



FREE eBook

LEARNING

gwt

Free unaffiliated eBook created from
Stack Overflow contributors.

#gwt

Table of Contents

About	1
Chapter 1: Getting started with gwt	2
Remarks.....	2
Examples.....	2
Installation.....	2
Create your first web application.....	2
Chapter 2: Events usage with HandlerManager	4
Examples.....	4
Simple Event and EventHandler usage syntax.....	4
Chapter 3: Server side communication with RestyGwt	6
Remarks.....	6
Examples.....	6
Defining and using your REST client.....	6
Tweaking default config.....	7
Credits	8

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [gwt](#)

It is an unofficial and free gwt ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official gwt.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with gwt

Remarks

This section provides an overview of what gwt is, and why a developer might want to use it.

It should also mention any large subjects within gwt, and link out to the related topics. Since the Documentation for gwt is new, you may need to create initial versions of those related topics.

Examples

Installation

Download and unzip the [GWT SDK](#). This contains the core libraries, compiler, and development server that you need to write web applications.

On Windows, extract the files from the compressed folder gwt-2.7.0.zip. On Mac or Linux, you can unpack the package with a command like:

```
unzip gwt-2.7.0.zip
```

The GWT SDK doesn't have an installer application. All the files you need to run and use the SDK are located in the extracted directory.

Also, you need to have the the [Apache ant](#) installed on your system in order to be able to run the web application locally. On mac you can install it using following command. It installs the apache using mac port.

```
sudo port install apache-ant
```

Create your first web application

GWT ships with a command line utility called `webAppCreator` that automatically generates all the files you'll need in order to start a GWT project. It also generates Eclipse project files and launch config files for easy debugging in GWT's development mode.

You can create a new demo application in a new `MyWebApp` directory by running `webAppCreator`:

Windows

```
cd gwt-2.7.0
webAppCreator -out MyWebApp com.mycompany.mywebapp.MyWebApp
```

Mac or Linux

```
cd gwt-2.7.0
```

```
chmod u+x webAppCreator
```

```
./webAppCreator -maven -out MyWebApp com.mycompany.mywebapp.MyWebApp
```

The `webAppCreator` script will generate a number of files in `MyWebApp/`, including some basic “Hello, world” functionality in the class `MyWebApp/src/com/mycompany/mywebapp/client/MyWebApp.java`.

Read [Getting started with gwt online](https://riptutorial.com/gwt/topic/1014/getting-started-with-gwt): <https://riptutorial.com/gwt/topic/1014/getting-started-with-gwt>

Chapter 2: Events usage with HandlerManager

Examples

Simple Event and EventHandler usage syntax

HandlerManager providing:

```
public class HandlerManagerProvider {  
  
    private static HandlerManager handlerManager;  
  
    private HandlerManagerProvider() {  
  
    }  
  
    public static HandlerManager get() {  
        return handlerManager != null ? handlerManager : (handlerManager = new  
HandlerManager(null));  
    }  
}
```

EventHandler:

```
import com.google.gwt.event.shared.EventHandler;  
  
public interface CustomEventHandler extends EventHandler {  
  
    void doMyAction(CustomEvent event);  
  
}
```

Event:

```
import com.google.gwt.event.shared.GwtEvent;  
  
public class CustomEvent extends GwtEvent<CustomEventHandler> {  
  
    public static final Type<CustomEventHandler> TYPE = new Type<>();  
  
    @Override  
    public Type<CustomEventHandler> getAssociatedType() {  
        return TYPE;  
    }  
  
    @Override  
    protected void dispatch(CustomEventHandler) {  
        handler.doMyAction(this);  
    }  
  
}
```

Handling event (code snippet):

```
import com.google.gwt.event.shared.HandlerManager;
import com.google.gwt.event.shared.HandlerRegistration;

[...]

private List<HandlerRegistration> registrations = new ArrayList<>();
private final HandlerManager handlerManager = HandlerManagerProvider.get();

private void bind() {
    registrations.add(handlerManager.addHandler(CustomEvent.TYPE, new CustomEventHandler() {

        @Override
        public void doMyAction(CustomEvent event) {
            //Action after event firing
        }
    }));
}
```

Firing event (code snippet):

```
private final HandlerManager handlerManager = HandlerManagerProvider.get();

public void fireEvent() {
    handlerManager.fireEvent(new CustomEvent());
}
```

Cleaning registrations after e.g. some window destroy (event handling for this window is no longer needed):

```
private void clearRegistrations() {
    for (HandlerRegistration registration : registrations)
        registration.removeHandler();
    registrations.clear();
}
```

Read Events usage with HandlerManager online: <https://riptutorial.com/gwt/topic/7275/events-usage-with-handlermanager>

Chapter 3: Server side communication with RestyGwt

Remarks

As RestyGWT is a [GWT generator](#), it will automatically create the implementation of your client interface. Under the hood it will use the [RequestBuilder class](#) to implement your calls.

Examples

Defining and using your REST client

You should already have your backend REST resource available. On the client side (GWT) your need to

1. Add RestyGwt dependency to your project with maven

```
<dependency>
  <groupId>org.fusesource.restygwt</groupId>
  <artifactId>restygwt</artifactId>
  <version>2.2.0</version>
</dependency>
```

2. Add the inheritance to your module file

```
<inherits name="org.fusesource.restygwt.RestyGWT"/>
```

3. Create your client interface

```
public interface PizzaService extends RestService {
    @POST
    @Path("pizzaorders")
    public void order(PizzaOrder request,
                    MethodCallback<OrderConfirmation> callback);
}
```

4. Use your client where you want in you app

```
PizzaService service = GWT.create(PizzaService.class);
service.order(order, new MethodCallback<OrderConfirmation>() {

    public void onSuccess(Method method, OrderConfirmation response) {
        //code your stuff here
    }

    public void onFailure(Method method, Throwable exception) {
        //code your stuff here
    }
});
```


Tweaking default config

Default config is configured in `import org.fusesource.restygwt.client.Defaults;`

1. Service root. By default, RestyGWT will use module name to get rest service root, to change it, call on module load:

```
Defaults.setServiceRoot("/rest/");
```

2. Date format. By default, RestyGWT will send unix timestamp as long for `java.util.Date`. If you don't want to do date processing on client side and just send a string, use

```
Defaults.setDateFormat("yyyy-MM-dd'T'HH:mm");
```

Read Server side communication with RestyGwt online:

<https://riptutorial.com/gwt/topic/4103/server-side-communication-with-restygwt>

Credits

S. No	Chapters	Contributors
1	Getting started with gwt	Community , denis , Emil Burzo
2	Events usage with HandlerManager	null
3	Server side communication with RestyGwt	Denis Tulskiy , Ronan Quillevere