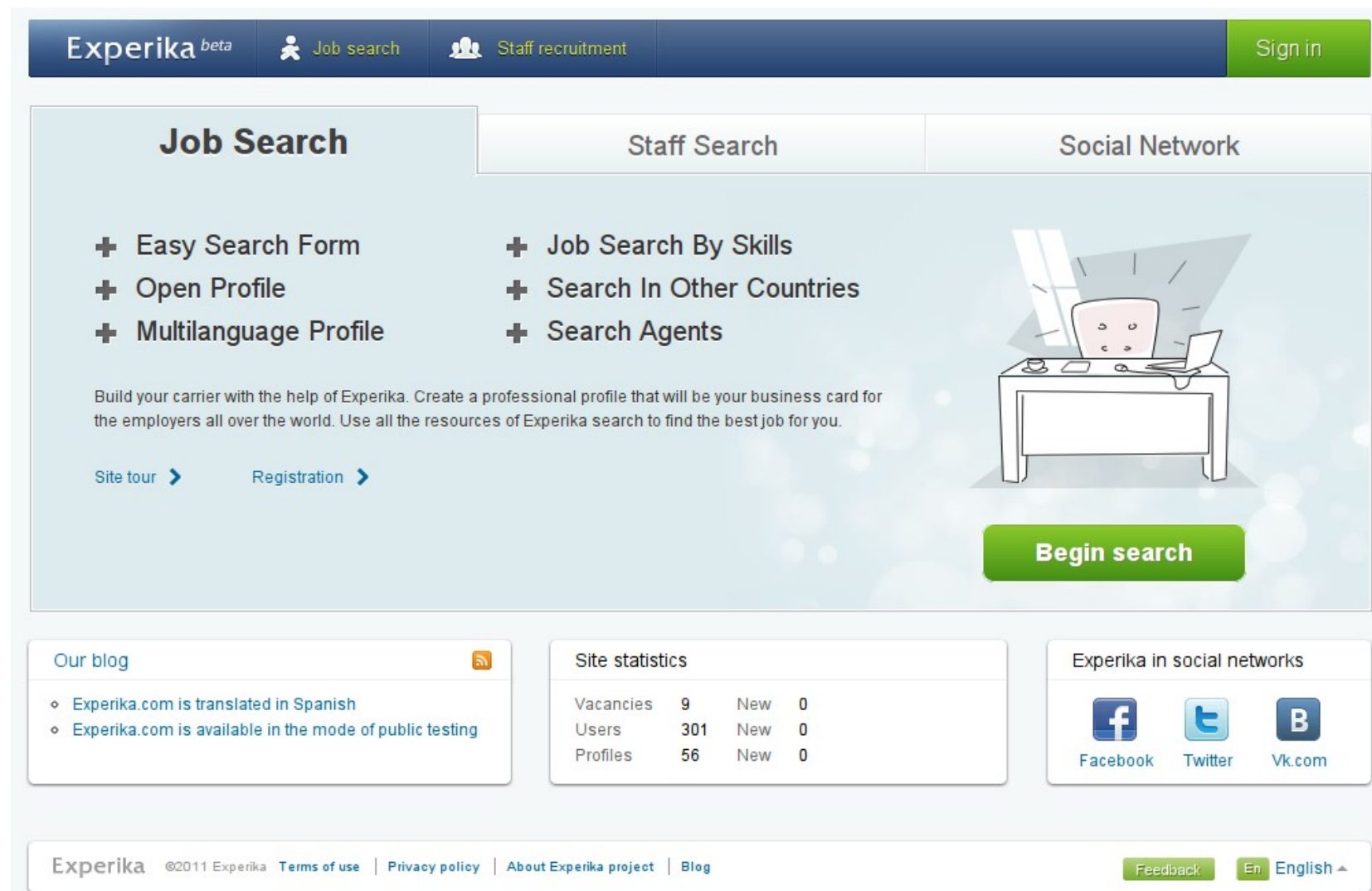


Позволяют использовать локализованные ресурсы

ResourceBundles

8. Заключение

Мы применяли эти техники на сайте Experika



experika.com

Добро пожаловать на сайт [Experika](https://experika.com)



experika.com

Благодарности

Виталию Гашку, за знакомство с mvp4g
и партнёрство

<http://twitter.com/vgashock>

Михаилу Кашкину, за наставничество

<http://www.vurt.ru>

Алексею Какунину, за 道 и 先生

<http://www.emdev.ru>

Котенко Антон



profiles.google.com/shaman.sir

Спасибо.

Ещё вопросы?



GWT FTW!



Филипп Дж. Фрай, д-р Зойдберг, Нибблер и Гипножаба —
персонажи мультсериала **Futurama**,
созданы **Мэттом Грейнингом**

Гомер Симпсон — персонаж
мультсериала **The Simpsons**,
создан **Мэттом Грейнингом**

Фразы, сообщённые этими персонажами
в данной презентации *не имеют* отношения
ни к самим персонажам,
ни к перечисленным сериалам,
ни к их создателю