GOOGLE AUTHENT

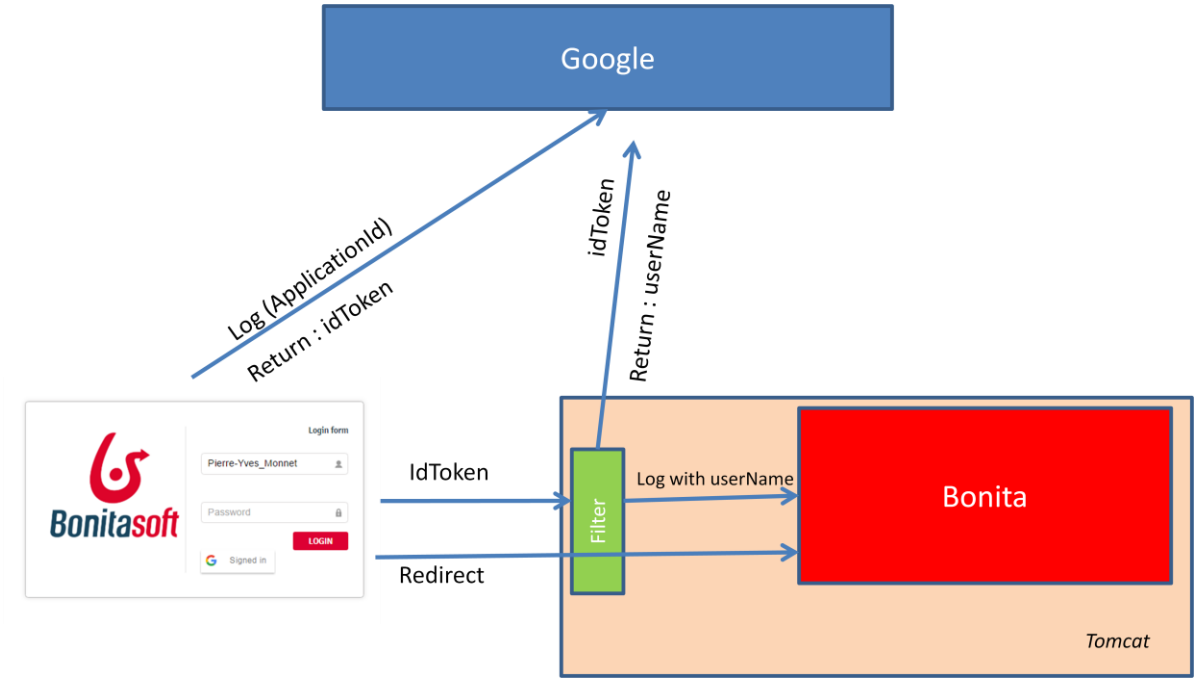
March 21 2016

SOMMAIRE

DETAILS

1. Principle

The Google Authentication principle is the following:



1/ On the login page, a new button Google Sign In is present. When the user clicks on, the browser asks the Google identification. This information is send to Google, which return an idToken. A JavaScript “OnSignIn” behind the button, in case of success, call an URL to the Bonita Portal, which contains the idToken.

Note: the Google Script embedded in the page check if the user is already connected. If this is the case, then the browser automatically call the OnSignIn() function : the login page blink and the redirection is automatic.

2/ a Filter are defined on the Tomcat level. This filter checks if the browser is already connected on the Bonita Portal. This is not the case on the first call. Then the filter uses the IdToken to call Google API. Google API returns all information (user name, email, etc…). Filter use the UserName to connect to the Bonita system.

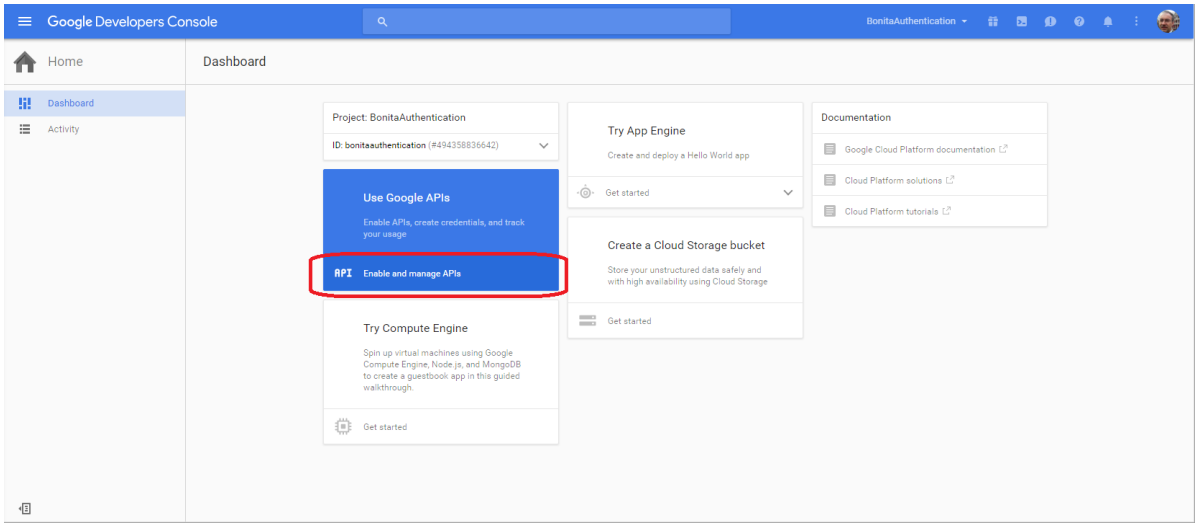
3/ the browser receive a “success”, then the onSignIn() Javacript play a redirect URL to the Portal page. A new URL is sent to the BonitaPortal, intercepted by Filter, but because user is already connected, filter does nothing.

2. Create a google Application

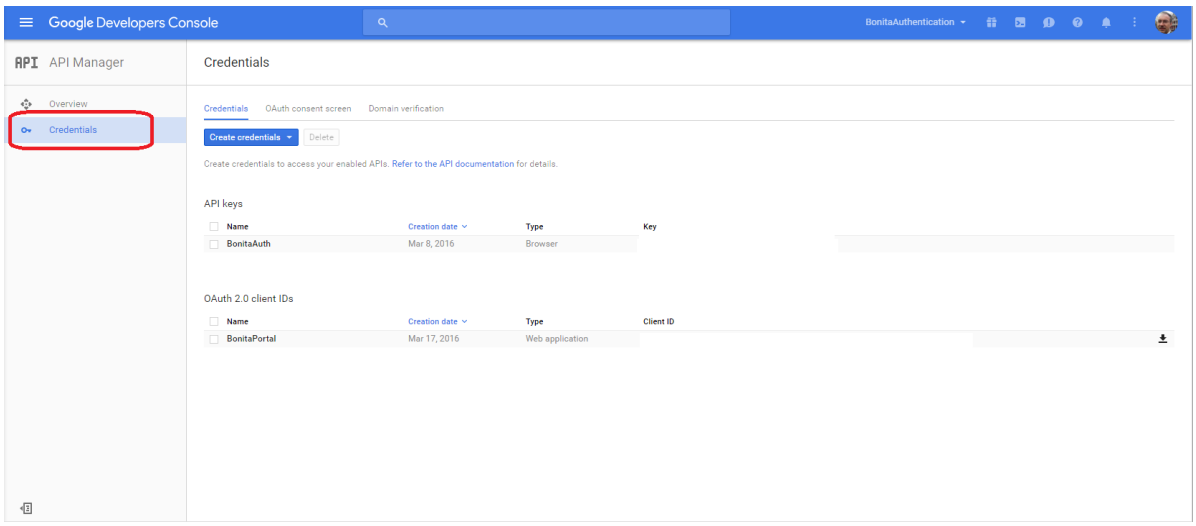
To let the Api connect, a Application ID must be give.

Go to https://console.developers.google.com/?hl=FR&pli=1

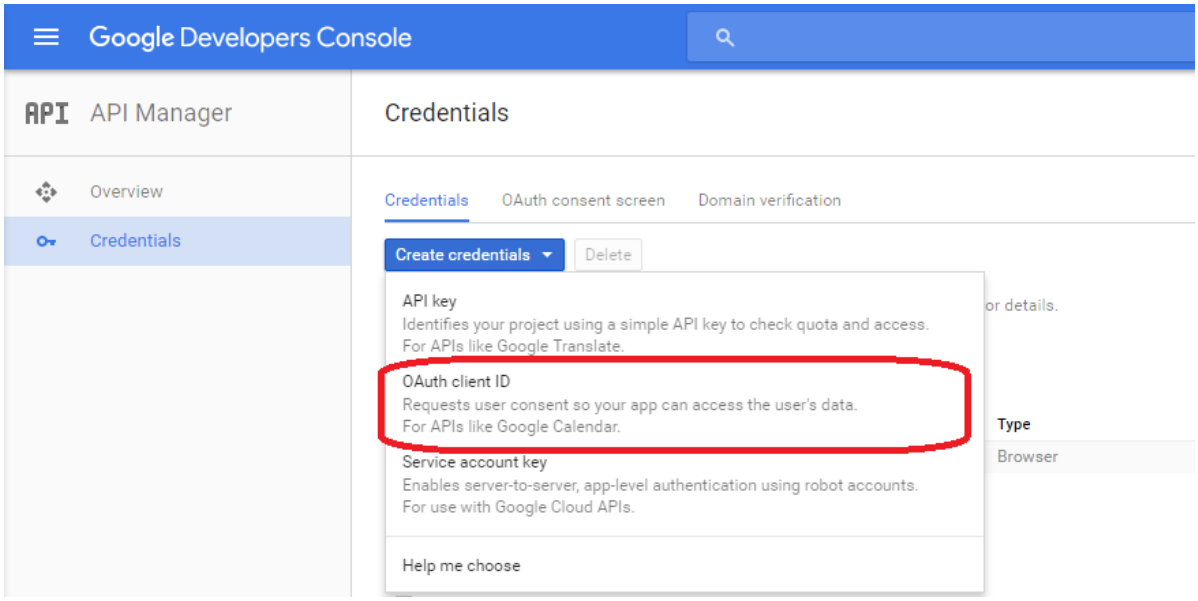
Connect with your Google ID



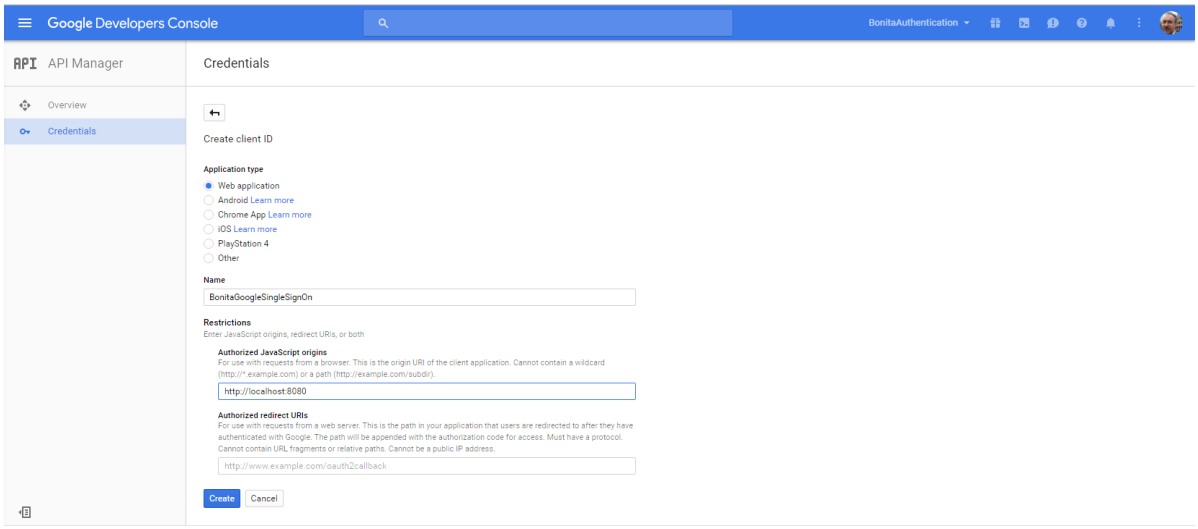
And create a project by clicking on Credential



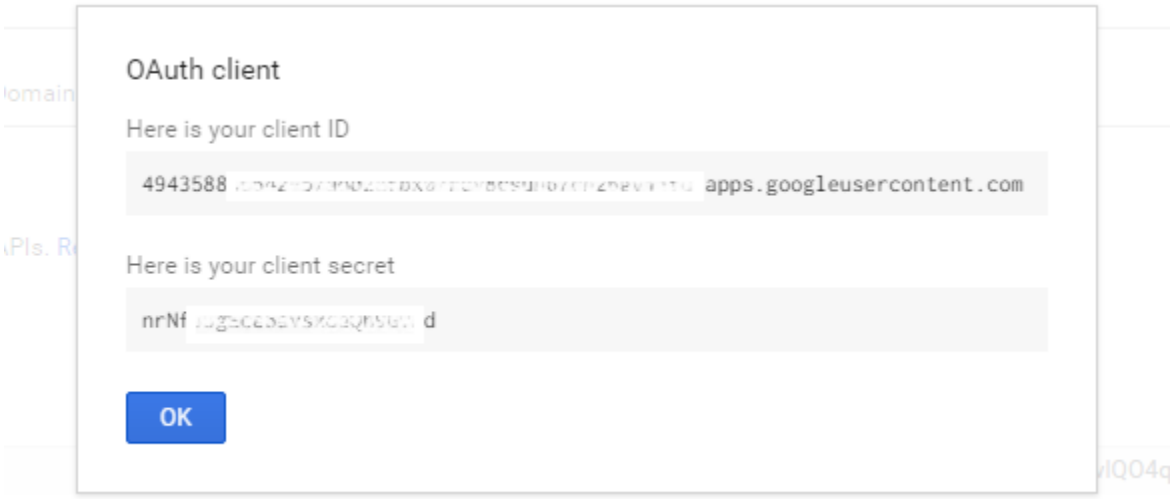
Choose a new Credential “OAuth Client ID”



Choose the parameters “Web Application”, then give a name. End, register each URL which will access the login page.



Then, Google API gives you back the ClientID



3. Change the login page

You can create your own login page, or modify the Bonitasoft page.

Note: if you change the Bonitasoft page, you should apply the change after each Bonita Upgrade.

In the login Page, add just BEFORE the </head> balise (you have to place here your KEY)

```
<meta name="google-signin-client id" content="HERE THE CLIENT KEY">
<script src="https://apis.google.com/js/platform.js" async defer></script>
<script>
function onSignIn(googleUser) {
  var form = $('#LoginForm');
  var actionUrlElement = form.attr('action');
  var profile = googleUser.getBasicProfile();
  console.log('ID                : ' + profile.getId()); // Do not send to your backend! Use
an ID token instead.
  console.log('Name              : ' + profile.getName());
  console.log('Image URL         : ' + profile.getImageUrl());
  console.log('Email             : ' + profile.getEmail());
  console.log('actionUrlElement: ' + actionUrlElement);
  var id_token = googleUser.getAuthResponse().id_token;
  console.log("ID Token: " + id_token);

  /*  .get('../bonita/portal/homepage?idtokengoogle='+id token,function(data,status) {
alert("get it"); }) */
  var dataLogin={}
  dataLogin.idtokengoogle = googleUser.getAuthResponse().id_token;
  dataLogin.namegoogle=profile.getName();
  if (actionUrlElement == null)
  {

 $.post("../bonita/portal/homepage", dataLogin, function(result){
   window.location.replace('../bonita');
 });
   }
  else {
 redirectUrl="../bonita";
 var redirectUrlPos = actionUrlElement.indexOf("redirectUrl=");
 if (redirectUrlPos!=-1)
 {
   redirectUrl = actionUrlElement.substring(redirectUrlPos+"redirectUrl=".length );
   // string is like /bonita/portal.... so the final should be ../bonita
   redirectUrl = ".."+decodeURIComponent( redirectUrl );
 }

 $.post(actionUrlElement, dataLogin, function(result){
   window.location.replace(redirectUrl);
 } );
 }

}
</script>
```

And then in the page, add

```
  <div class="g-signin2" data-onsuccess="onSignIn"></div>
```

For example, we add it just after the Login button

4. Server side

On the server side, the filter has to be added and configured.

Place the following JAR file in <TOMCAT>/webapps./bonita/WEB-INF/lib:

And the filter library

```
FilterGoogleAuthent-1.0.0.jar
```

and

```
google-api-client-1.19.0.jar
google-api-client-gson-1.21.0.jar
google-api-services-plus-v1-rev137-1.19.0.jar
google-http-client-1.21.0.jar
google-http-client-gson-1.21.0.jar
```

```
google-http-client-jackson2-1.19.0.jar
google-oauth-client-1.19.0.jar
guava-jdk5-13.0.jar
```

Theses jars are fetch form the MAVEN dependency

```
<dependency>
    <groupId>com.google.apis</groupId>
    <artifactId>google-api-services-plus</artifactId>
    <version>v1-rev137-1.19.0</version>
</dependency>
<dependency>
    <groupId>com.google.api-client</groupId>
    <artifactId>google-api-client-gson</artifactId>
    <version>1.21.0</version>
</dependency>
```

Then configure the web.xml under <TOMCAT>/webapps./bonita/WEB-INF.

ATTENTION: the filter mapping part must be place at the beginning of the filter mapping part. Filter mapping order is very important, and the Google Filter mapping must be executed in first.

```
<filter>
   <filter-name>GoogleFilter</filter-name>
   <filter-class>com.bonitasoft.googleauthent.FilterGoogle</filter-class>
   <init-param>
         <param-name>googleServerClientId</param-name>
         <param-value>494358836642-
hnu0gufcrur2tupb2cq4cgigc5113g00.apps.googleusercontent.com</param-value>
     </init-param>


     <init-param>
         <param-name>technicalsUsers</param-name>
         <param-value>
user|patrick.gardenier,user|walter.bates,group|/acme/sales,role|qualityManager,profile|t3
         </param-value>
     </init-param>

    <init-param>
         <param-name>technicalLoginPassword</param-name>
         <param-value>walter.bates/bpm</param-value>
     </init-param>

     <init-param>
         <param-name>acceptClassicalLogin</param-name>
         <param-value>true</param-value>
     </init-param>

   <init-param>
         <param-name>bonitaPassword</param-name>
         <param-value>bpm</param-value>
     </init-param>
     <init-param>
         <param-name>log</param-name>
         <param-value>true</param-value>
     </init-param>
    <init-param>
         <param-name>ping</param-name>
         <param-value>true</param-value>
     </init-param>
  </filter>


  <filter-mapping>
    <filter-name>GoogleFilter</filter-name>
    <url-pattern>/portal/*</url-pattern>
  </filter-mapping>
  <filter-mapping>
    <filter-name>GoogleFilter</filter-name>
    <url-pattern>/loginservice</url-pattern>
  </filter-mapping>
<filter-mapping>
   <filter-name>GoogleFilter</filter-name>
   <url-pattern>/login.jsp</url-pattern>
  </filter-mapping>
```

Filter parameters are:

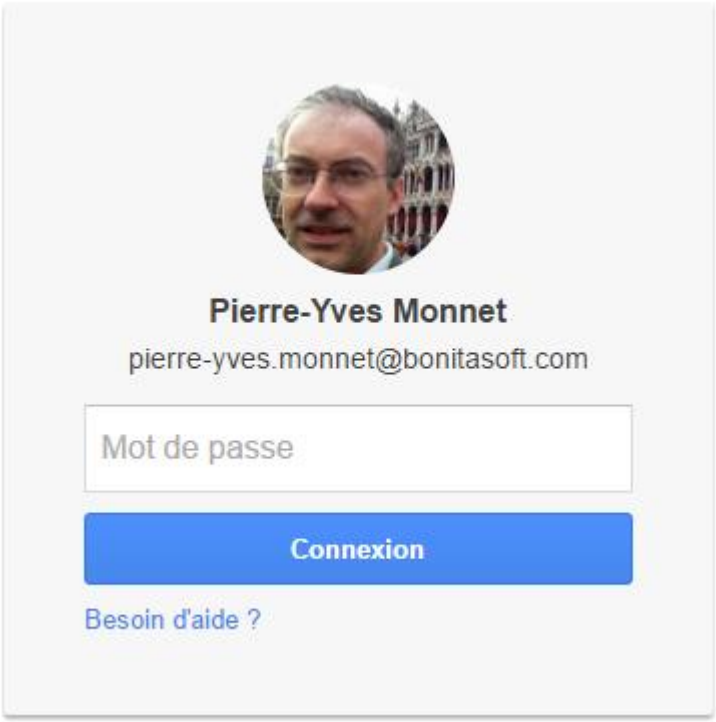
Parameter	Explanation
googleServerClientId	The client id, the same you put in the Login page
acceptClassicalLogin	True/false. If true, if a classical login request arrived, the filter accept it and let Bonita do it's job. Then, you can still connect to the portal with some users which are not link to a google engine
technicalUsers	Some user you don't want the filter manage, like the user install, or any user you created to let your application connect to the engine by the API or the RESTAPI. List separate by a comma. Format is user|<10sername>,group|<grouppath>,role|<roleName>,profile|<profilename>
technicalLoginPassword	To calculate the technicalUser, an access is necessary to Bonita. Format is <username>/<password>
bonitaPassword	When a user is recognize and validated by Google, the filter will connect it to Bonita. Doing this, its need to use a password. Specify it here
Log	True/false. If false, only init parameters and error are logs.
Ping	Allow the filter to give you a PING page on the url bonita/portal/filtergoogleping

Note: if you accept the classicalLogin, the technicalUsers list has no meaning.

5. How to configure an user

In order to match the Google authentication and the Bonita User, you have to create the user in Bonita with some rule:

- Use the Google Name. The google name appears when you log : for example, mine is « pierre-Yves Monnet »



- Bonita does not accept space, so replace them by a _. My Bonita Login name is for example Pierre-Yves_Monnet
- The Filter can't access the Google password and need to use one to connect to Bonita. So, use a "fake password" for all users. Register the fake password in the Filter. Do not use "bpm", well known
- Protect the BonitaPassword : if you accept the "acceptClassicalLogin", that means user can connect with the Google name and the fake password. Protect this password.

HEADQUARTERS

GRENOBLE, FRANCE

32, rue Gustave Eiffel

38000 Grenoble

EMEA, ASIA & LATIN AMERICA

PARIS, FRANCE

73-77, rue de Sèvres

92100 Boulogne-Billancourt

NORTH AMERICA

SAN FRANCISCO, USA

51 Federal St. Suite 305

San Francisco, CA 94107

Bonitasoft

www.bonitasoft.com