Installation and User Guide

Prerequisites:

- Eclipse or Jetbrains IDE
- A Slack workspace to be used as a communication channel

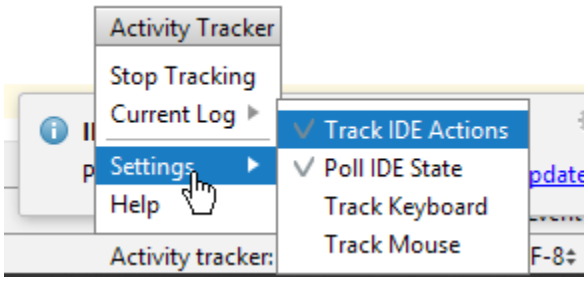
Step 1: Installing activity tracker plugin

If using Eclipse IDE:

1. Download "rabbit-eclipse_1.0 final.zip" from https://code.google.com/archive/p/rabbit-eclipse/downloads
2. Unpack the contents of the zip file into the "dropins" folder in the Eclipse IDE install location (e. g. C:\Users\USER\eclipse\jee-latest-released\eclipse\dropins)
3. Restart Eclipse IDE if it was running

If using Jetbrains line of products (PyCharm, IntelliJ,..)

1. Download Activity Tracker plugin from IDE Settings -> Plugins -> Browse Repositories or from https://github.com/dkandalov/activity-tracker
2. Restart IDE if it was running.
3. Enable "Poll IDE State" in settings



Step 2: Register a bot user to a Slack workspace

1. Go to https://api.slack.com/bot-users and click on "Create your Slack app"
2. Name the app (for example Indikom) and choose the workspace.
3. Click on "Bots" and "Add a bot user" and "Add bot user".

4. Click on “OAuth & Permissions” in the side menu and add the “incoming-webhook” permission scope
5. Click on “Install App” on the side menu and install the bot into the workspace.
6. Add the newly created bot user to the channel to be monitored by inviting it as a new user.

Step 3: Setup the application

Upon the initial startup of the application, the connection window on figure 1 will be displayed. If it is not displayed, it can be accessed by clicking on “General” and then “Connection settings” as seen on figure 2.

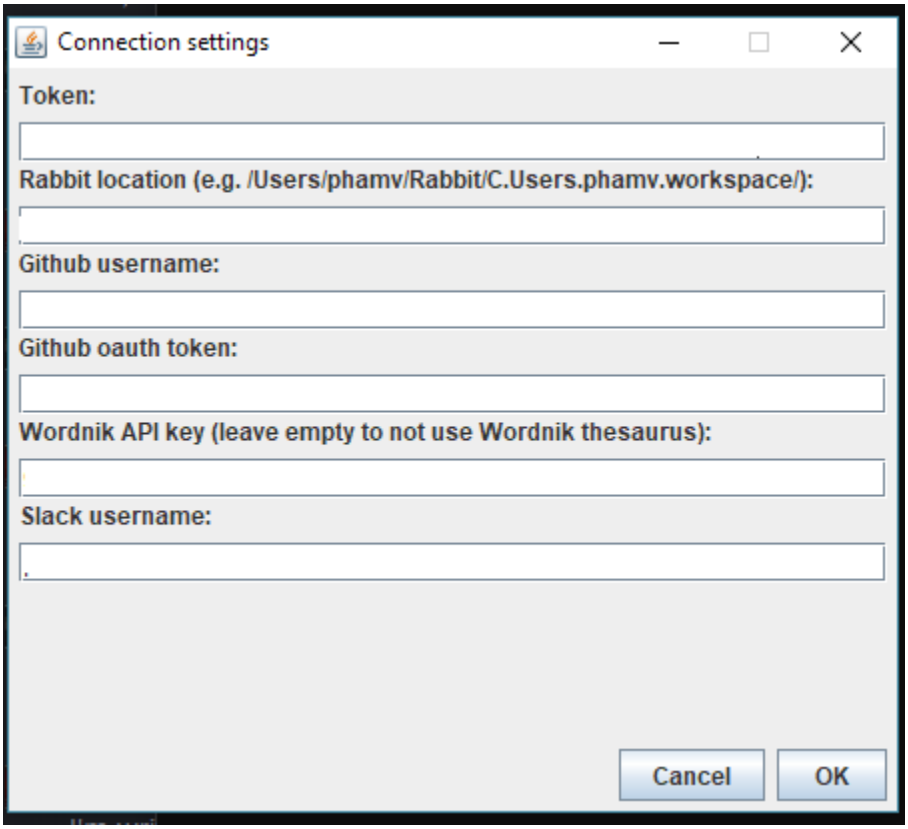


Figure 1: Connection settings

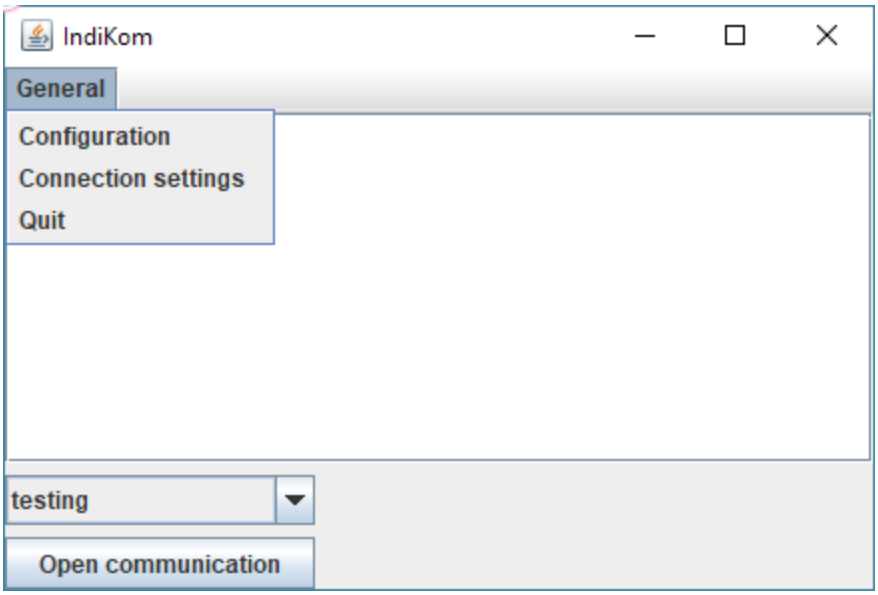


Figure 2: Accessing connection settings

To setup the application, enter the following credentials into the connection settings:

- **Token:** The Bot User OAuth Access Token of the Slack bot registered in step 2. It can be found by clicking on "OAuth & Permissions" on the side menu. It begins with xoxb-
- **User activity location:** the location where the activity tracking plugin setup in step 1 stores its files. An example path for Rabbit Eclipse is listed in Figure 1 for Windows users. For Jetbrains IDE users it is the location of the tracking log accessible in the plugin menu as seen in Figure 7.
- **GitHub username:** the GitHub username, if the user wishes to use GitHub activity tracking
- **GitHub oauth token:** the GitHub oauth token if the user wishes to use GitHub activity tracking
- **Wordnik API key:** the API key for the Wordnik thesaurus. For the purpose of this experiment, this key will be provided.
- **Slack username:** the username in the chosen Slack workspace to filter out the user's own messages.

Upon clicking "Ok" the application will save these settings and close.

Step 4: Using the application

Upon launching the application after initial setup, users will be greeted with the main application window seen in Figure 2. Users can choose the channel they wish to be monitored from the dropdown in the bottom left of the window, as seen on Figure 3.

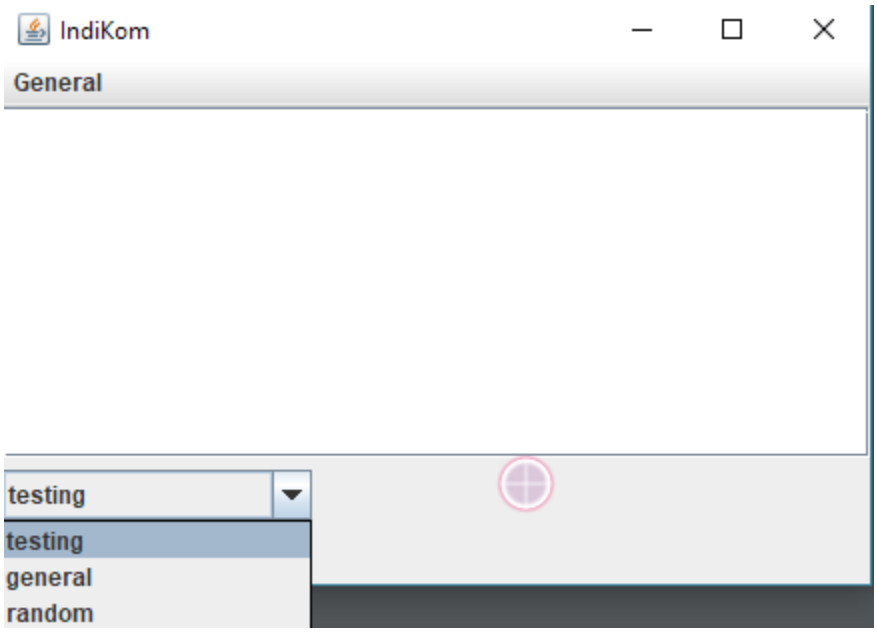


Figure 3: Channel selection

Users can start monitoring the channel by clicking the “Open communication” button in the bottom left after choosing a channel as seen in Figure 2. All incoming messages to the chosen channel will be displayed in the main window, as seen in Figure 4.

Figure 4: Messages in the main window

Messages that are deemed relevant to the current work context of the user will be additionally displayed in a transparent popup window as seen on Figure 5. Users can flag these messages as relevant or irrelevant by hovering over this notification window and pressing the left or right mouse button.

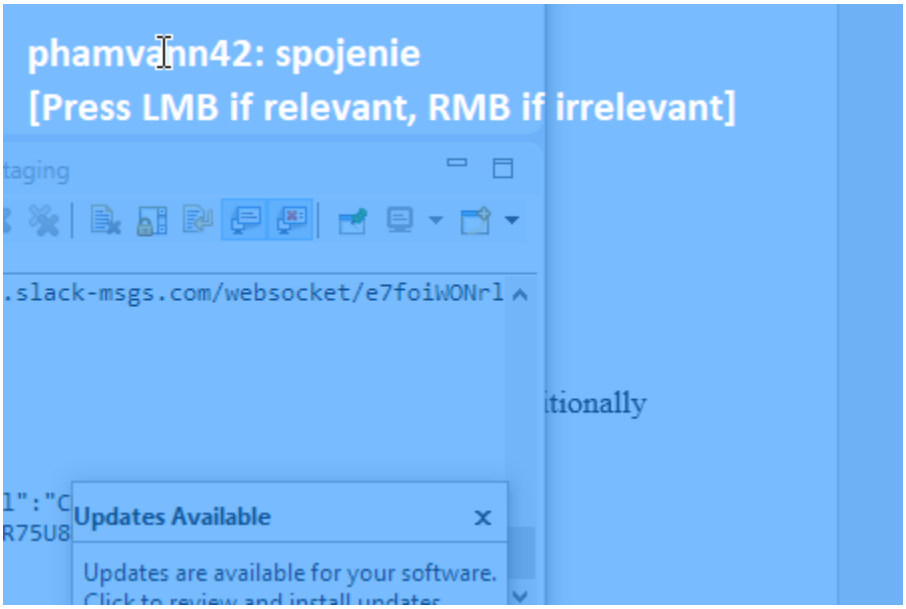


Figure 5: Message notification

The Rabbit Eclipse plugin setup in step 1 is used to track user activity in the form of artifacts. An update can be triggered by accessing the Rabbit tab at the bottom of the IDE and pressing the refresh button in the top right corner highlighted in red in Figure 6.

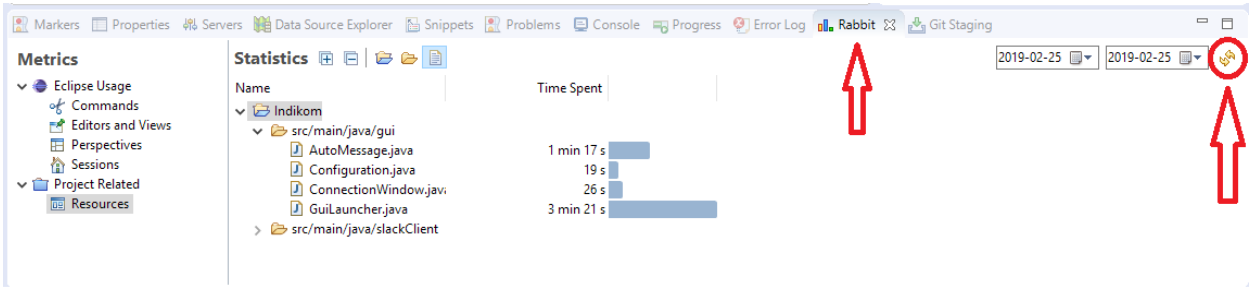


Figure 6: Rabbit plugin window

If it is the first time the application is launched that day, please work on your project for a short amount of time in order to build up work activity first. Then trigger an update in Rabbit and launch the application.

If using the Jetbrains line of products, you can access the Activity Tracker plugin by clicking on the text in the bottom right of the screen as seen on Figure 7. It is recommended to clear the log regularly, due to the quickly increasing file size. Logs can be cleared by hovering over “Current Log” and then “Clear Tracking Log”.

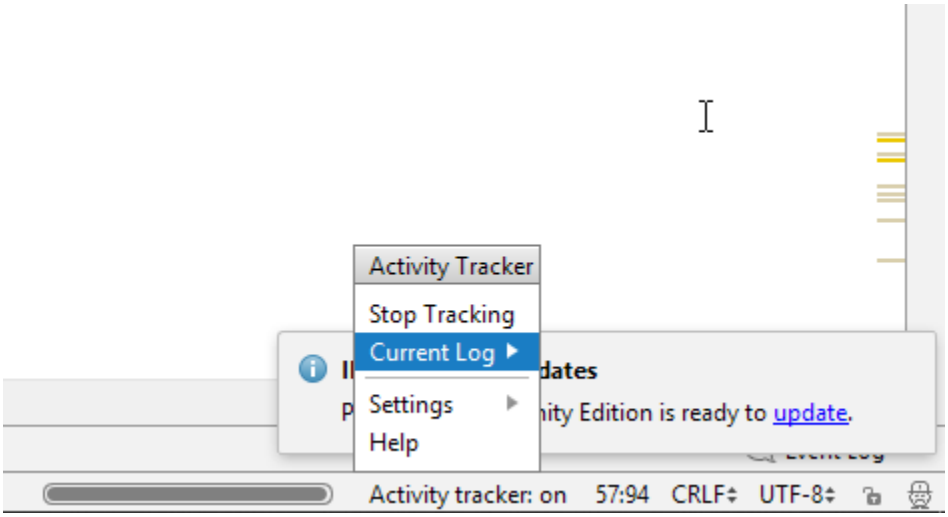


Figure 7: Activity Tracker for Jetbrains IDE