Developer Guide

Java Visual Debuggers – Jeliot and Online Java-Visualizer

Overview

**[1.](#_Toc435739248)****[Requirements](#_Toc435739248)** [2](#_Toc435739248)

**[1.1.](#_Toc435739249)****[Jeliot Plugin](#_Toc435739249)** [2](#_Toc435739249)

**[1.2.](#_Toc435739252)****[Java-Visualizer](#_Toc435739252)** [2](#_Toc435739252)

**[2.](#_Toc435739253)****[Installation](#_Toc435739253)** [2](#_Toc435739253)

**[2.1.](#_Toc435739255)****[Jeliot Plugin](#_Toc435739255)** [2](#_Toc435739255)

[2.1.1. Server Side 2](#_Toc435739259)

[2.1.2 Client Side 5](#_Toc435739260)

**[2.2.](#_Toc435739261)****[Java Visualizer and Coding Arena](#_Toc435739261)** [9](#_Toc435739261)

# **Requirements**

## **Jeliot Plugin**

Please make sure you have a working copy of Moodle 2.0 or above hosted on your server and have admin rights. Also make sure the client has a working copy of Java and Java Web Start on his/her machine, and has added your website to the list of exceptions in Java Security.

If you plan on **installing just the Jeliot plugin and not Java Visualizer**, it’ll generate a view having “Visualize Online”. So you’ll need to configure the view.php in the Jeliot plugin and remove the line:

echo '<center><a title="Visualize Online" target="\_blank" href="'.$baseurl.'"><button>Visualize Online!</button></a>';



## **Java-Visualizer**

Please make sure you have a Linux environment, superuser or root access and a working internet connection (to download backend). Also, you must have git installed in your Linux environment and have a webserver (eg: apache).

# **Installation**



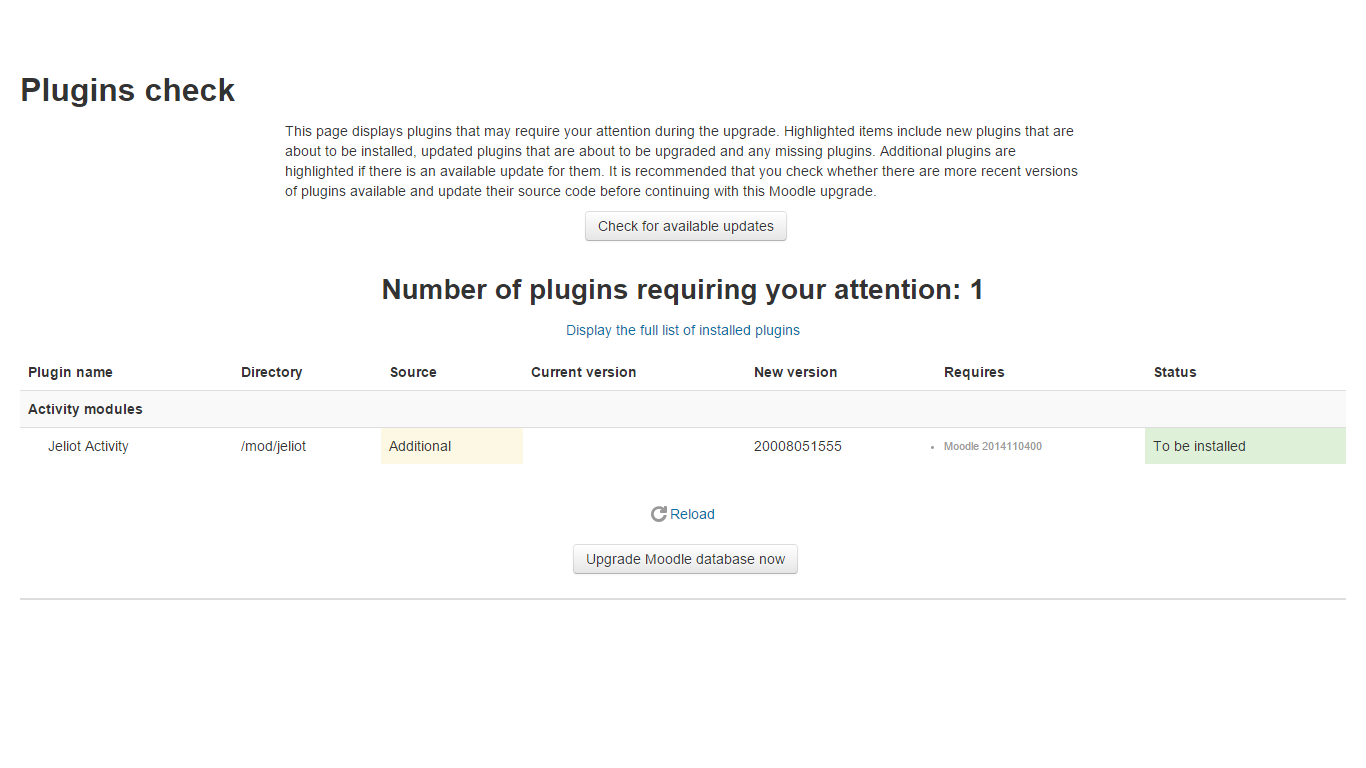
## **Jeliot Plugin**



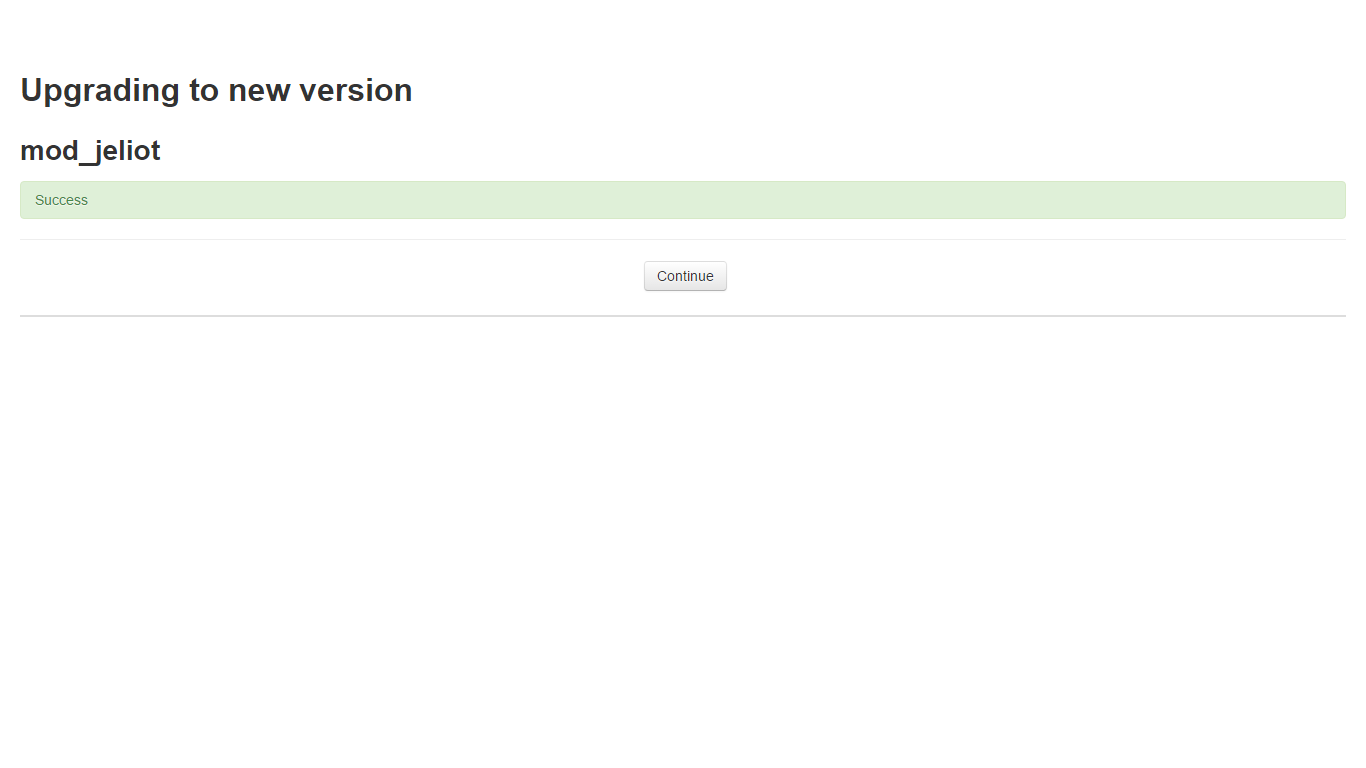
### Server Side

You must have set up Moodle 2.0 or above to continue.

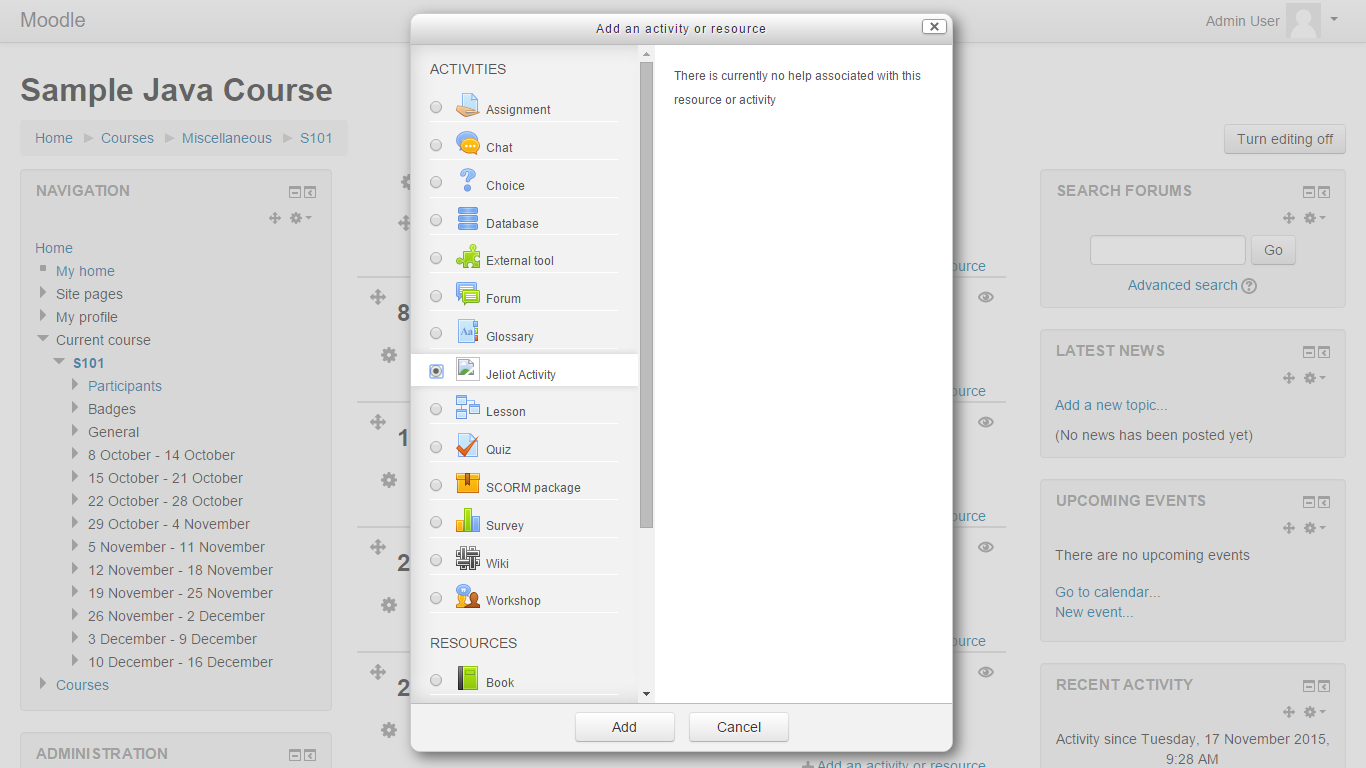
1. If Jeliot plugin is zipped, extract it to a folder. Rename that folder as “jeliot”. Else, if you have an unzipped version, rename the folder to “jeliot” if it’s something else.
2. Copy the “jeliot” folder inside /moodle/mod/ directory, i.e. place the jeliot folder inside the mod folder located in moodle (hosted in the webserver directory).
3. Open your browser and navigate to your installed Moodle URL.
4. If you’re logged in, log out and re-login as Moodle admin.
5. You’ll see the following screen. Click on Update Moodle Database now. It may take time, so please be patient.



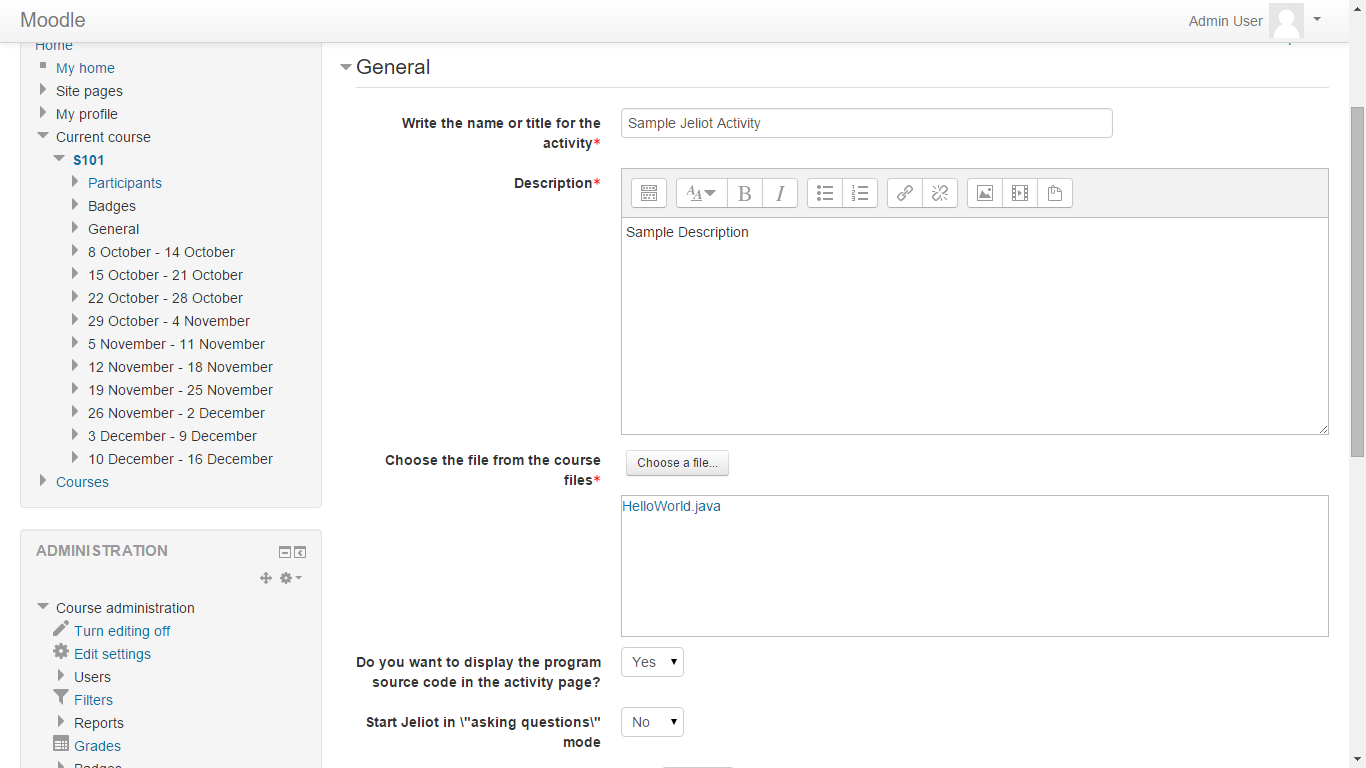
1. Click on Continue on the next page.



1. Now you should be able to add a Jeliot Activity. Head over to a course, and click on Add an Activity or Resource. Select Jeliot Activity. Press Add.



1. Add a Name and Description to your activity. Upload the Java file you want the students to see and visualize using Jeliot.



1. Click on Save and Display. You’ll be able to see your uploaded code and a button called Visualize in Jeliot. Clicking on that will prompt to download a JNLP file, which when run by Java Web Start, launches Jeliot along with the Java Code.

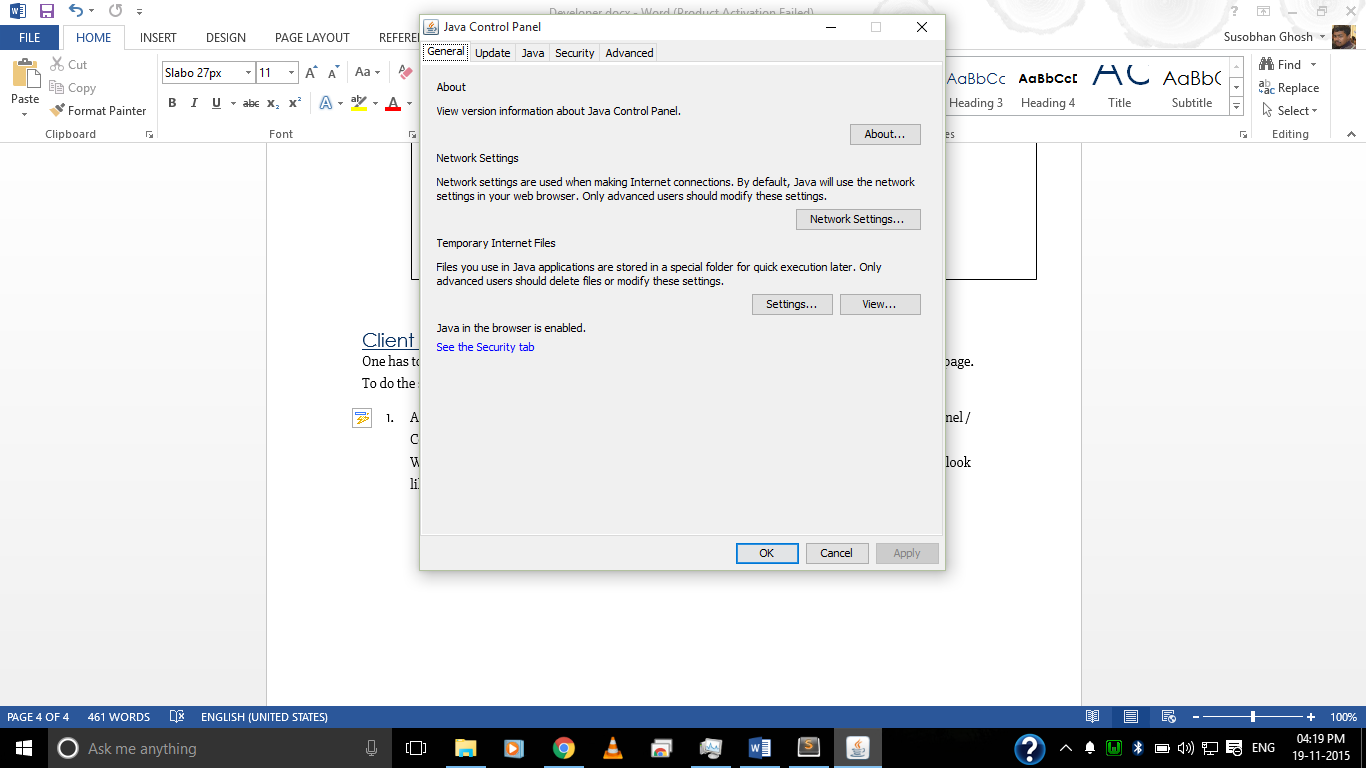


### Client Side

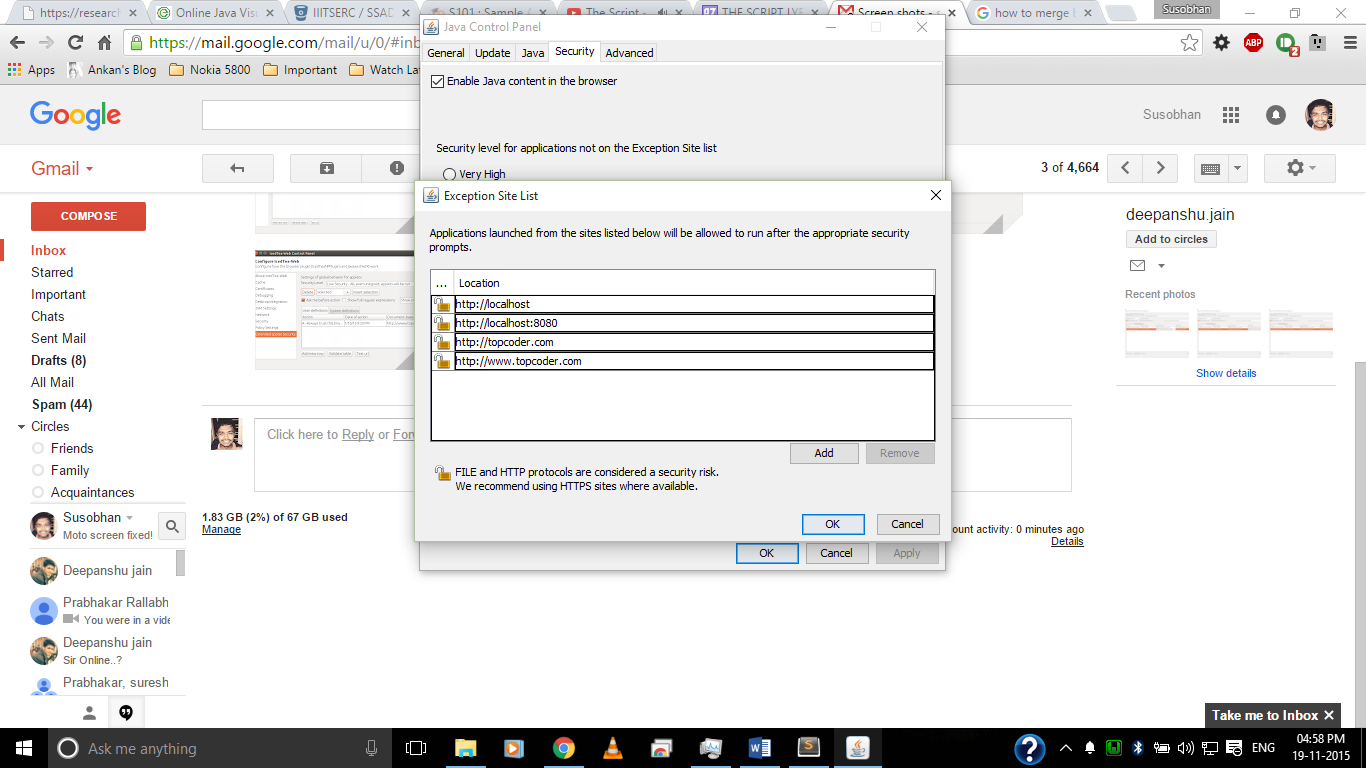
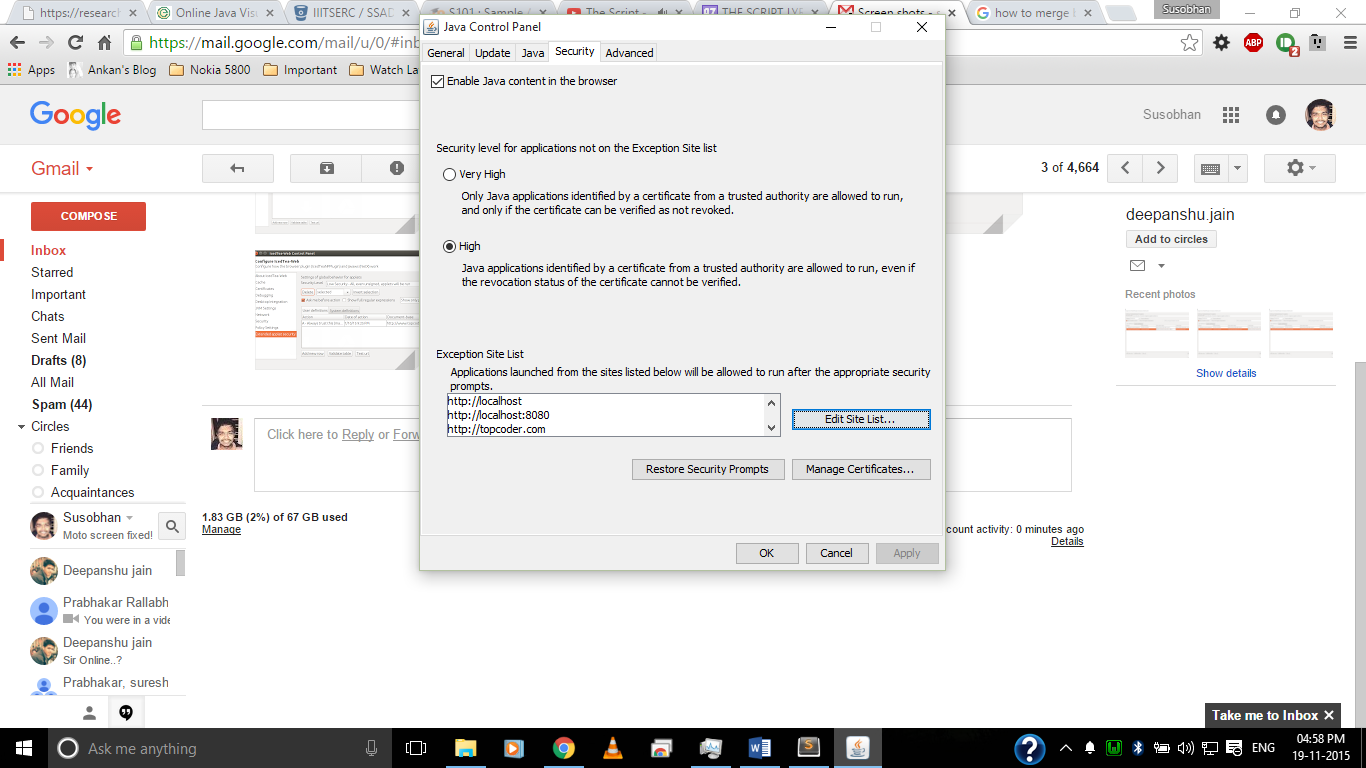
One has to add your website to their list of exceptions to run the JNLP file downloaded from the Jeliot page. To do the same, follow the given steps:

#### For Windows:

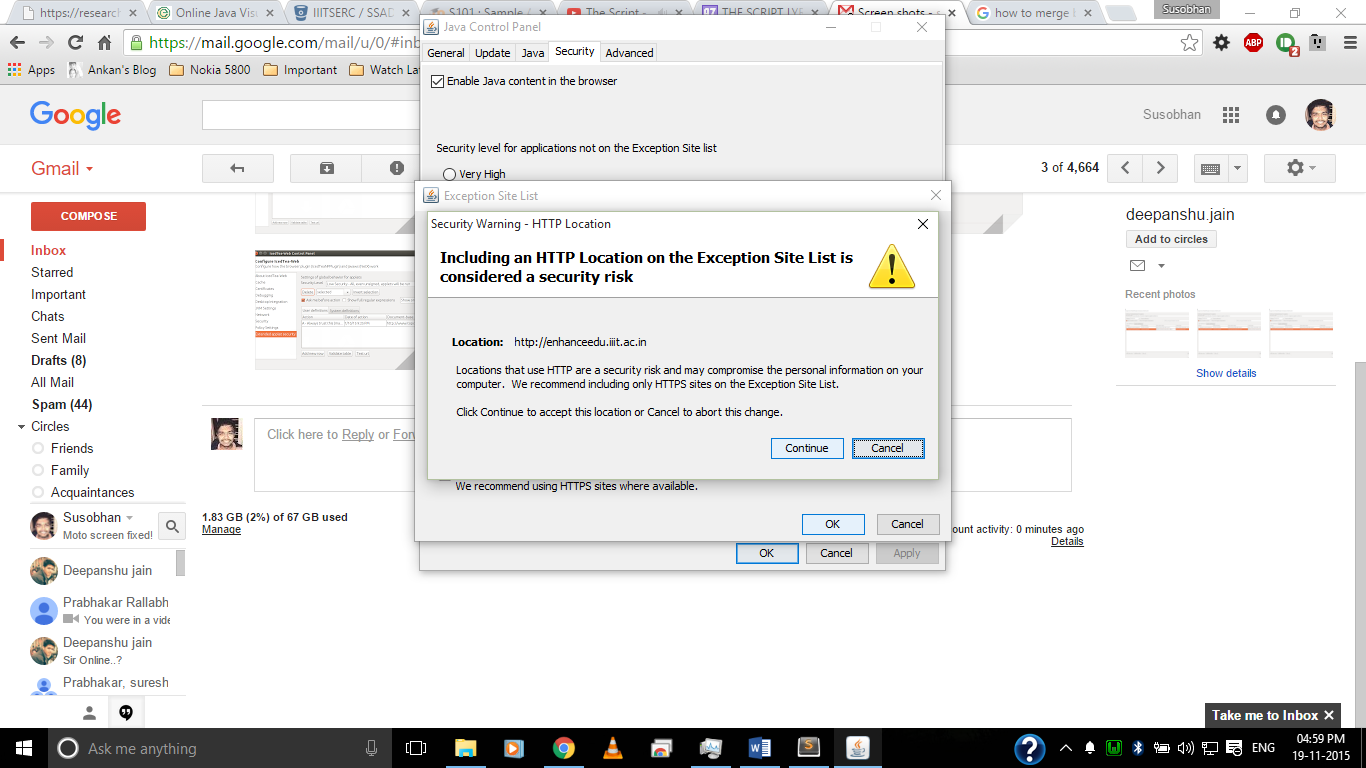
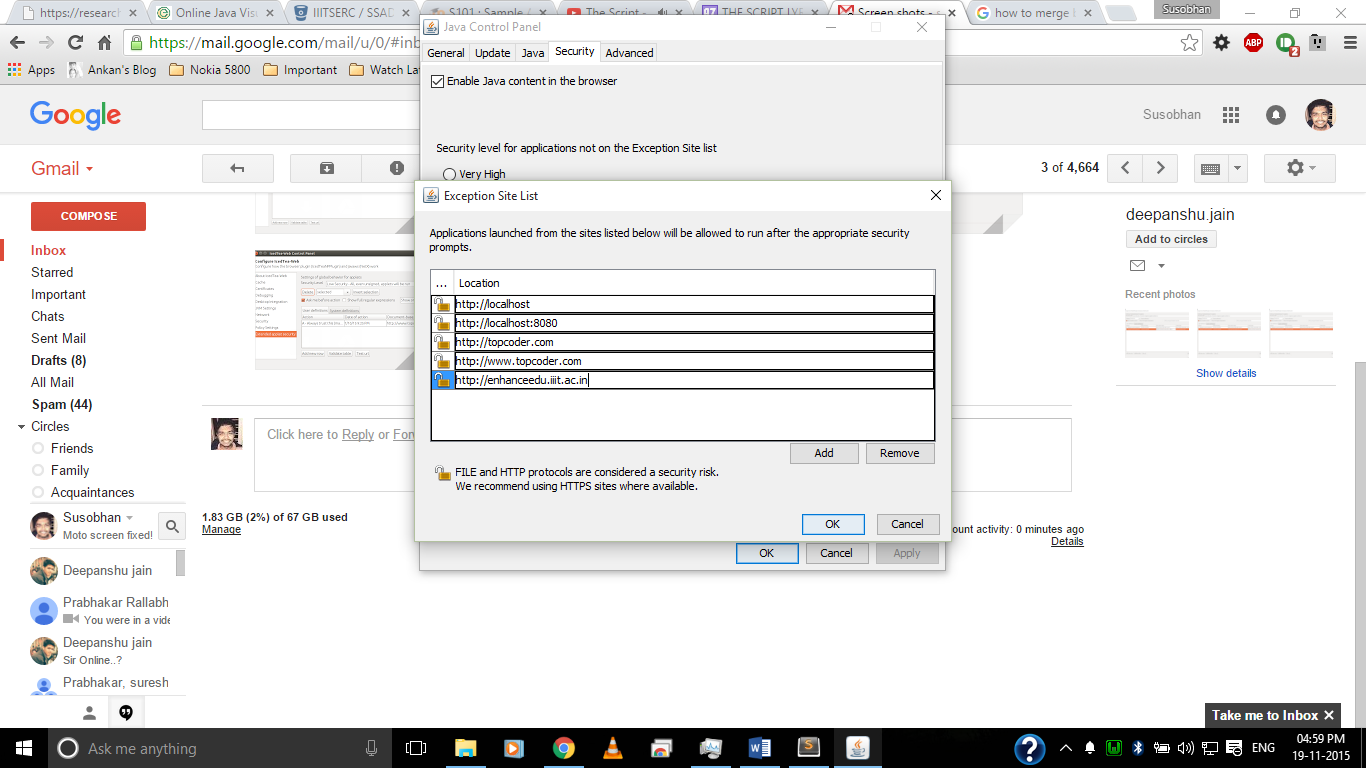
1. After downloading the JNLP file and installing Java (preferably Java 8), open Java Security Panel / Control Panel /settings. On Windows 10 and Java 8– Look for Configure Java. Clicking on it will open a setting which will look like this:



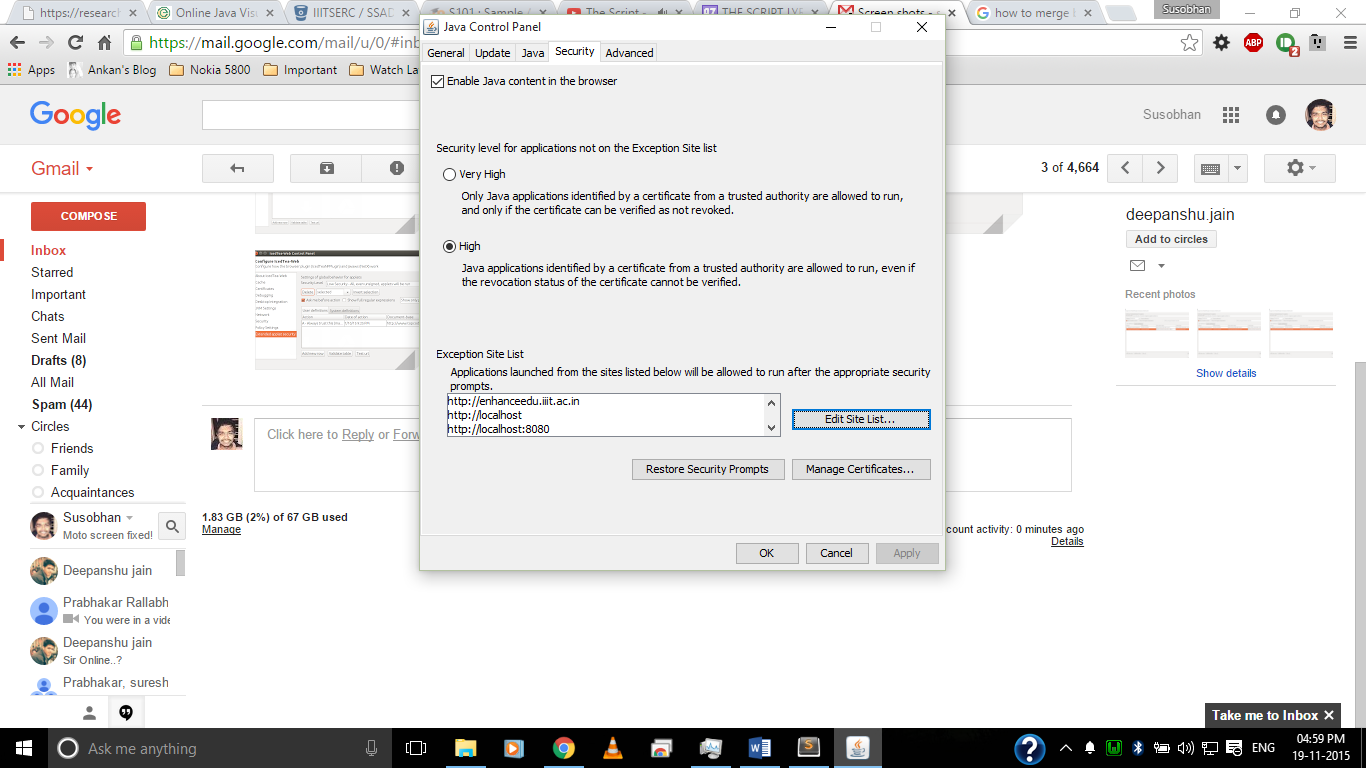
1. Go to the security tab. In Windows, Security Tab and Click on Edit Site List.



1. Click on add and enter your site name from where JNLP was downloaded (eg: <http://enhanceedu.iiit.ac.in>). And press OK. You’ll get a confirmation message. Press continue.



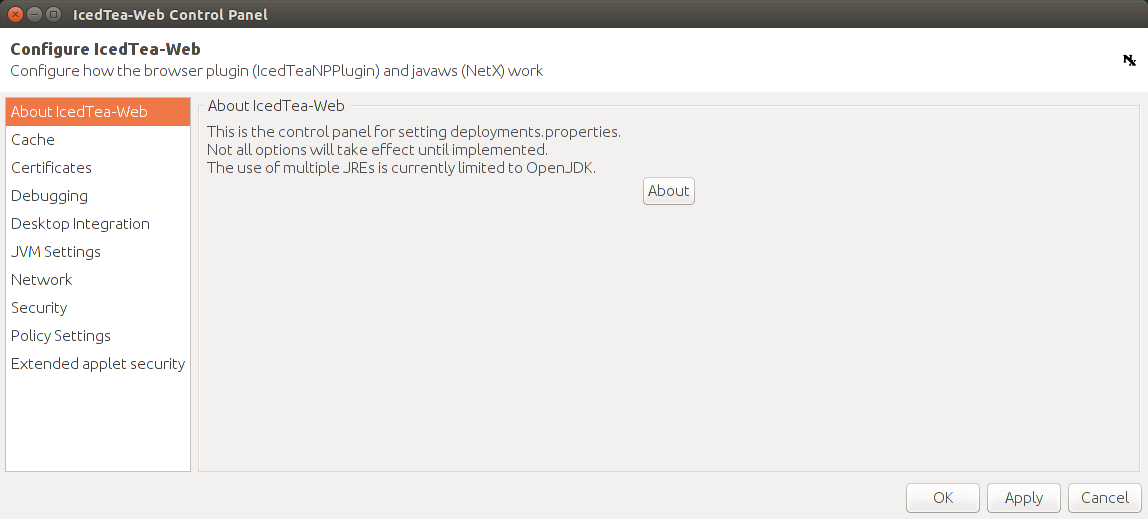
1. Press OK again to confirm all the changes.



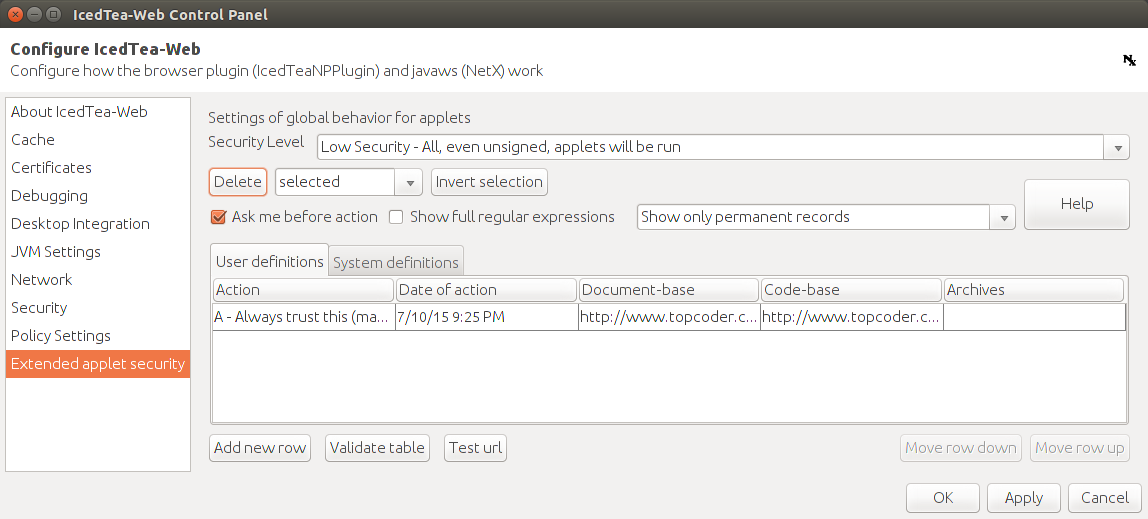
1. Now you can safely run the downloaded JNLP file using Java Web Start (click I accept and continue when prompted) and it will display Jeliot and the updated code on your machine.

#### For Linux:

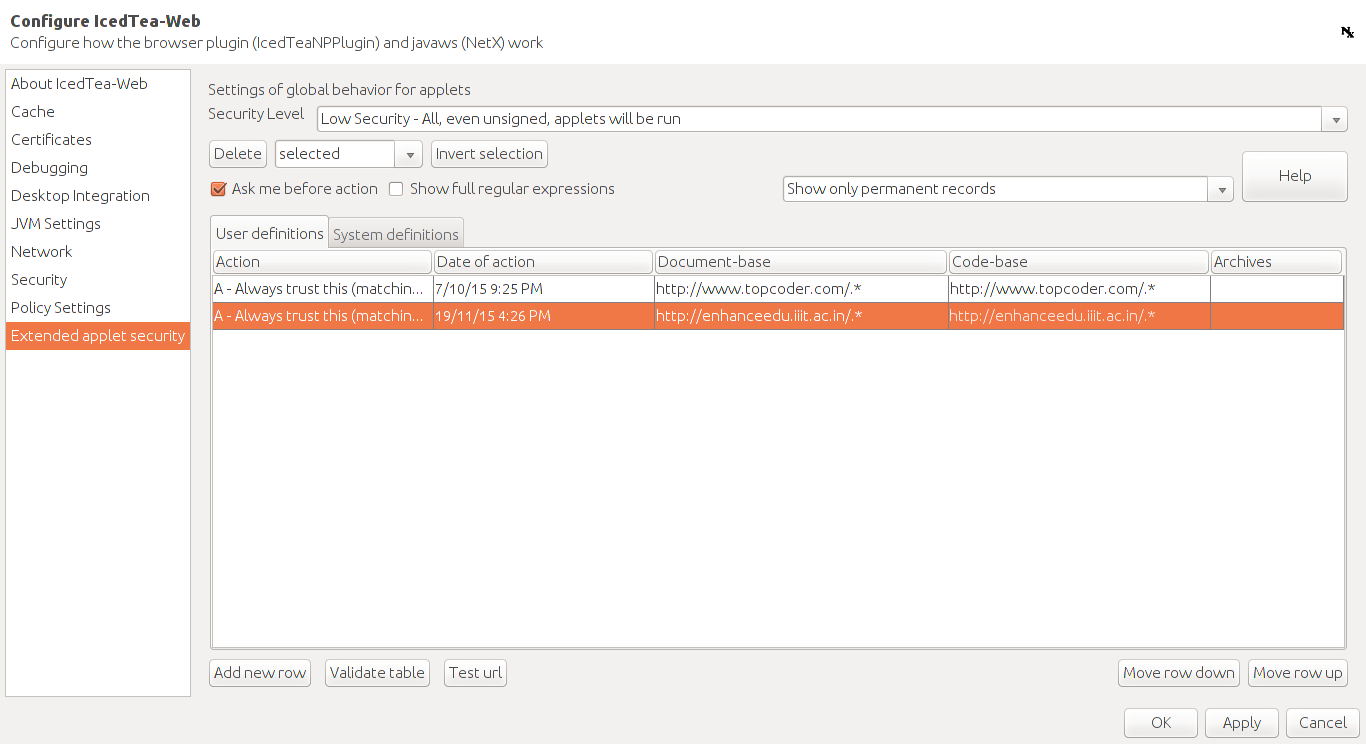
1. Having installed Java, open the Java Control Panel. On Ubuntu 14.04 and Java 7, one can find IcedTea Web Control Panel, which when opened will look like the following. Java Control Panel may differ over different Java versions.



1. Head over to the security tab. In IcedTea Web Control Panel, it is labelled as Extended applet security.



1. Click on Add new row. Fill in your website URL from where JNLP was downloaded (eg: <http://enhanceedu.iiit.ac.in>). Both Document-base and code-base entries should be filled. This may differ in different Java versions and control panels.

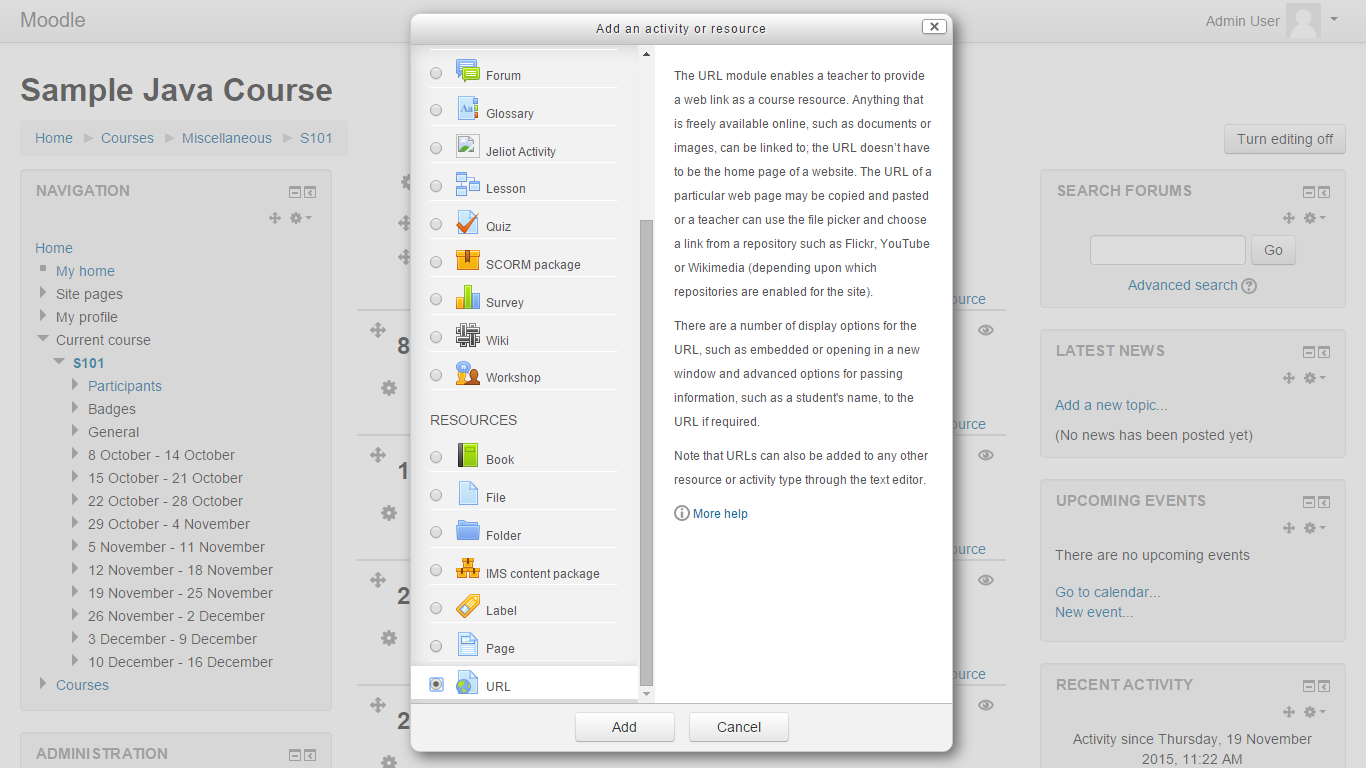


1. Click on Apply and exit by clicking on OK. Now you can run your JNLP file using Java Web Start (IcedTea Web Start or appropriate Java Web Start app) and run Jeliot with the updated Code.

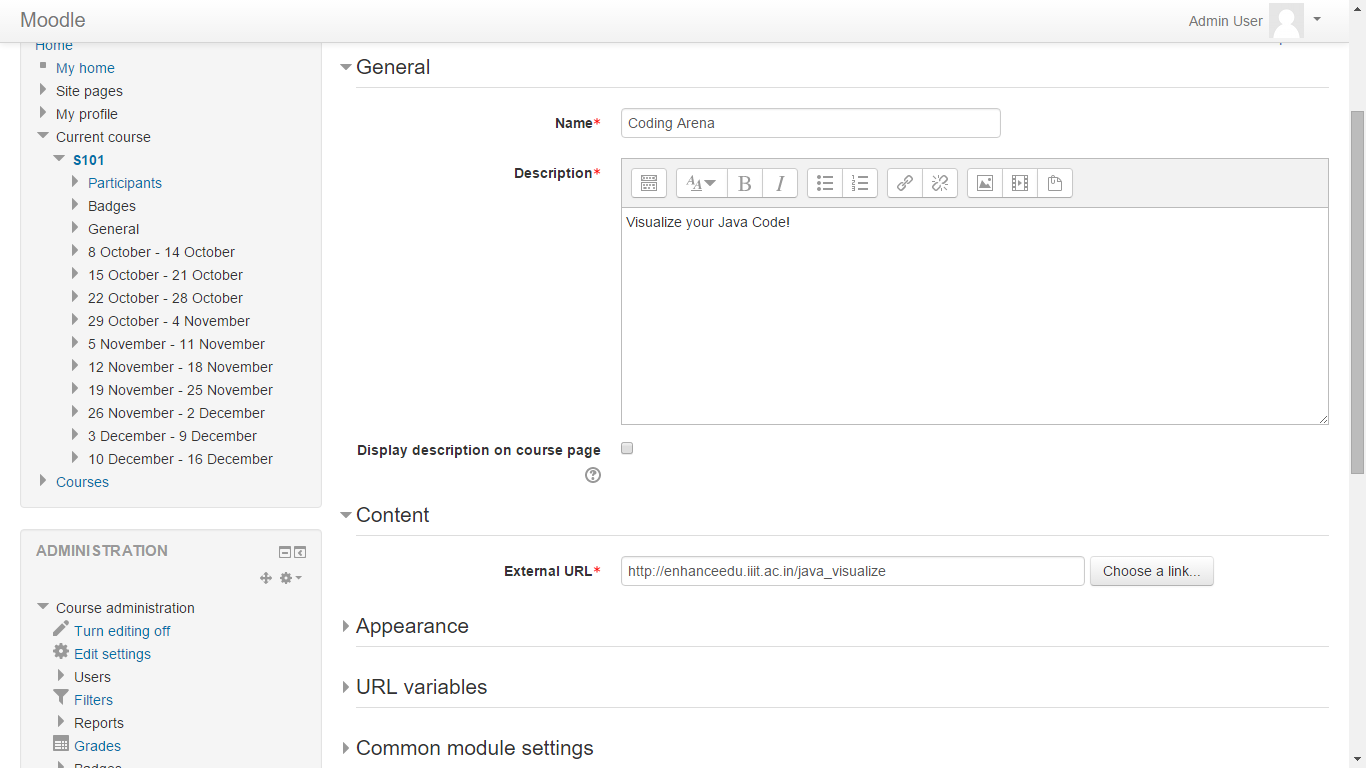
## **Java Visualizer and Coding Arena**

One needs a working internet connection and superuser access to Linux and git installed to proceed with the installation. Steps:

1. Install safeexec. It can be found here along with setup instructions: <https://github.com/cemc/safeexec>
2. Install java\_jail. Java\_jail can be found here along with setup instructions: <https://github.com/daveagp/java_jail> . Make sure you test with the two given commands that your installation is correct and successful. If it is throwing errors / unexpected outputs, try increasing value or nproc or –mem parameter, or if still unsuccessful, try posting an issue on that project. Couple of issues are already present that can help troubleshoot your issues, make sure to give them a read before proceeding.
3. After installing java\_jail and safeexec properly to a directory not accessible by the web server, copy the java\_visualize folder to the same directory level as of Moodle. That is, if Moodle is at /var/www/html/moodle, java\_visualize should be located at /var/www/html/java\_visualize.   
   If you plan on putting it in another sub-directory, make sure to change view.php of Jeliot and change the $baseurl attribute appropriately.
4. Open the jv-config.json file. Edit the first two lines -   
     
   "safeexec-executable-abspath": "/your/path/to/safeexec/safeexec"  
   "java\_jail-abspath": "/your/path/to/java\_jail/"  
     
   Enter the full path to safeexec (including safeexec file) and full path to the java\_jail directory.  
   Block access to this jv-config.json file using .htaccess to prevent web users to access your settings.
5. If you had changed parameters during java\_jail installation (test 2 using safeexec), change the appropriate values in java\_safe\_ram\_maketrace.php (from line 58). “mem” is for the memory argument, nproc is for the nproc argument and so on. If this is not setup properly, it’ll break your installation.
6. Change the group owner of the “upload” directory inside java\_visualize to your web-user using chgrp command. Ubuntu has www-data, CentOS has apache.
7. Give write permission to the webserver user group for the “upload” folder. Use the following command to achieve the same while inside the java\_visualize folder:   
   sudo chmod 775 upload
8. Open your web browser and navigate to Moodle. Select a course and turn editing on. On the very top, add an activity or resource, select URL. Click on Add.



1. Name the Resource as “Coding Arena”. Give a proper description. And give the URL to your java\_visualize page. Click on Save and Display! Hurray, you have linked your Coding Arena and made it available to the students.



1. Navigate to java\_visualize. You can now type your code, and save it, visualize it online, or even start it using Jeliot!