

## Travis Alerts Notifier

This PDF will guide you through the steps of setting up our application to work with any Slack account, with our application hosted on any server.

Our application is a pre-packaged .jar file which supports any environment supporting Java.

### Required Files

Our project folder is available at:

Pre-packaged .jar file: <https://s3.amazonaws.com/ctms-bucket/travis-alerts-0.2.1-MASTER.jar>

### Background Steps

1. Ensure a port is available on your chosen-server for both incoming and outgoing connections. If this is anything other than **port 80**, you will update the ***application.properties*** file within the project folder with your chosen port value. (*src > main > resources > application.properties*).
  - a. **Ensure that you recompile the project after doing this, with the command 'mvn package'**

### Setting up a new Travis Alerts Notifier server for Slack

1. Go to <https://api.slack.com/apps> and select 'Create new app'
  - b. Under "App Name", enter "Travis Alerts Notifier"
  - c. Under "Development Slack Workspace", select "mdsol"
  - d. Under "**Add Features and Functionality**", select "Incoming Webhooks"
    - i. Enable incoming webhooks.
  - e. Under "**Add Features and Functionality**", select "Slash Commands"
    - i. Create the following commands, with the following settings.
    - ii. **Command Get Status**
      1. 'Command' field – `"/getstatus"`
      2. 'Request URL' field – `http://[hostname]/command/getstatus`
        - a. **Ensure that you replace [hostname] with the servers actual hostname and include and port numbers on the end**
      3. 'Short Description' field – "Gets status of current branch."
      4. 'Usage hint' field – `"[repo] [branch]"`
    - iii. **Command Start Polling**
      1. 'Command' field – `"/startpolling"`
      2. 'Request URL' field – `http://[hostname]/command/startpolling`
        - a. **Ensure that you replace [hostname] with the servers actual hostname and include and port numbers on the end**
      3. 'Short Description' field – "Starts polling a new GitHub branch."
      4. 'Usage hint' field – `"[repo] [branch] [minutes]"`
    - iv. **Command Stop Polling**
      1. 'Command' field – `"/stoppolling"`
      2. 'Request URL' field – `http://[hostname]/command/stoppolling`
        - a. **Ensure that you replace [hostname] with the servers actual hostname and include and port numbers on the end**
      3. 'Short Description' field – "Stops a polling service."
      4. 'Usage hint' field – `"[repo] [branch]"`
    - v. **Command Configure**

1. 'Command' field – `"/configure"`
2. 'Request URL' field – `http://[hostname]/command/configure`
  - a. **Ensure that you replace [hostname] with the servers actual hostname and include and port numbers on the end**
3. 'Short Description' field – `"Configures a channel to work with Travis Alerts!"`
4. 'Usage hint' field – `n/a`
- f. Under **"Add Features and Functionality"**, select `"Permissions"`
  1. Scroll down to `"Redirect URLs"`
  2. Select `"Add New Redirect URL"`
  3. Enter `"http://hostname/configure"`
    - a. **Ensure that you replace [hostname] with the servers actual hostname and include and port numbers on the end**

### Configuring the server to listen for incoming connections.

1. Go to <https://api.slack.com/apps>
2. Select the app which you previously created.
3. Scroll down to `"App Credentials"`
4. Go to <https://developer.travis-ci.com/authentication> and follow the steps to generate an `"API token for a private project on travis-ci.com"`
5. Add the following environment variable, ensuring to replace [Token] with the previously generated token.
6. `export TRAVIS_TOKEN="[Token]"`
7. For Slack, add the following environment variables to the server, replacing their values with the values under `"App Credentials"`
8. `export TRAVIS_ALERTS_CLIENT_ID="[Client ID]"`
9. `export TRAVIS_ALERTS_CLIENT_SECRET="[Client Secret]"`
10. Download the pre-packaged .jar file to any server with Java installed.
11. Run the Java file with `"java -jar [java file name]"`

Note: By default, Travis Alerts Notifier listens for and accepts connections on port `'80'`. You can change the port number in the project folder under `src > main > resources > application.properties`

**Ensure that you recompile the project after doing this, with the command `'mvn package'`**