# Image content description and faces celebrity recognition

[1. Introduction 2](#_Toc474746159)

[2. Bluemix scenarios to develop and deploy an application 2](#_Toc474746160)

[3. Before start, what will you need? 3](#_Toc474746161)

[4. Get code from GitHub and import it to eclipse 3](#_Toc474746162)

[5. Create visual recognition service 9](#_Toc474746163)

[6. Add Watson Java SDK with dependencies to your project 12](#_Toc474746164)

[7. Run your app 16](#_Toc474746165)

[8. Deploy a standalone java application to Bluemix 22](#_Toc474746166)

[9. Code source documentation 28](#_Toc474746167)

[10. References 28](#_Toc474746168)

## Introduction

Watson visual recognition IBM API uses deep learning algorithms to analyze images and videos. The objective of this tutorial is to investigate and improve the set of built-in classes of Watson visual recognition to better fit the two following use cases:

* ***Image description using the tow mains classes (FacesDetect and VisualClassification)***
* ***Celebrity faces recognition (limited to training sample of celebrity faces since the building of IBM databases)***

More details on Watson visual recognition classes and methods are in the link[http://watson-developer-cloud.github.io/java-sdk/docs/java-sdk3.3.1/com/ibm/watson/developer\_cloud/visual\_recognition/v3/VisualRecognition.html#VisualRecognition-java.lang.String-](http://watson-developer-cloud.github.io/java-sdk/docs/java-sdk3.3.1/com/ibm/watson/developer_cloud/visual_recognition/v3/VisualRecognition.html)

The program built to achieve the both objectives can be designed with the following flowchart (Figure1.):

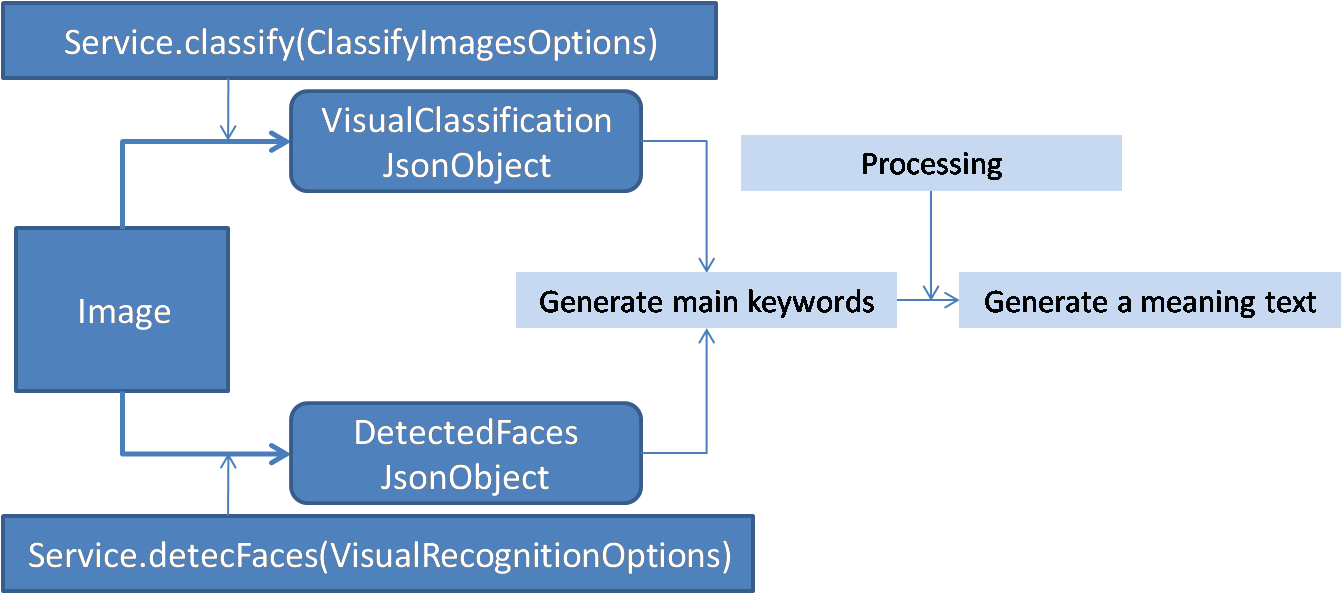


Figure . Flow chart of global program of Image content description

Figure 1. presents the global program that will read an input image and display a text describing its content and can be summarized as follows :

1. ***Input a image***
2. ***Instantiate a visual recognition service***
3. ***Create VisyalRcognitionOptions and ClassifyImagesOptions of image***
4. ***Call detectFaces and classify methods of VisualRecognition service with VisyalRcognitionOptions and ClassifyImagesOptions as arguments respectively and return Jsons objects (DetectedFaces and VisualClassification).***
5. ***Process the two obtained Jsons objects (DetectedFaces and VisualClassification) to display a meaning text describing image content***

To obtain a description of an image and recognize faces celebrity, you should follow the steps below.

## Bluemix scenarios to develop and deploy an application

There are several scenarios to develop, run and deploy an application using Bluemix

1. Create, develop and deploy application using just Bluemix using applications and services(all steps on cloud)
2. Create and run application in local using a Bluemix services : for example, create java application with eclipse or download code from GitHub, add key API and URL endpoint of a services and run application as java application, after deploy it to Bluemix
3. Hybrid scenario (Bluemix cloud and local) : create application on Bluemix(cloud) and import it to local, modify and deploy to Bluemix, etc

**On this tutorial we follow the second scenario**

## Before start, what will you need?

* **Bluemix account,** provide access to create visual recognition service
* **Eclipse IDE LUNA**
* **Java 8**
* [**Cloud Foundry command-line interface**](https://www.ng.bluemix.net/docs/)**(CLI)**

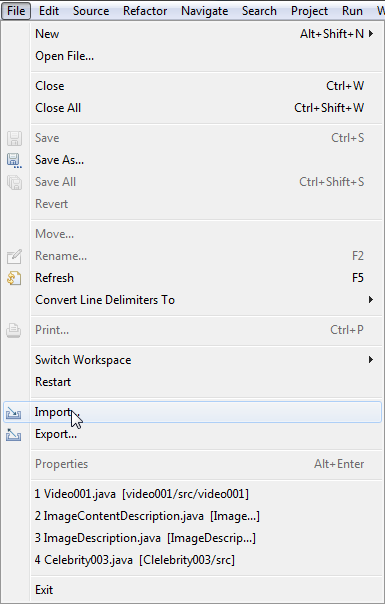
## Get code from GitHub and import it to eclipse

**Step1. Download the code source from Github repository into eclipse**

1. Go to a GitHub repository using this link (link to repository) (waiting invitation to join GitHub IBM team to share the code source used in this tutorial)
2. Click the “ImageContentDescription.war” button to download the WAR file.
3. Save the WAR file under your workspaces
4. Unzip the file in your eclipse workspace

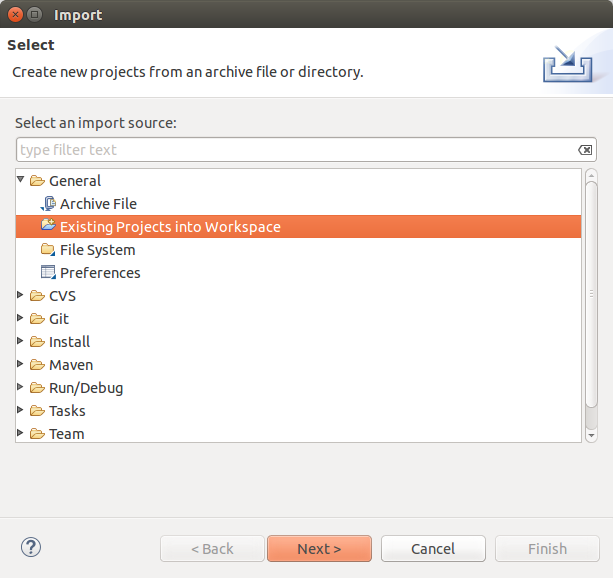
**Step 2.Import the app into your Eclipse workspace**

1. Select **File > Import.**

****

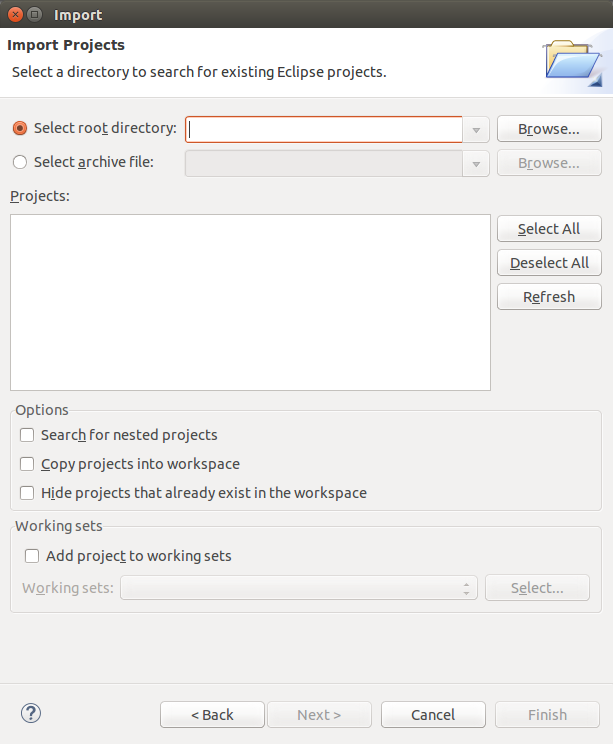
1

1. Then select **General>Existing project into workspace**



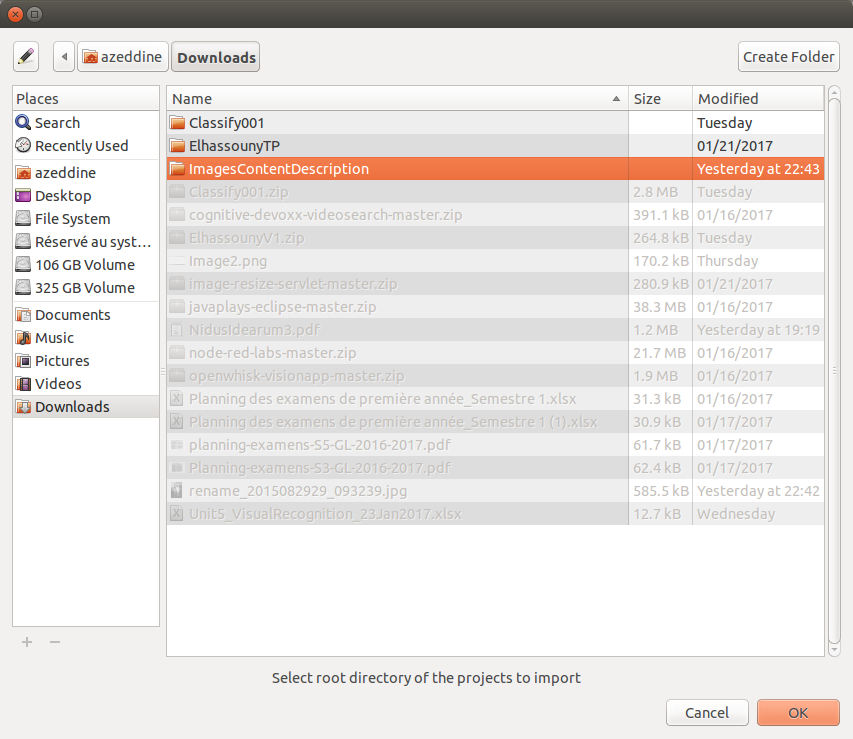
1

1. Click **Browse** button to navigate to your project unzipped

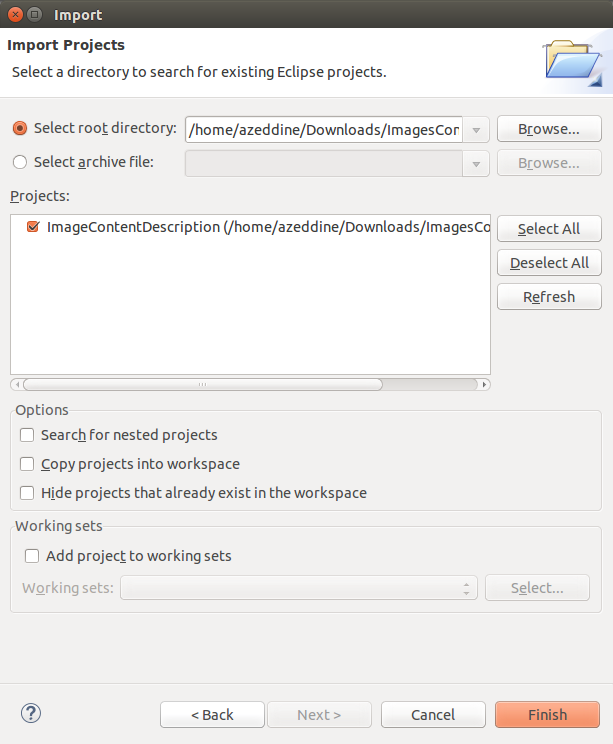
****

1

1. Navigate to your unziped folder (**ImageContentDescription**) and click **Ok button**

****

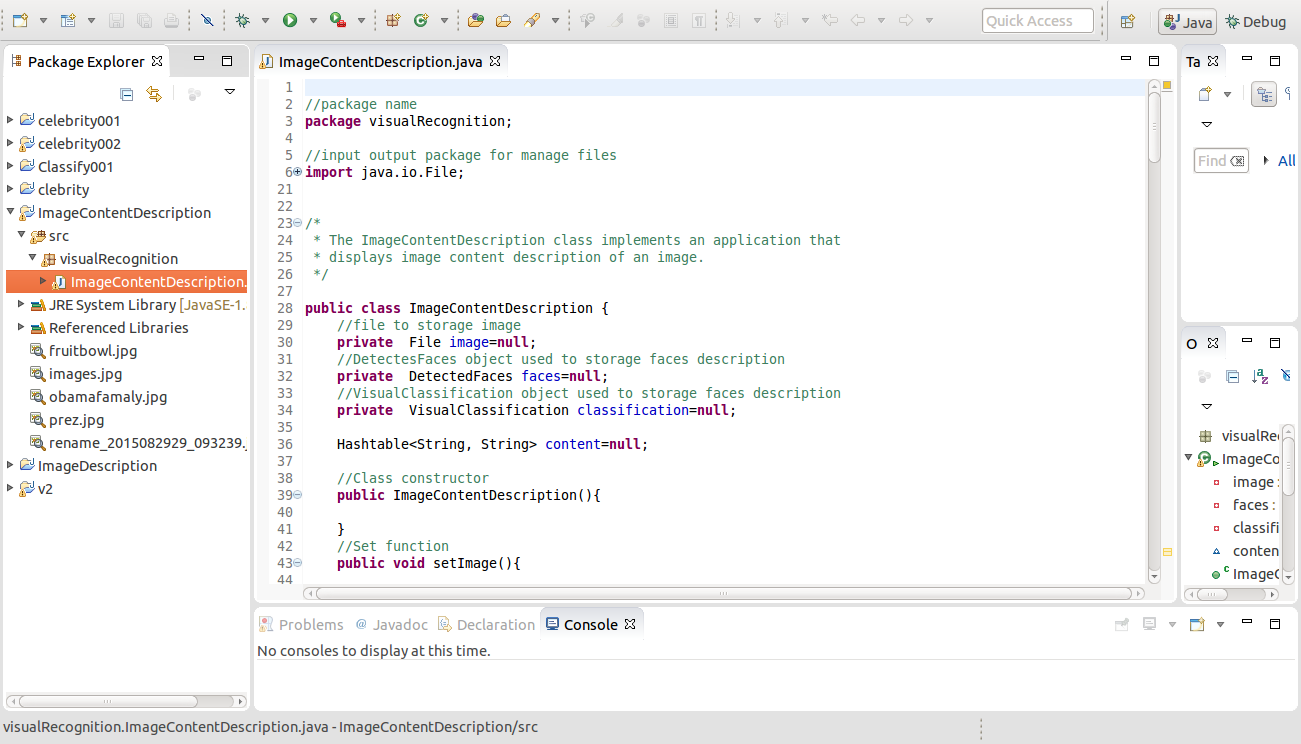
1. Check **ImagesContentDescription option Button** and click **Finish**

****

2

1

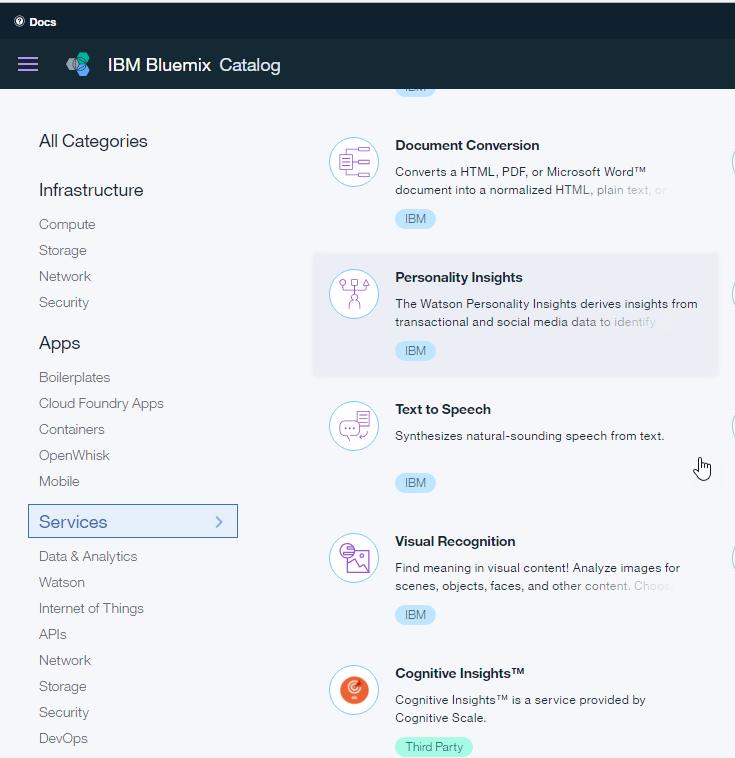
1. Check that the ImagesContentDescription folder is imported and explore its structure (for more details, you can see README.txt)



## Create visual recognition service

**Step 3.Instantiate a visual recognition service using** [**Bluemix Catalog**](https://console.ng.bluemix.net/catalog/)

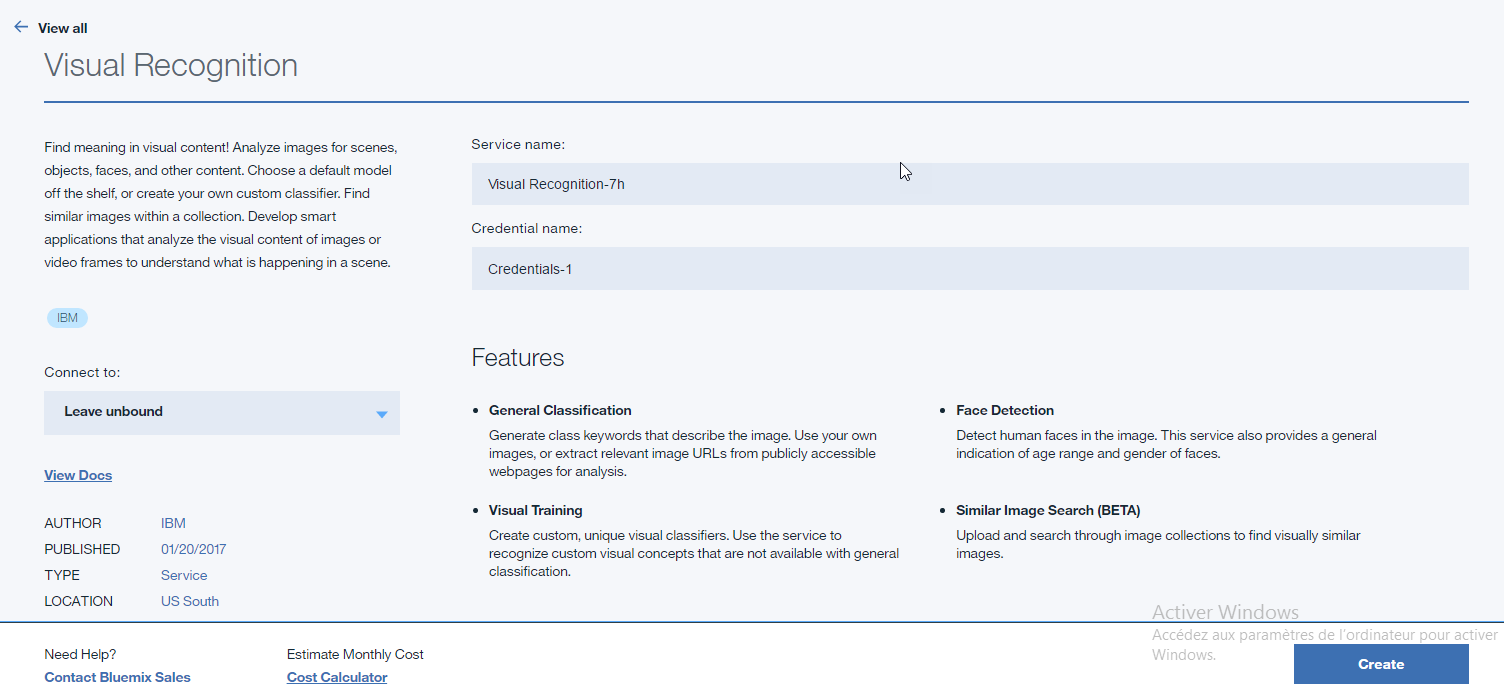
1. Log in to Bluemix<https://console.ng.bluemix.net/> (using your ID\_IBM and password)
2. In the **Catalog** tab, click on **Watson** (Services) and choose the **Visual Recognition** service



1

2

1. Name your service (or let the default name) and click **Create** button

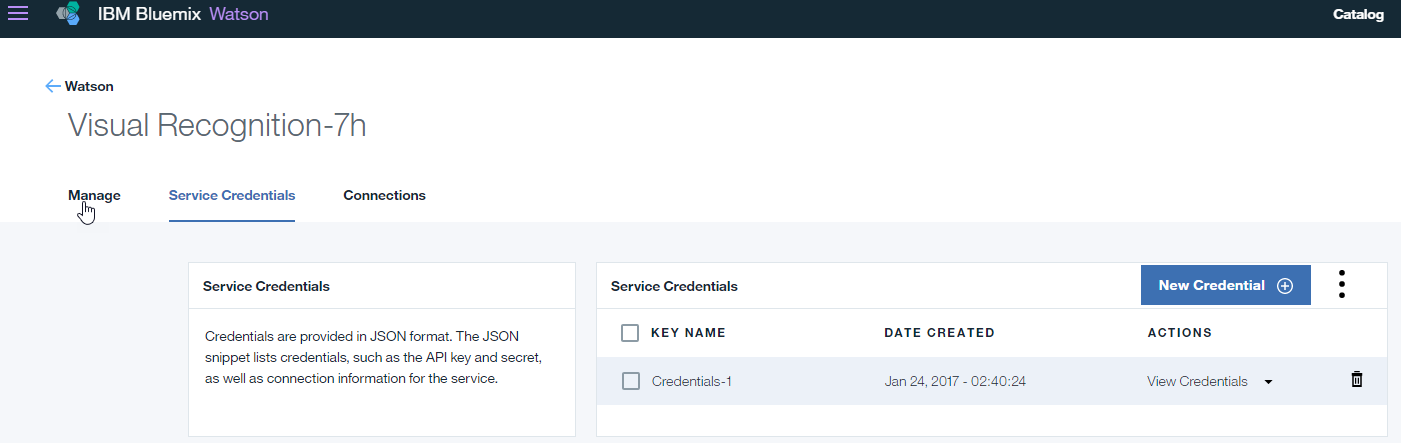
****

3

2

1

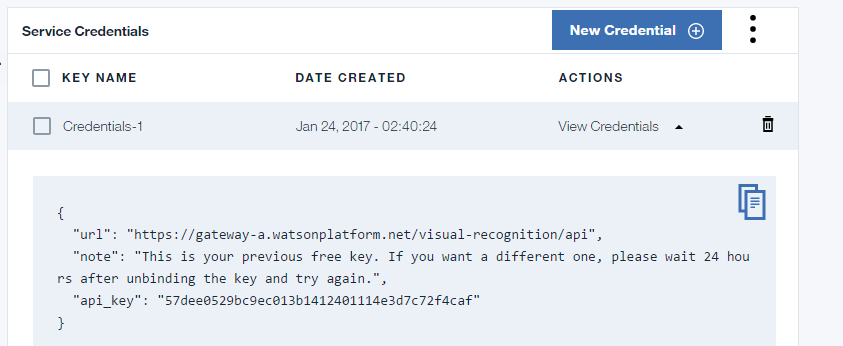
1. Navigate to your **Dashboar and** click on your visual recognition service created



2

1

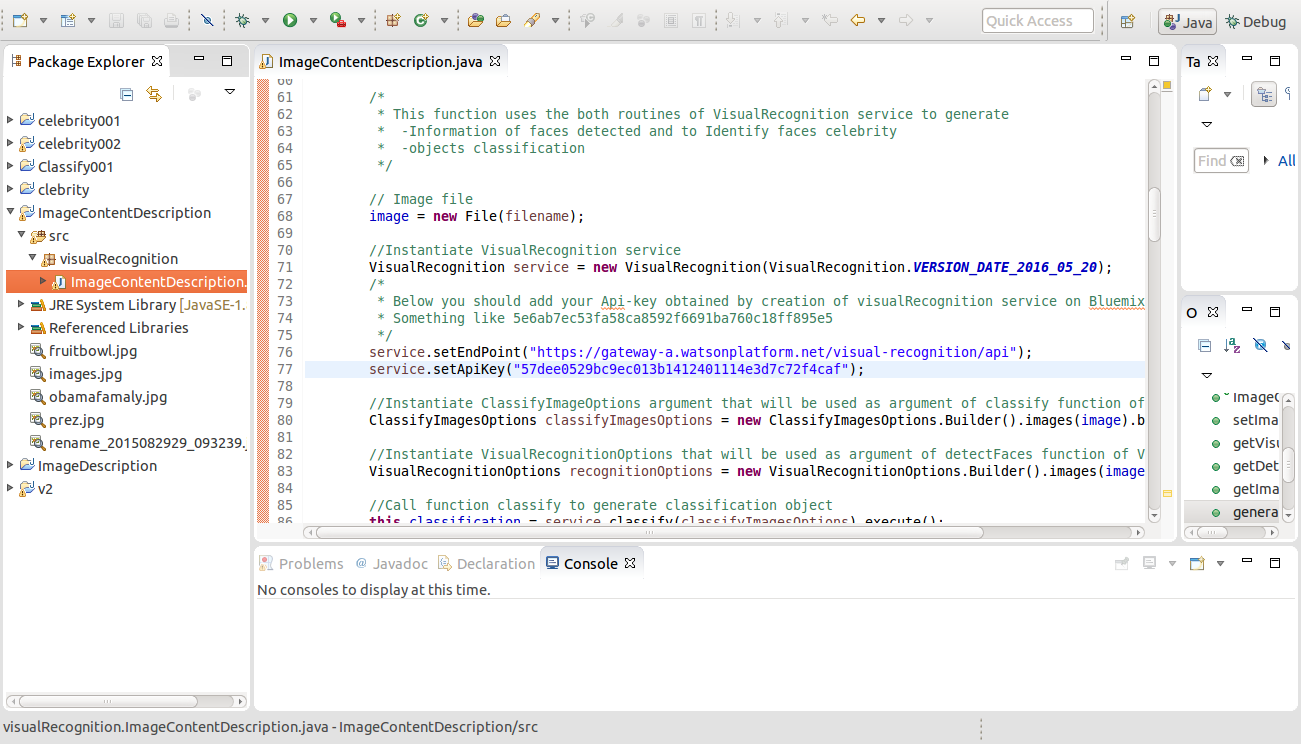
1. Click on the service **credential tab**, to get the ApiKey, click on the **view credentials**

****

1

**Step 4.Input your Visual Recognition Service API Key**

1. Comeback to your project in eclipse, navigate to res/ImageContentDescription.java and set your Visual Recognition Service **API Key** obtained above to service.setApiKey(paste it in selected place below, in the code source).

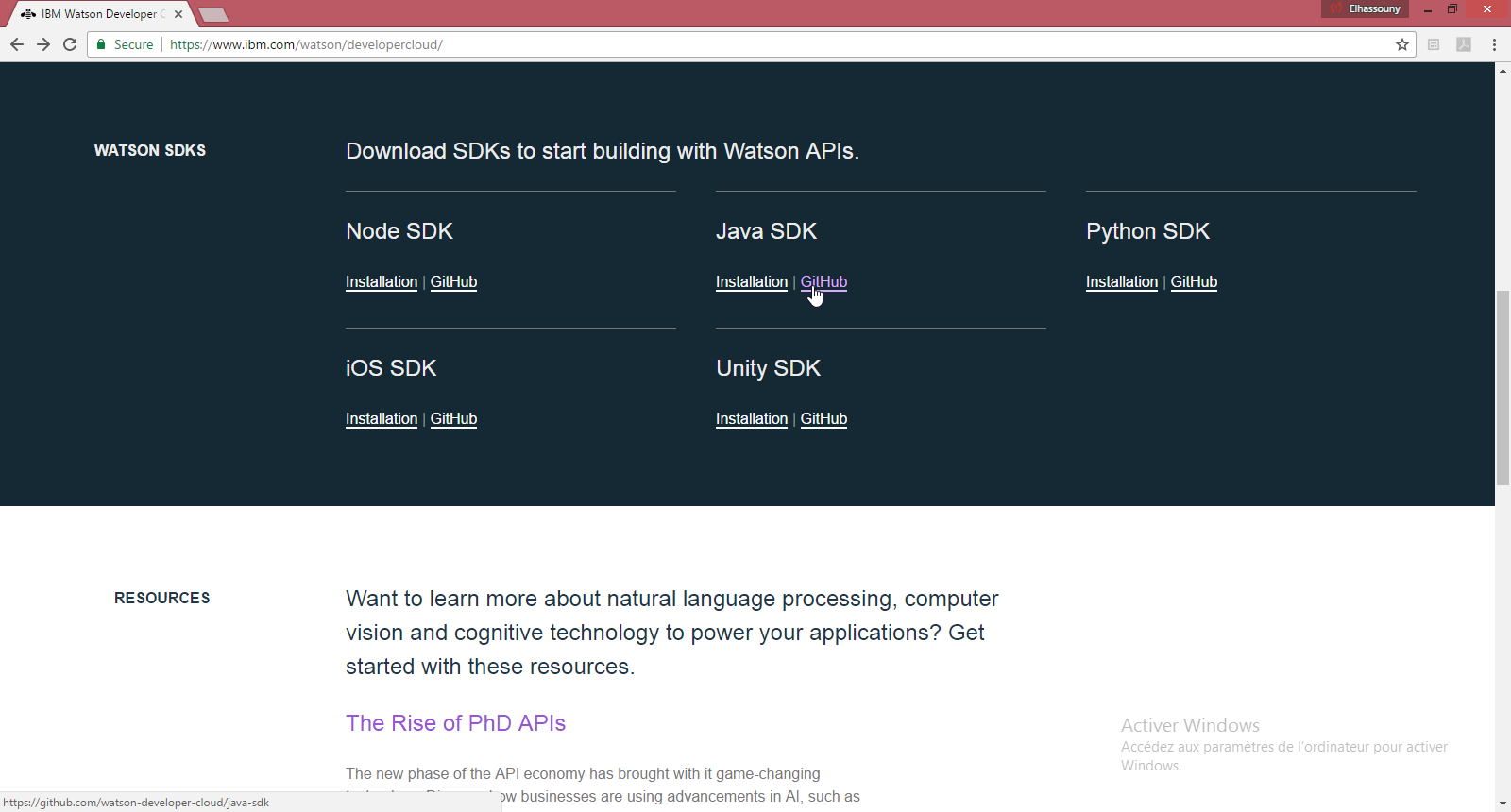


1

## Add Watson Java SDK with dependencies to your project

**Step5. Add Watson Java SDK with dependencies**

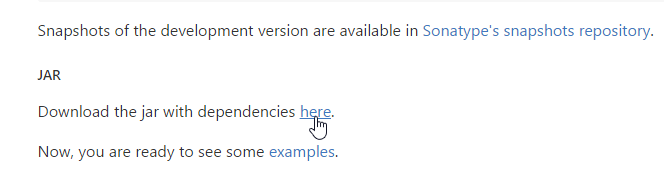
1. Click this link : <https://www.ibm.com/watson/developercloud/>, in the web page, scroll horizontal scrollbar to that part (following image) and click GitHub under Java SDK title



1

1

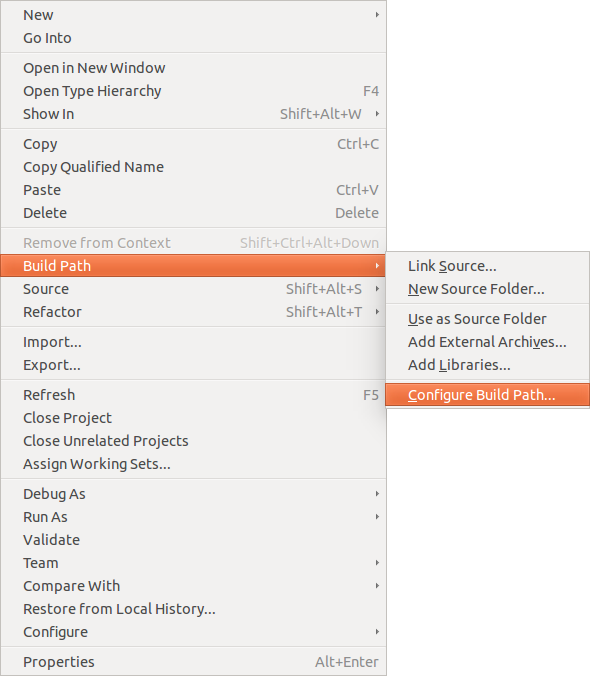
1. Downloads the Watson java **sdk dependencies jar** files using that link



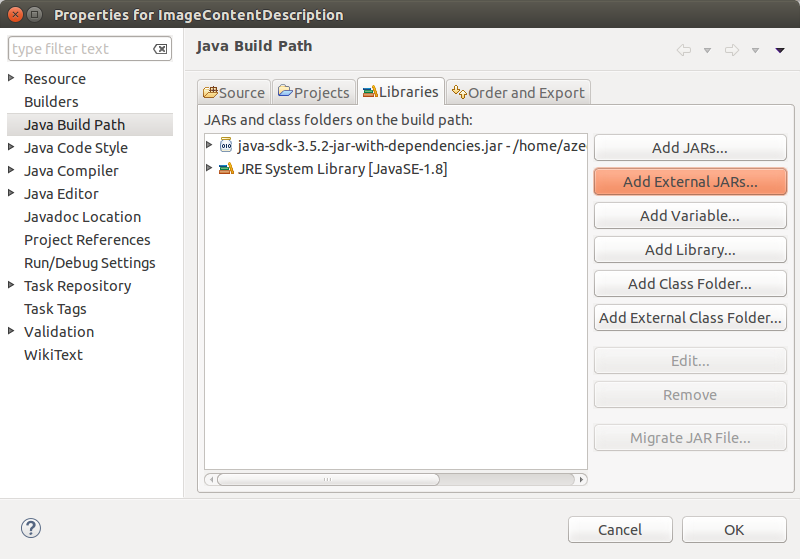
1



1. Once the jar file is downloaded, return to eclipse and **writeclick** the project name, browser to **Built path**and **s**elect **configure built path**



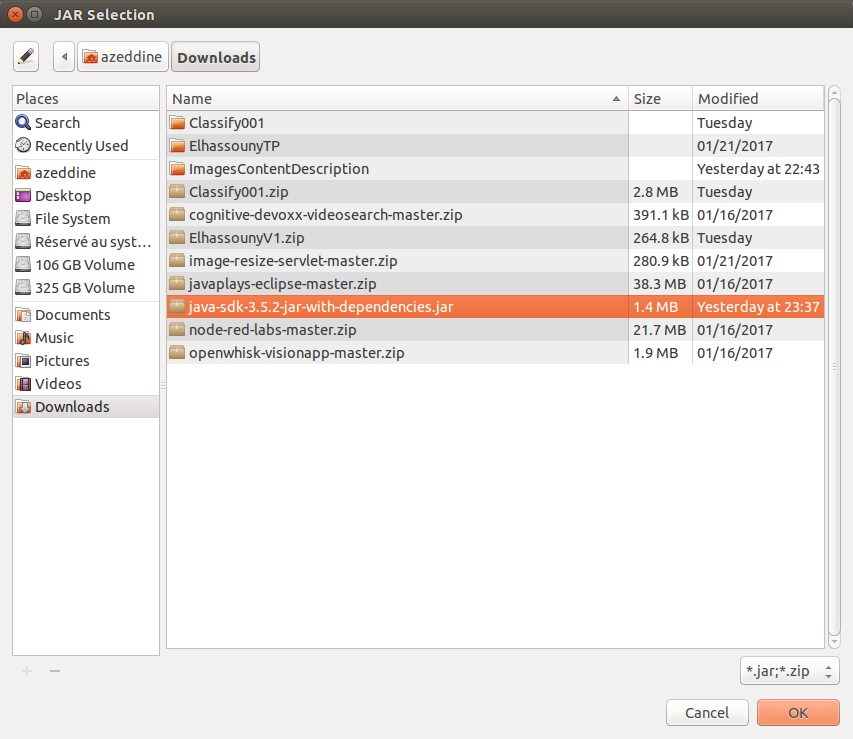
1. In the **library tab**, click on the **Add extenrnal JARs**

****

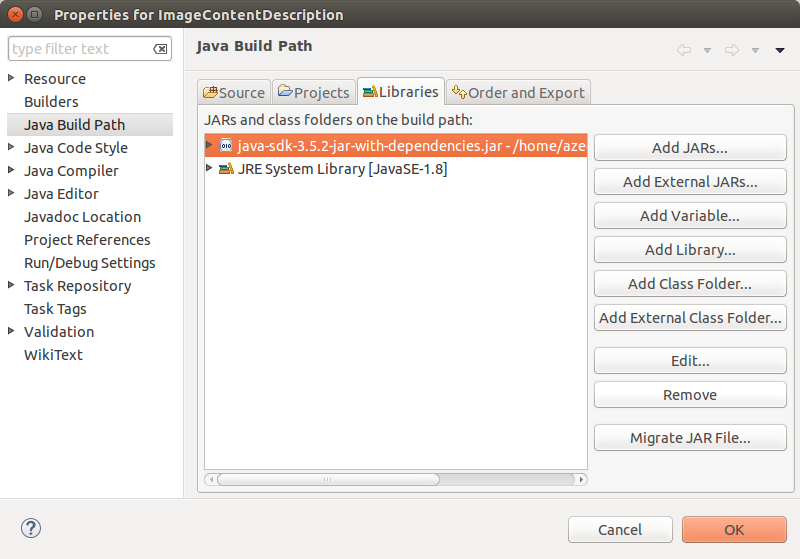
1

2

1. Navigate to the jar file **java-sdk-3.5.2-jar-with-dependencies.jar**, select it and click **Ok**



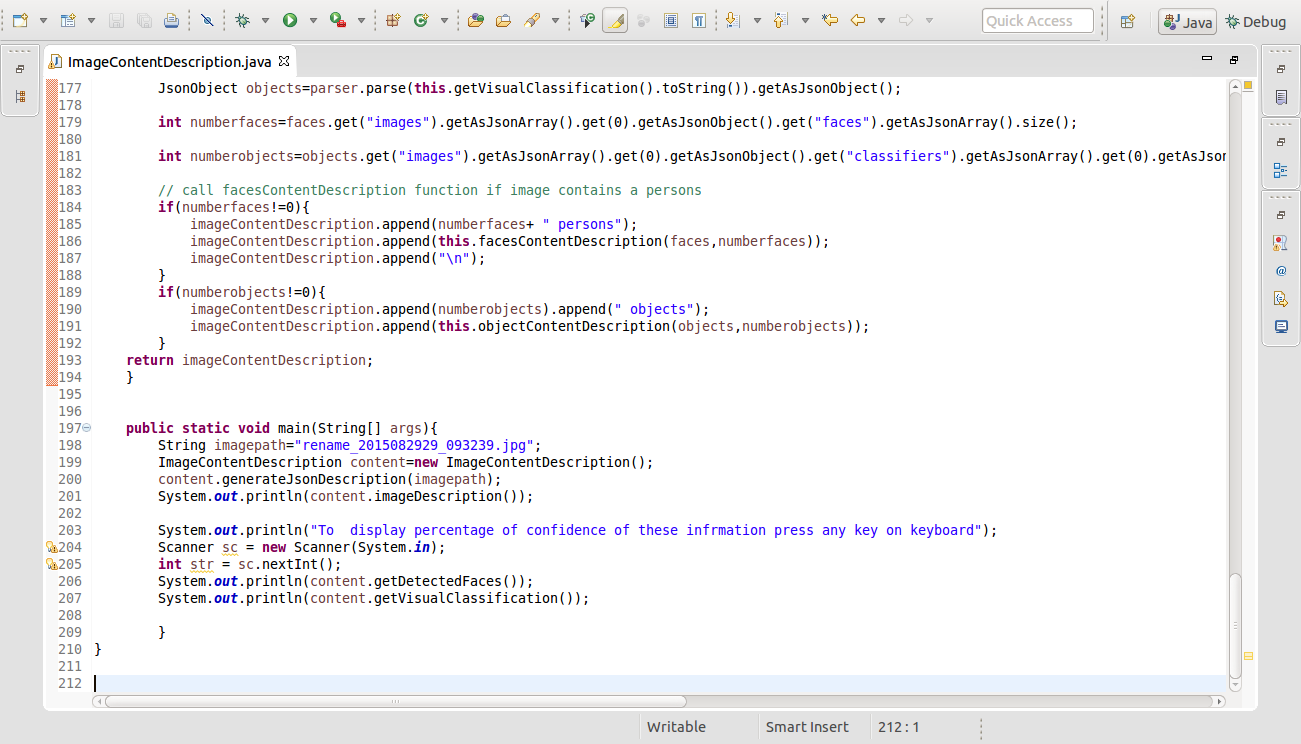
1. Check that the Jar file is added to your project and click **Ok button**

****

## Run app

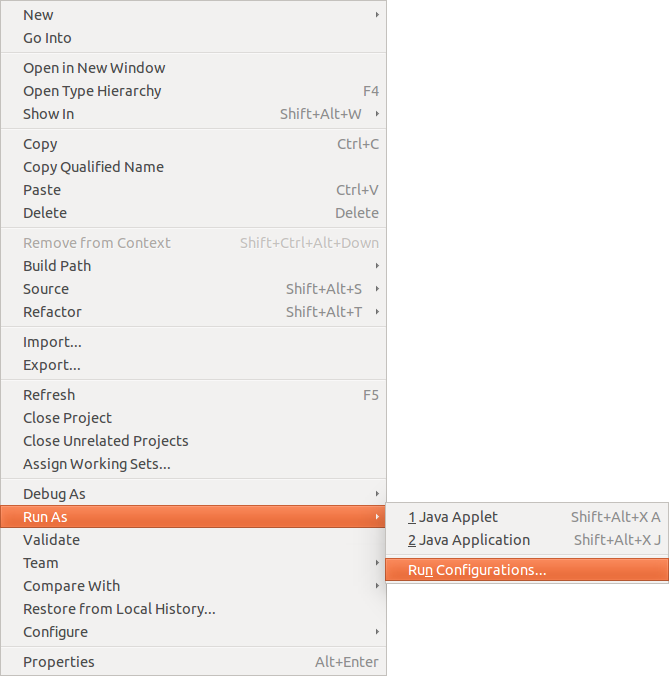
**Step 6.Run your application in local with eclipse**

1. To test the program, copy path of your image or use the paths of images (loaded with project)

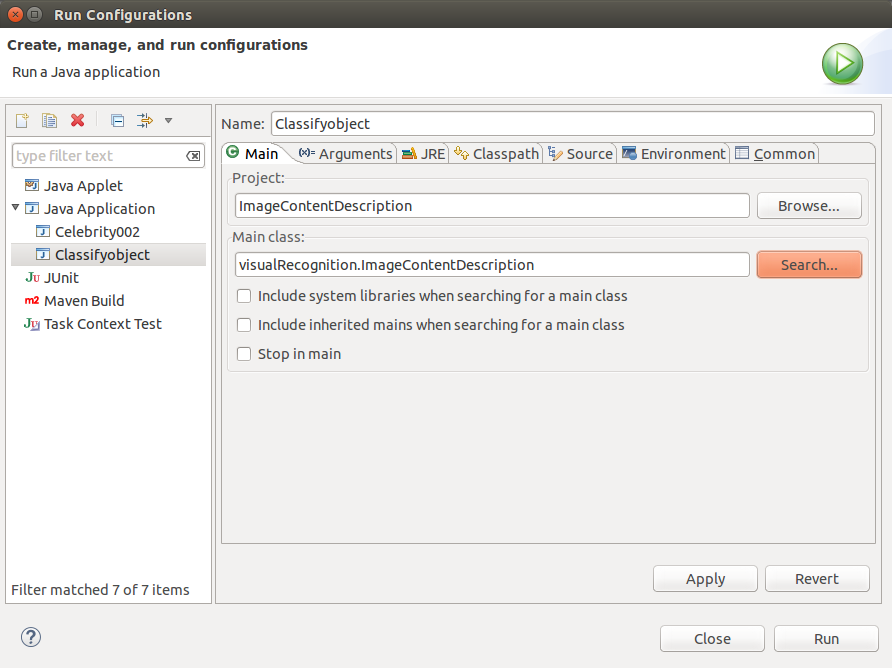


1

1. To run the project, select the project, **write click** on, and select **Run as**, **run application**



1. Select the ImageContentDescription project using **Browse Button,** select main class using **Search button and click Run Button to execute your project.**

****

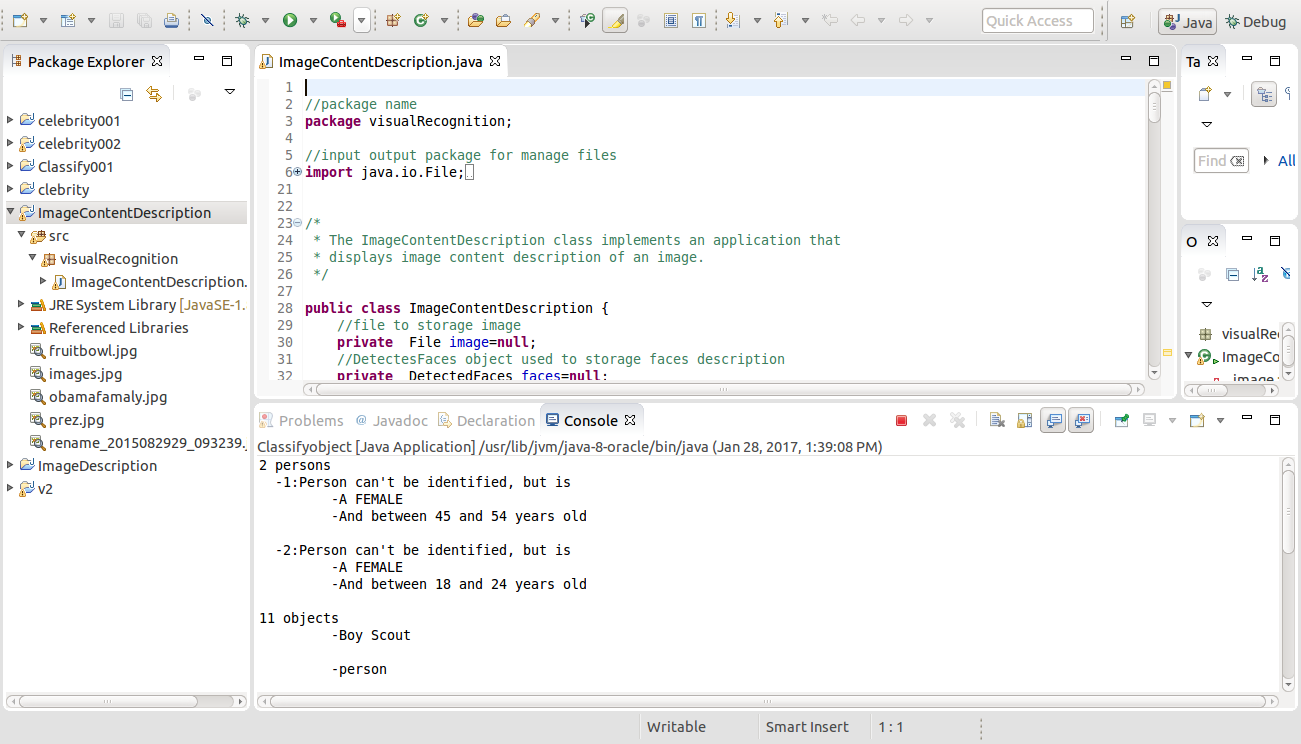
2

1

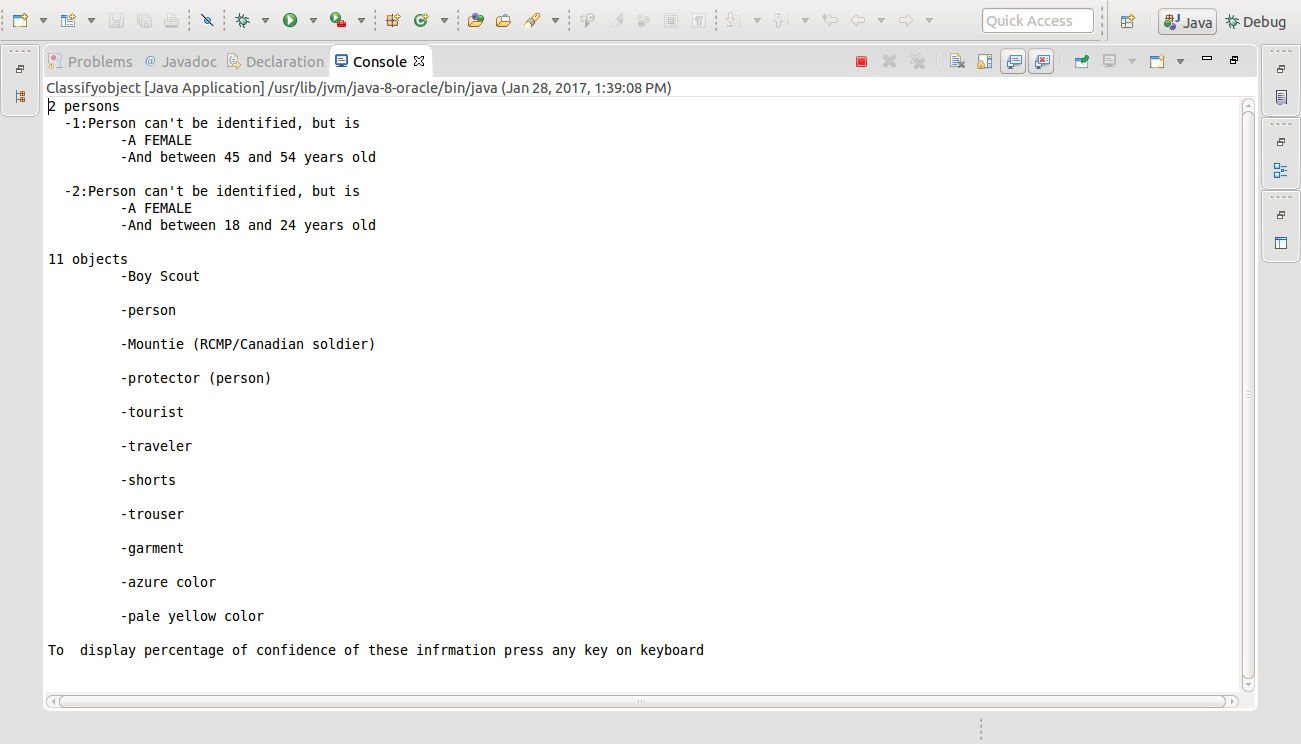
1. You will get, for this image input, the result below

Image used in this tutorial is given by Bjoern Steffens (visual recognition team member)

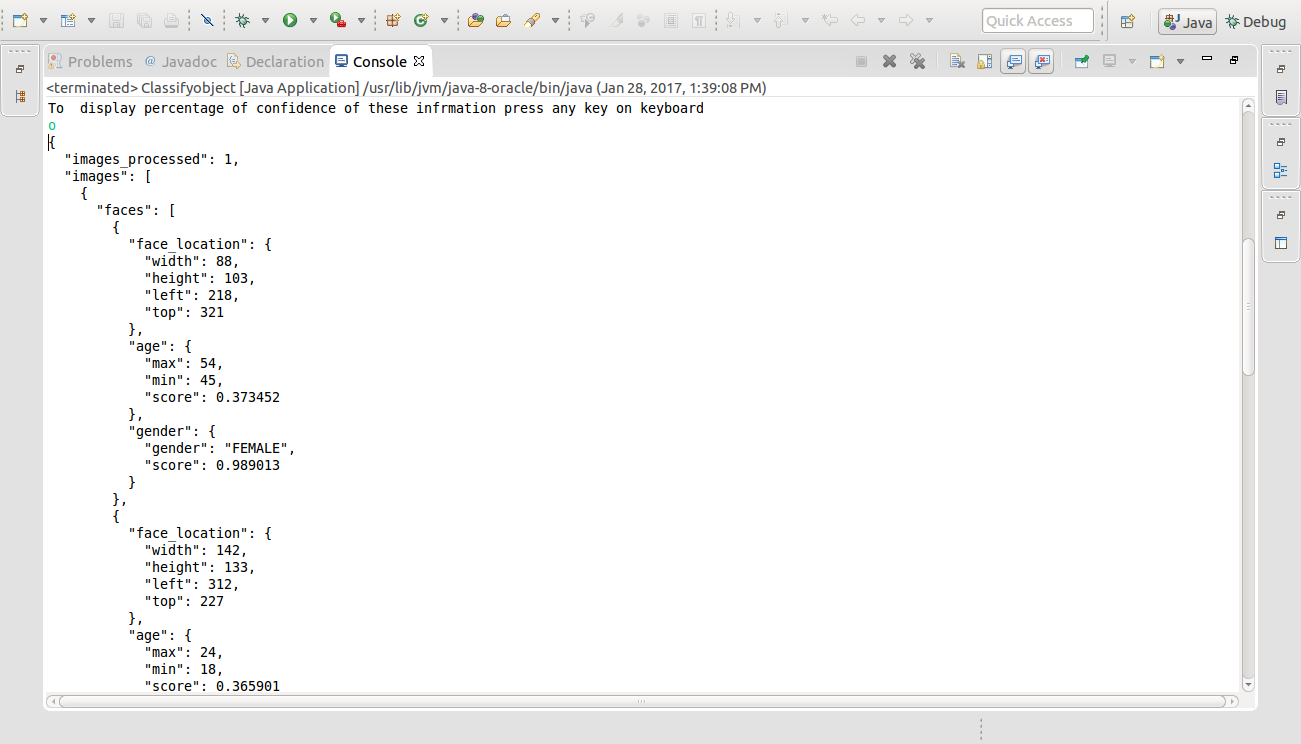


****

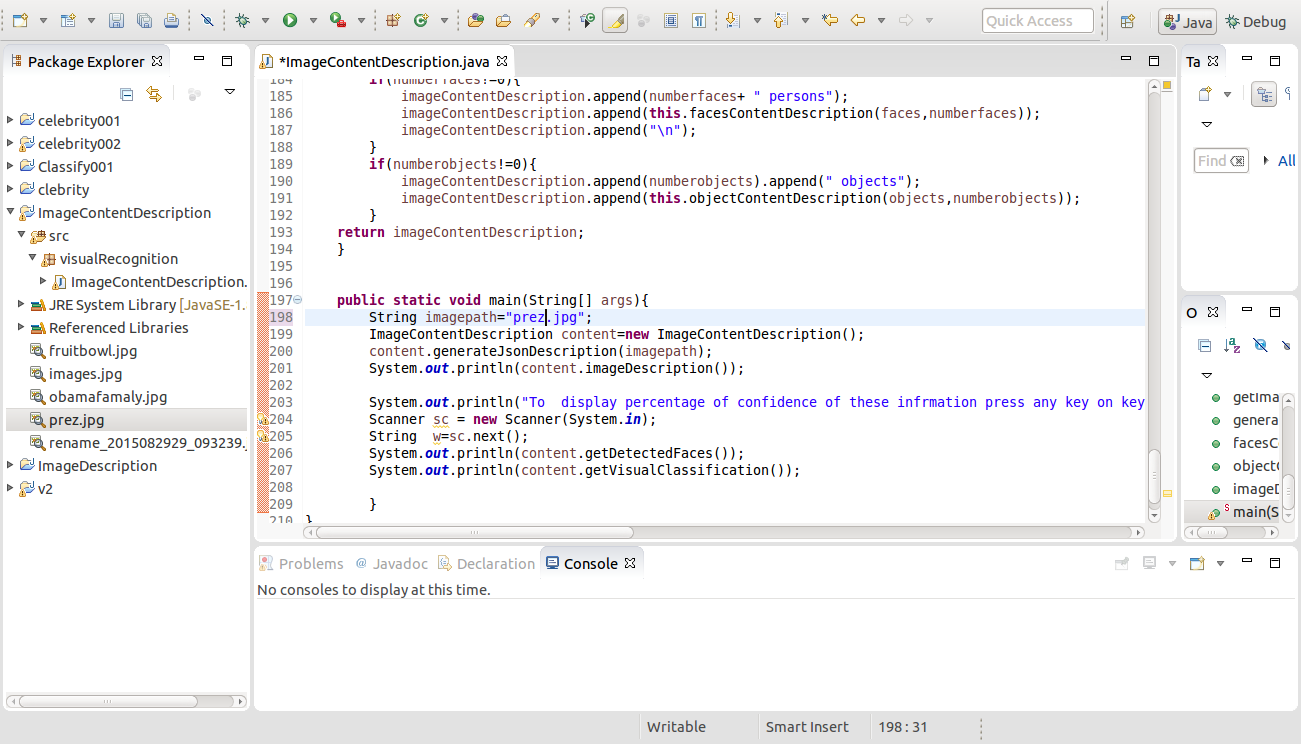
1. Maximize the console window to show all results

****

1. To display Jsons files with more details tape any key

****

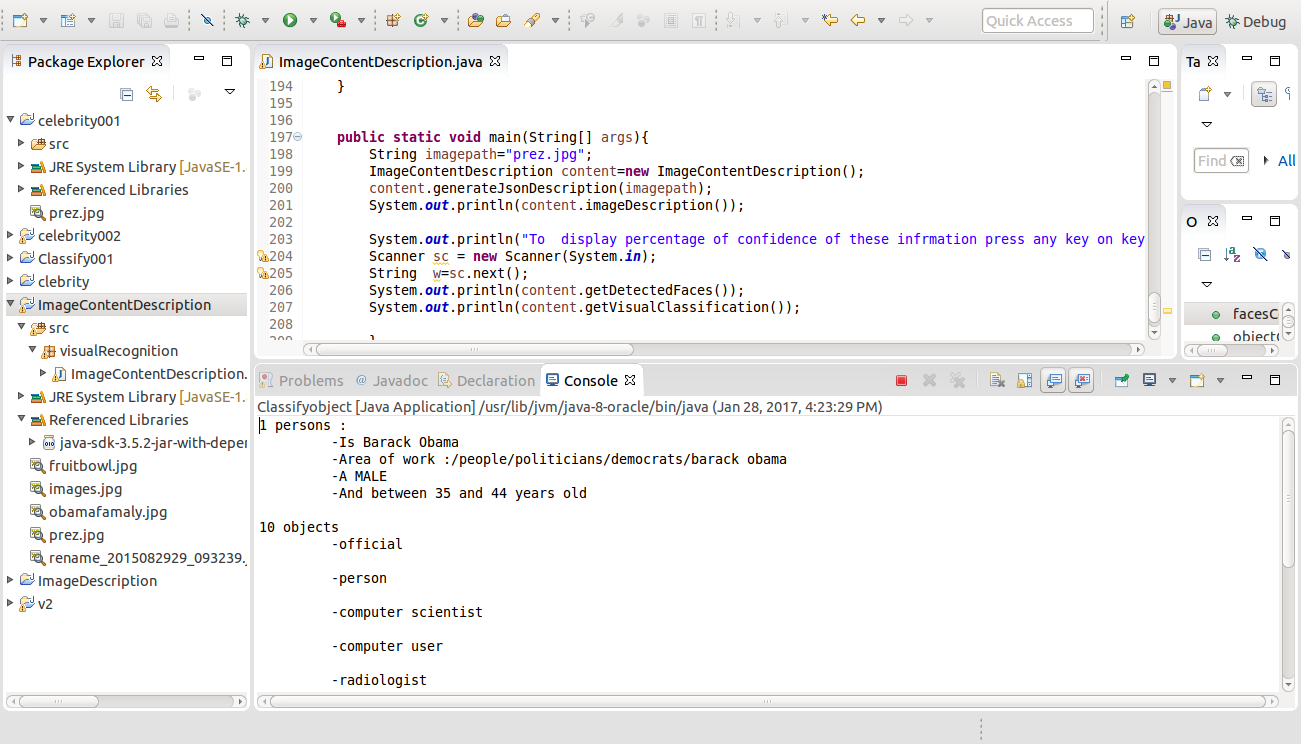
1. Another test of the program using Obama image to show the capabilité of the program to recognize face celebrity
   1. Change the actual path to the image path of Obama



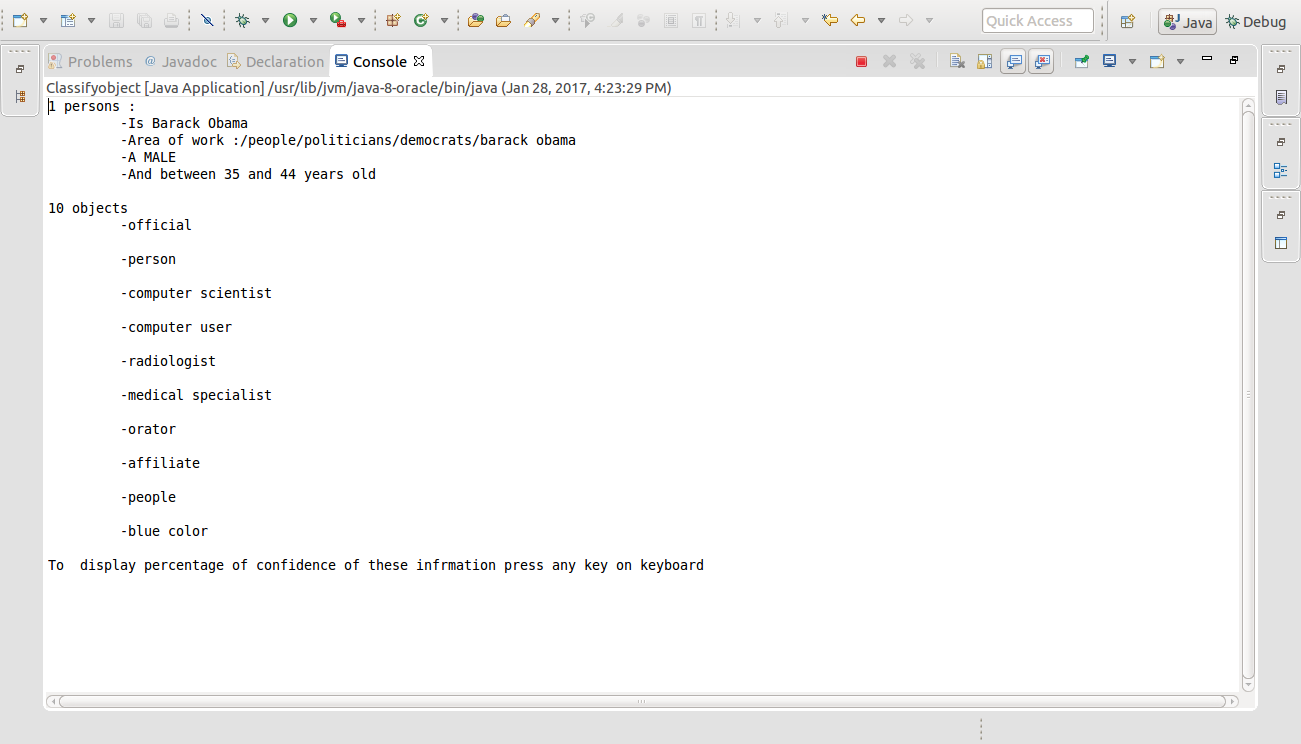
1

1. You will get the follow result for Obama picture



****

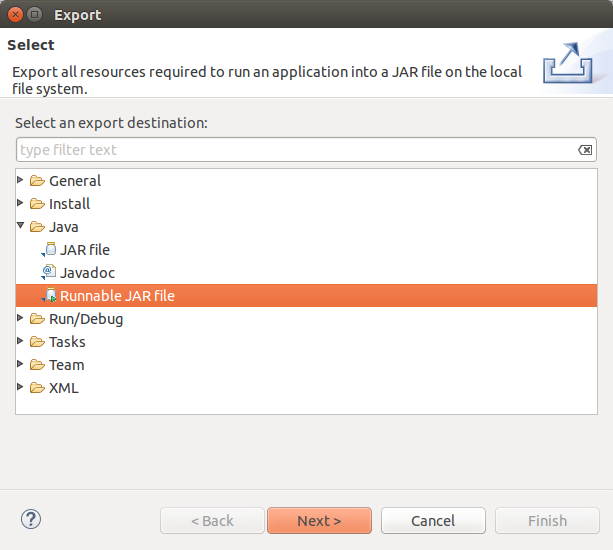
1. Maximize the console window to show all result

****

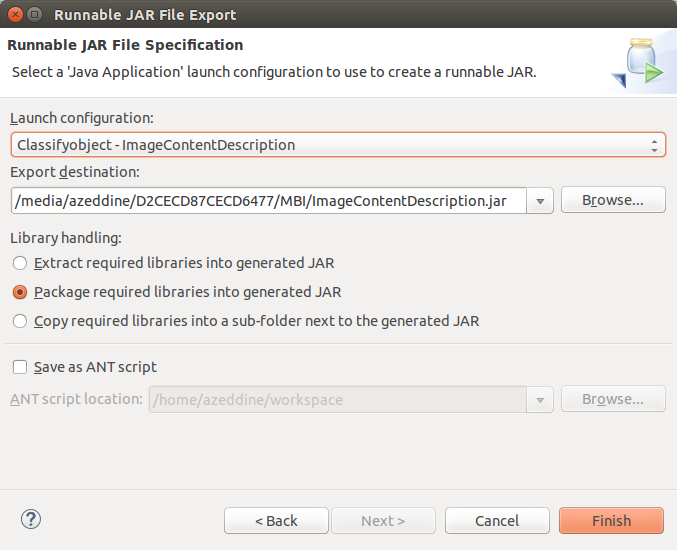
## Deploy a standalone java application to Bluemix

**Step7. Create a runnable JAR file to deploy the application to Bluemix**

1. To create a **runnable JAR** file, select **File > Export**, and in the **Export window**, make sure that you export not as a **standard JAR file**, but as a **Runnable JAR** file, then click **Next**



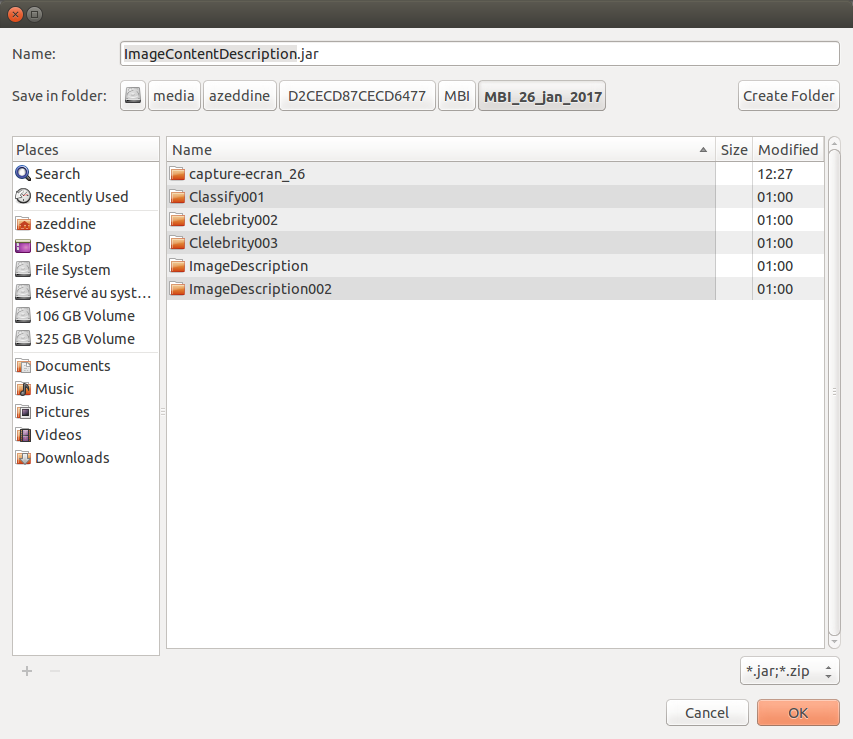
1. Select **ImageContentDescriptionas lunch configuration**, and click **Browse** button



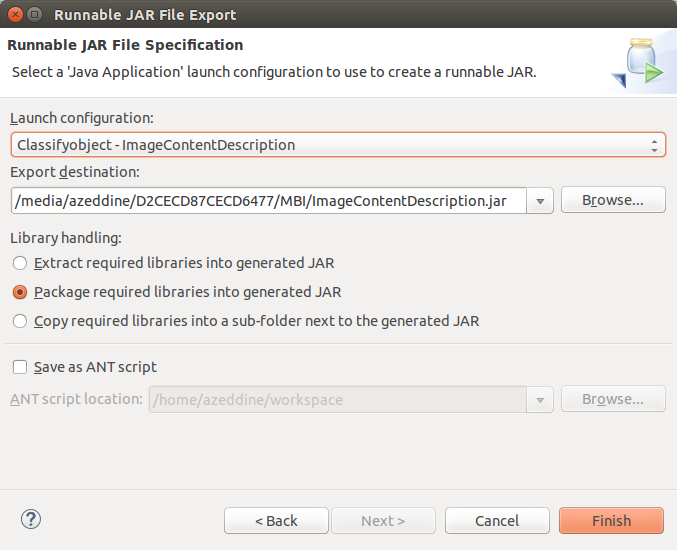
2

1

1. Navigate to folder where you will export your lunch configuration, enter name of your zip file and click **OK button**



1. To crate runnable JAR file, click **Finish** button



3

**Step8. Deploy a standalone java application to Bluemix**

In this section we will explain how do you make a Java program with a main () method run in Bluemix? (For more see <http://www.ibm.com/developerworks/cloud/library/cl-move-java-app-hybrid-cloud3-bluemix-trs/>)

1. Install **Cloud Foundry command-line interface (CLI)**
   1. **To get CLI use this link**

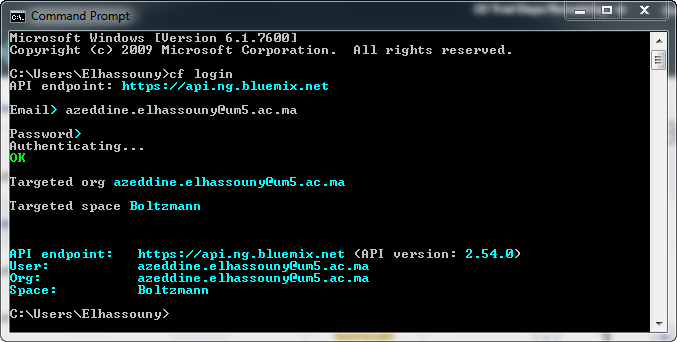
[**https://console.ng.bluemix.net/docs/starters/install\_cli.html**](https://console.ng.bluemix.net/docs/starters/install_cli.html)

* 1. download and install the Cloud Foundry command line interface

1. Lunch terminal command and login in cf using the command ***cf login*** :

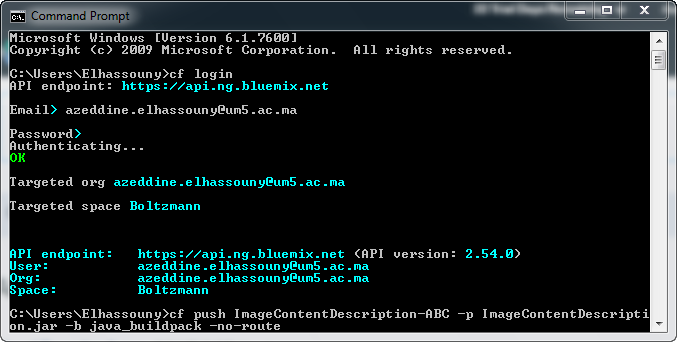


1. Enter your IBMid (your email address that you use to sign in Bleumx) and your password

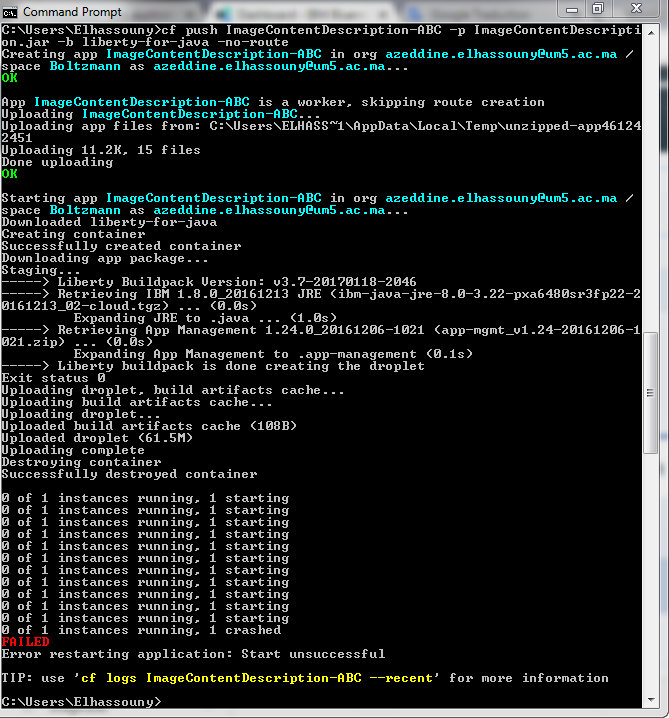


1. Using either the commands to deploy your java standlone application to Bluemix

Enter: ***cf push hellojavamain-ABC -p hellojavamainrunnable.jar -b java\_buildpack -no-route***  
or ***cf push hellojavamain-ABC -p hellojavamainrunnable.jar -b liberty-for-java -no-route***



1. If all perform without FAILD error, comeback to your Bluemx space to check the deployment of your app to Bluemix and you can see the execution of your application by click **logs.**



**(For me this step ended by FAILD, then, I can see my app deployed to my Bleumix space but I can’t see the execution in logs tab, this problem will be checked** later

## Code source documentation

For code source comments, explore documentation with javadoc in the folder ImageContentDescription/doc

## References

1. <https://bluecloudnews.com/jazzbot/how-to-create-bot-with-watson-visual-recognition-java/>
2. <https://developer.ibm.com/recipes/tutorials/bluemix-watson-apis-quickstart-using-java-sdk-version-2016/>
3. <http://watson-developer-cloud.github.io/java-sdk/docs/java-sdk3.3.1/com/ibm/watson/developer_cloud/visual_recognition/v3/VisualRecognition.html#VisualRecognition-java.lang.String->
4. <http://www.ibm.com/developerworks/cloud/library/cl-move-java-app-hybrid-cloud3-bluemix-trs/>
5. <https://www.ibm.com/watson/developercloud/visual-recognition/api/v3/>