

Geomajas Hammer GWT wrapper project

Geomajas Developers and Geosparc

Geomajas Hammer GWT wrapper project

by Geomajas Developers and Geosparc

1.0.0

Copyright © 2010-2014 Geosparc nv

Table of Contents

1. Introduction	1
2. Development guidelines	2

Chapter 1. Introduction

The Geomajas Hammer GWT Project is a stand-alone project under the Geomajas banner. Its goal is to provide an easy way to allow you to integrate the excellent Hammer js Javascript library into your own GWT applications

Chapter 2. Development guidelines

Inherit Hammer GTW module. To use Hammer GWT in your project you should inherit HammerGWT.gwt module.

```
<module>

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

< inherits name="com.google.gwt.i18n.I18N"/>

< inherits name="org.geomajas.hammergwt.HammerGWT" />

</module>
```

Register hammer events to existing GWT widget

```
IsWidget myWidget;

HammerTime hammerTime = HammerGWT.create(myWidget);

// add generic hammer handler

// Note that only tap and drag event are handled here

NativeHammerHandler hammerHandler = HammerGWT.on(hammerTime, new
HammerHandler() {

@Override public void onHammerEvent(NativeHammerEvent event) {

event.preventDefault(); event.preventNativeDefault();

Element target = event.getTarget();

switch (event.getType()) {

case TAP:

//do something on tap

break;

case DRAG:

default:

//do something on drag break;

}

} }, EventType.TAP, EventType.DRAG);
```

Unregister hammer events

```
//remove drag event of previously created HammerTime instance

HammerGWT.off(hammerTime, EventType.DRAG, hammerHandler);
```

Change initial setting of hammer GWT

```
HammerTime hammerTime3 = HammerGWT.create(myWidget);

//          Set          hammer          gwt          options
hammerTime3.setOption(GestureOptions.HOLD_TIMEOUT, 2); //set hold
time out default to 2 sec

// disable drag events registration

// so even if drag event is registered will not be handled by hammer
hammerTime3.setOption(GestureOptions.DRAG, false); n
```

Create Hammer GWT widget

```
// to create hammer gwt widget you should extend HammerWidget class

public class HammerGWTEExampleWidget extends HammerWidget {

public HammerGWTEExampleWidget() {

super();

}

@Override

public Widget createWidget() {

//create gwt widget SimplePanel panel = new SimplePanel();

panel.setSize("100px", "200px");

panel.getElement().setId("hammerWidget"); return panel; } }

// create instance of hammer gwt widget

HammerGWTEExampleWidget      myHammerWidget      =      new
HammerGWTEExampleWidget();

//register tap handler myHammerWidget.registerTapHandler(new
HammerTapHandler() {

@Override public void onTap(NativeHammerEvent event) {

//do something on tap }

});

//unregister          tap          event          handler
hammerGWTWidget.unregisterHandler(EventType.TAP);
```