Download di file da GoogleDrive

CONFIGURAZIONE PER QUARTA E QUINTA

Installare http://plugins.netbeans.org/plugin/49069/plantuml

Installare plugin Gradle in NetBeans

In NetBeans 10 installare plugin <http://plugins.netbeans.org/plugin/74901/oracle-js-parser-implementation>

Fare il download di

<https://developers.google.com/drive/api/v2/downloads>

<https://developers.google.com/api-client-library/java/google-api-java-client/download>

Passi

1. Creare un progetto Single Gradle Project senza main class
2. consultare pagina <https://developers.google.com/drive/api/v3/quickstart/java> e seguire le istruzioni oppure
3. andare alla pagina <https://console.developers.google.com/cloud-resource-manager>
4. creare un nuovo progetto con il nome GoogleDriveDownload
5. cliccare in alto a sinistra sulle tre lineette e selezionare DashBoard
6. cliccare su Abilita API
7. selezionare e abilitare Google DRIVE API
8. creare le credenziali con i seguenti parametri
   1. Google Drive API
   2. Altra UI
   3. Dati utente
9. Creare un ID (per esempio GoogleDriveMyID)
10. Selezione come indirizzo email per il consenso quello dell’utente Google al cui drive si vuole accedere (per esempio ….@gmail.com) e come nome prodotto un nome che permetta all’utente di identificare il software a cui sta dando l’autorizzazione di accedere ai file (per esempio TapShool)
11. creare un Cloud Platform project tramite bottone o utilizzando la pagina Cloud Platform project
12. Scarica le credenziali nella cartella src\main\resources
13. Copiare il contenuto di build.gradle della pagina web nel file presente in NetBeans in modo che il risultato finale sia simile al seguente

apply plugin: 'java'

apply plugin: 'application'

mainClassName = 'NBGoogleDriveDownload'

targetCompatibility = 1.7

version = '1.0'

sourceCompatibility = '1.8'

[compileJava, compileTestJava]\*.options\*.encoding = 'UTF-8'

// NetBeans will automatically add "run" and "debug" tasks relying on the

// "mainClass" property. You may however define the property prior executing

// tasks by passing a "-PmainClass=<QUALIFIED\_CLASS\_NAME>" argument.

//

// Note however, that you may define your own "run" and "debug" task if you

// prefer. In this case NetBeans will not add these tasks but you may rely on

// your own implementation.

if (!hasProperty('mainClass')) {

ext.mainClass = ''

}

repositories {

mavenCentral()

// You may define additional repositories, or even remove "mavenCentral()".

// Read more about repositories here:

// http://www.gradle.org/docs/current/userguide/dependency\_management.html#sec:repositories

}

dependencies {

// TODO: Add dependencies here ...

// You can read more about how to add dependency here:

// http://www.gradle.org/docs/current/userguide/dependency\_management.html#sec:how\_to\_declare\_your\_dependencies

testCompile group: 'junit', name: 'junit', version: '4.10'

compile 'com.google.api-client:google-api-client:1.23.0'

compile 'com.google.oauth-client:google-oauth-client-jetty:1.23.0'

compile 'com.google.apis:google-api-services-drive:v3-rev110-1.23.0'

}

1. Creare il file java con il main in modo che abbia lo stesso nome indicato nella direttiva mainClassName =
2. Compilare il tutto con >Build e poi Run

Per cambiare le autorizzazioni

1. Modificare l’inizializzazione di SCOPES
2. Rimuovere il file presente nella cartella Tokens
3. Rieseguire il programma
4. Accettare le nuove richieste da parte dell’applicazione