# User’s Documentation

*User’s Documentation of the program Discord Crypto Alert Bot*

*Programming in Java (NPRG013) at MFF UK.*

Author: Tomáš Pop

Date of issue: 18/05/2022

Version: 1.0

# Short Project Description

Crypto Alert Bot is an application written in Java which serves as a cryptocurrency alert service for Discord servers. Furthermore, it can hold a watchlist for each user on a particular server where the bot is deployed.

The application operates with Binance API containing around 1000 cryptocurrency pairs (symbols and their prices) updated with a little to none delay compared to the prices binance.com provides to regular users. Since Binance on its own lists way more than 1000 cryptocurrency pairs, it is recommended to use rather conventional cryptocurrency pairs or to check the availability by taking a look at https://api.binance.com/api/v3/ticker/price and CTRL+F the desired cryptocurrency symbol with its price.

# How To Use

In order to use the application, you may utilize the IntelliJ Idea IDE to run the gradle build within the IDE or there are gradlew (Linux) and gradlew.bat (Windows) scripts provided to compile and run the program from the command line.

To deploy the bot to the server of your choice, use the link provided:

https://discord.com/oauth2/authorize?client_id=923881722964213770&permissions=8&scope=bot

## Windows

### Prerequisities

For Windows users, it is highly recommended to use IntelliJ 2021+ (The author used IntelliJ Idea Ultimate 2021.2). In case, you do not want to compile the program from IntelliJ, no prerequisities are required since the gradle script takes care of it itself. It is expected that the user has Java 16 or 17 installed (in case of IntelliJ Idea, the configuration can be set up accordingly). Otherwise for the command line users, the installation of openjdk-16-jdk or openjdk-17-jdk is highly recommended.

### Installation

Installation process is made via gradlew.bat script in the root directory which takes care of all dependencies needed. It is only required to open the project folder in the IDE. The environment should install its dependencies. In case, the build does not run automatically, make use of Build >> Build Project or by CTRL+F9 and afterward to run Run >> Run 'Entrypoint' or by SHIFT+F10. Alternatively, within the interface, there are build and run symbols which can even ease the process [1].

For command line users, open the command line and locate to the root directory of the project and run "gradlew.bat run", all dependencies should get installed in the initial run. The expected command line output signaling that the bot is running is added in the attachment [2]. In the Discord server where you deployed the bot.

## Linux

### Prerequisites

The application can be launched via IntelliJ Idea IDE (although it was not tested by the author) and the exact process should be as in the Windows Installation part. The installation part below is targeted for the command line users.Installation

The installation process is analogical to the Windows Installation part in the second paragraph, the only difference is to use the commands using ./gradlew, instead of gradlew.bat. In the attachment [3], there is an expected command line output signaling the bot is running.

## Features – Expected Behaviour

Each user on the server where the bot is deployed should be able to run the commands with a specified leading symbol (to detach from the regular conversation). The dialog between an user and the bot is rather simple and consists of a variety of commands which contain zero, one or two additional arguments. The main functionality which is expected to used the most is that each user can create own watchlist of cryptocurrencies and manage own alertion storage. Otherwise, all currently supported commands by the bot (!help command) are shown below:

## Hardware/Software Requirements

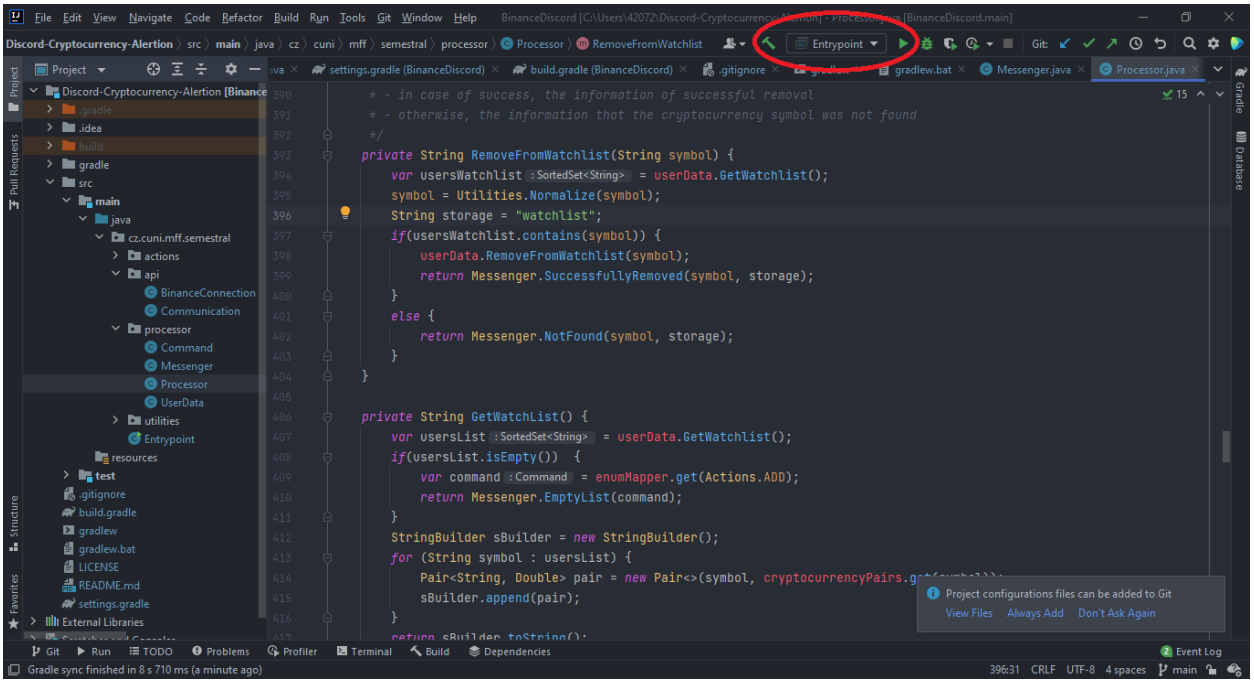
- For an end user, there are no special requirements although it is demanded to be an user of either of Windows or Linux operating system

- Note that you should be connected to the internet throughout the whole process since the bot (currently) only runs from your machine, it will not be able to get new data concerning the Binance API. Furthermore, in case of longer disconnection, the bot disconnects too (although it takes time before it really shows with the online x offline icons provided) and wipes currently held storages so it is either suggested to host the bot on a machine with stable connection or another option could be to deploy the application to a cloud service

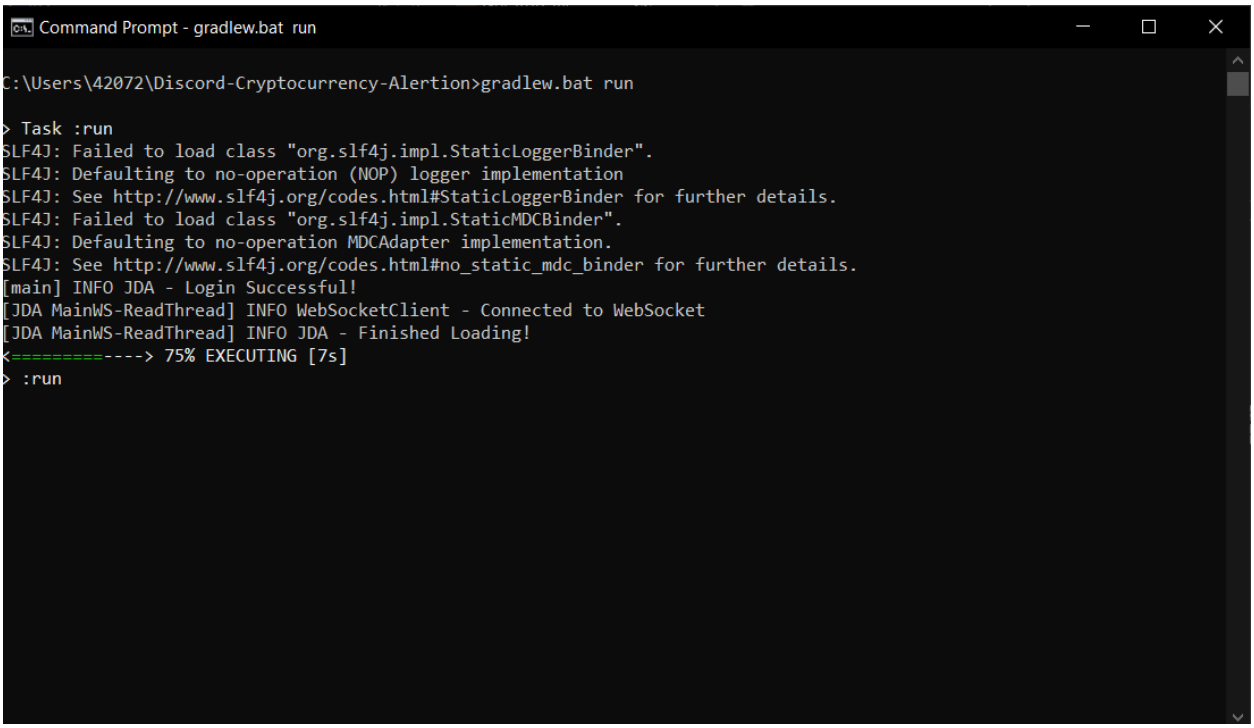
- The application was compiled and run using a device with the CPU AMD Ryzen 7 4800H, RAM 16 GB, OS Windows 10 Home (64-bit) with WSL Ubuntu 20.04.

# Attachments

[1]: Windows – IntelliJ Idea – Compile and Run (afterward, the bot should be running in the Discord servers where it was deployed)




[2]: Windows – Command Line – Compile and Run (afterward, the bot should be running in the Discord servers where it was deployed)



[3]: Linux – Command Line – Compile and Run (afterward, the bot should be running in the Discord servers where it was deployed)