**TuitTuit**

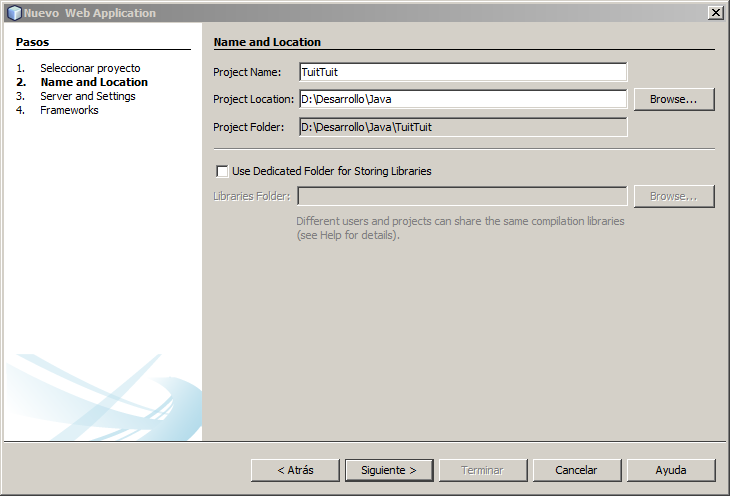
Cliente Twitter

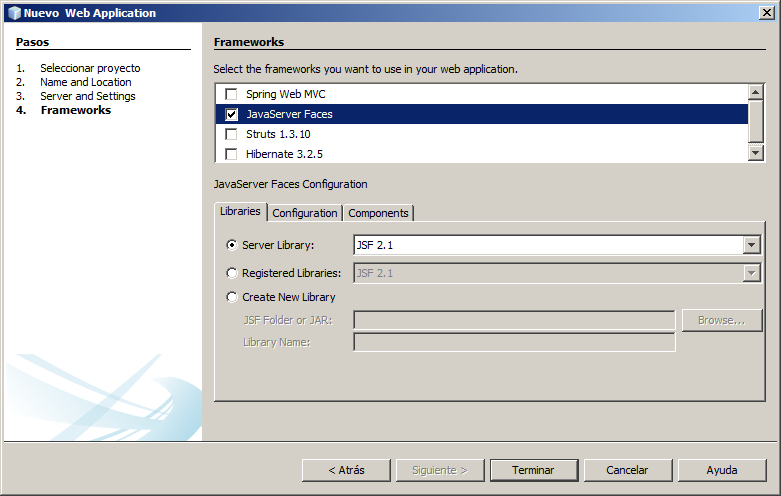
Muga Ampuero Luis

Salinas Francia Antonio

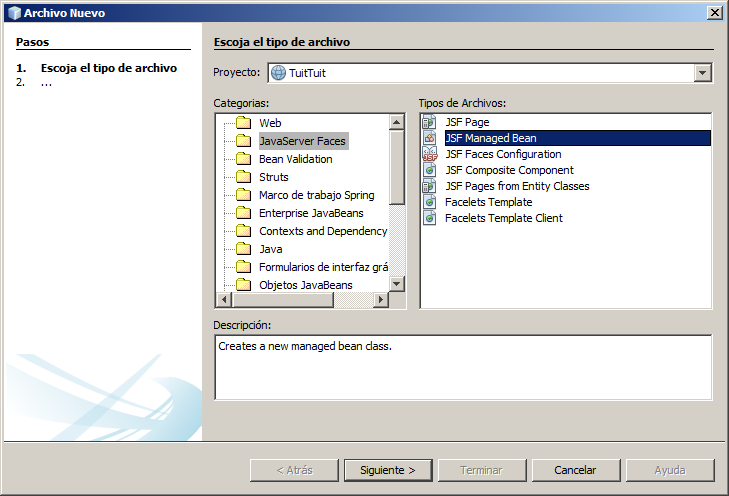
**I. Creando el Proyecto**

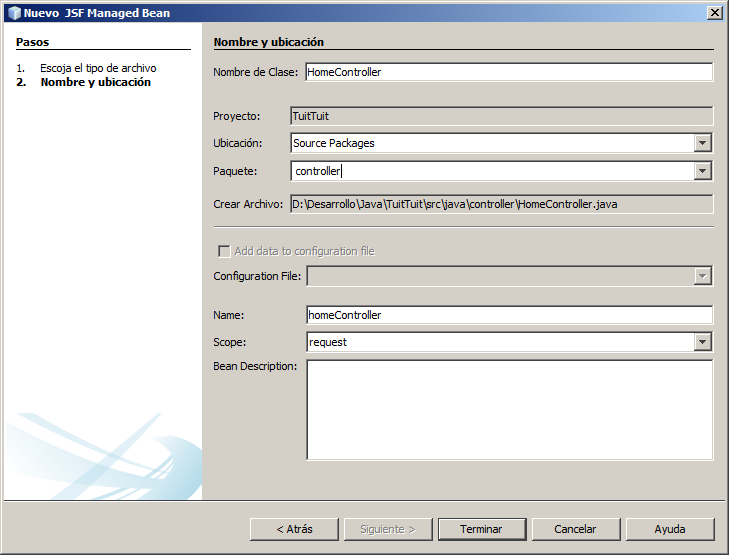
1. Seleccionamos File > New Project -> Java Web -> Web Application project.
2. Nombraremos al proyecto TwitterSwingClient. Y seleccionamos su ruta para almacenarlo.





1. Creamos nuestro controlador.

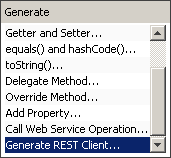




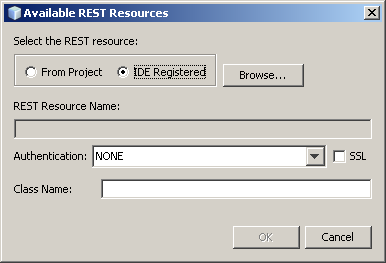
**II. Implementando el Web Service**

1. En nuestro HomeController.java.
2. Presionamos Alt-Insert el menú de insertar código aparecerá.

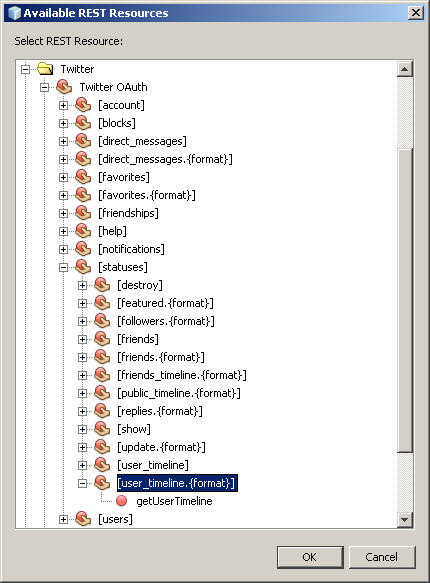
**Importante:** Si la opción Generate REST Client no aparece, entonces el soporte “Java EE support” no está instalado o activado. Vaya a Tools >Plugins, abra la pestaña instalación y verificar que “Java Web y EE support” se encuentre activado.



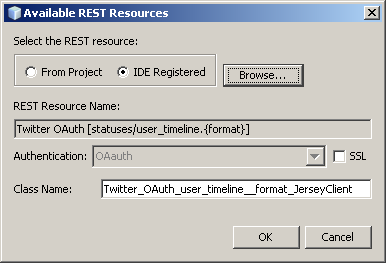
1. Clic en Generate REST Client. Se abrirá el siguiente cuadro.



1. Seleccionamos IDE Registered y hacemos clic en Browse. Navegamos hacia Twitter > Twitter OAuth> [statuses] > [user\_timeline.{format}]. Clic en OK.



1. Ahora se muestran los cuadros de dialogo disponibes para TwitterOAuthuser\_timeline, el nombre de la clase y el tipo de autentificación OAuth. Clic OK.

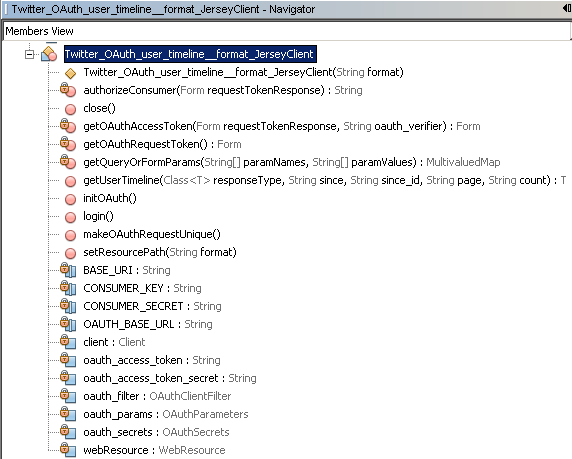


1. Se abre un cuadro de dialogo preguntando si deseamos generar los objetos Java desde el esquema XML en el archivo WADL file. Clic en Yes.
2. Al final de la clase, el IDE generó una clase interna llamada Twitter\_OAuth\_user\_timeline\_\_format\_JerseyClient.

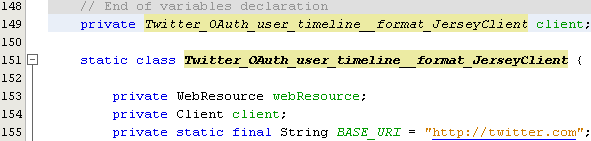
Esta clase contiene lo siguiente:

* + CONSUMER\_KEY: Clave del cliente
  + CONSUMER\_SECRET: Cadena secreta del cliente
  + initOAuth(): Método para la inicialización OAuht
  + getUserTimeline(): MétodoHTTP: getUserTimeline
  + makeOAuthRequestUnique(): Múltiples APIS para inicio de sesión
  + login: Permite loguear en cuentas de Twiter.

Aquí se puede ver la estructura de clases.



1. Ingresamos el siguiente código después de la clase Twitter\_OAuth\_user\_timeline\_\_format\_JerseyClient,



1. Ubicamos en un método. Sobre este método, creamos uno Nuevo llamado initUserInfo que arroja un MalformedURLException y un IOException.

private void initUserInfo() throws MalformedURLException, IOException {

}

Insert the following code into the method body of initUserInfo. Comments in the code explain what the code does.

private void initUserInfo() throws MalformedURLException, IOException {

//Create an instance of the internal service class

client = new Twitter\_OAuth\_user\_timeline\_\_format\_JerseyClient("xml");

//Log in, get tokens, and append the tokens to the consumer and secret

//keys

client.login();

client.initOAuth();

//Call getUserTimeline, get a list of statuses, pass the most recent

//status as a StatusTypeobject, and display the text of that object

//in the JTextField

Statuses statuses = client.getUserTimeline(Statuses.class, null, null, null, "1");

StatusTypest = statuses.getStatus().get(0);

jTextField1.setText(st.getText().trim());

//Get a UserType object from the StatusType object, get the URL of that

//user's icon, and display that icon in the JLabel

UserType user = st.getUser();

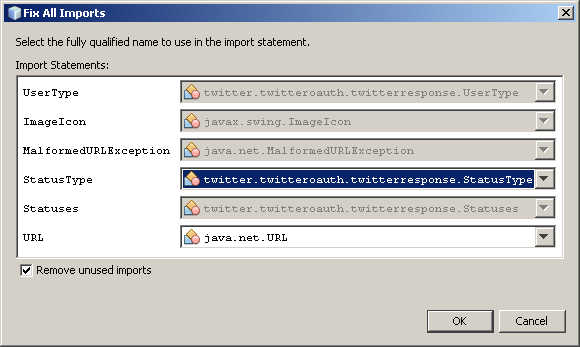
String iconSrc = user.getProfileImageUrl();

URL iconUrl = new URL(iconSrc);

ImageIcon icon = new ImageIcon(iconUrl, user.getScreenName());

jLabel1.setIcon(icon);

}

1. Abrimos el cuadro de Importar y seleccionamos twitter.twitteroauth.twitterresponse.StatusTypey.
2. Agregamos un bloque try/catch al constructor TwitterJForm para llamar a initUserInfo cuando la aplicación inicie.

publicTwitterJFrame() {

initComponents();

try {

initUserInfo();

} catch (IOException ex) {

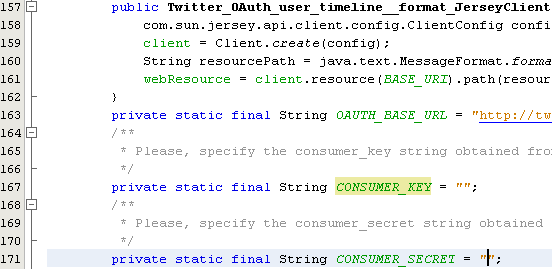
Logger.getLogger(TwitterJFrame.class.getName()).log(Level.SEVERE, null, ex);

}

}

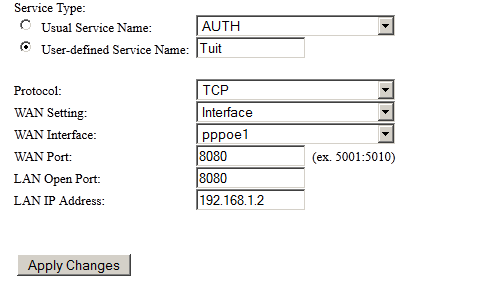
**III. Obteniendo las llaves OAuth de Twitter**

1. En el navegador web ir a la página [Twitter>Applications](http://twitter.com/apps)y [Register a new application](http://twitter.com/apps/new). Se necesita estar logueado en una cuenta.
2. Ingresamos el nombre de la aplicación que debe ser única en Twiter.
3. Ingresamos una descripción de nuestra aplicación.
4. Ingresamos una URL para nuestra aplicación.
5. En el tipo de aplicación seleccionamos Cliente.
6. En el tipo de acceso seleccionamos lectura y escritura.
7. Copiamos el Consumerkey del navegador. En el IDE buscamos la línea CONSUMER\_KEY y colocamos el valor dentro de las comillas.



**Anexo 1: Publicando el proyecto a través de NAT**

1. Ingresamos a nuestros router y generamos una regla de NAT para poder publicar nuestra aplicación en Internet, en nuestro caso nuestra PC donde se encuentra nuestro proyecto es la IP 192.168.1.2 y el puerto 8080.



**Anexo 2: Creando un redireccionamiento en un dominio**

1. Dentro de la configuración de nuestro dominio, nos dirigimos a la creación de una redirección y colocamos la dirección completa de nuestra aplicación para poder ingresar a ella de manera más cómoda.

