INFOSAFE
Installation Guide

InfoSafe

Seed Analytics

—

Team Name:	**FrAgile**
Team Members:	Christof Steyn
	Chris Mittendorf
	Karel Smit
	Yané van der Westhuizen
	Alistair Ross
Team Contact:	fragile.cos301@gmail.com

_____

Table Of Contents

_____

Introduction

Welcome to the Infosafe program!

Before you are able to begin using the system there are a few things you'll need help with. This Document will show you step by step how to install all aspects of the program so that you can start using Infosafe.

In this installation guide you will:

- Check if your machine and network are compatible with the system
- Install the necessary software needed to run the system
- Be guided on how to clone and make a copy of the Infosafe repository on your local machine
- Be guided on how to install the system on your local machine and prepare it for use
- Understand the necessary requirements to deploy and host your version of the system

Infosafe is a Java Spring-Boot application that uses a Javascript-React frontend. These parts of the systems are independent and will have their own supporting software and installation procedures. The system is set up to be deployed and hosted on AWS as well as have its database and storage capabilities on AWS too.

Once your installation and setup is finished and you want to get to know the system, you can access the user manual here in order to help you and familiarise yourself with the InfoSafe system.

_____

Hardware and Software Requirements

This section will provide information on the current hardware and software requirements of the system as well as some limitations.

Operating System

As Infosafe is still in development it is currently only supported for use on Microsoft Windows. The system will run on Windows 10 and Windows 11. If you are currently not running either of these Windows versions on your machine you can upgrade here and follow the installation instructions on their website.

Hardware and Network Requirements

As of this current stage in project development there are no real hardware requirements but we can recommend some of the following that will make installation and program operability on acceptable levels for the user:

- Memory - at least 4gb RAM
- Storage - at least 5gb available for project files. Either HDD or SSD will be fine
- Processor - any modern CPU will run the program to completion and at acceptable levels (Intel i3/Ryzen 3 or greater)
- Internet - a stable internet connection is recommended to maintain communication with the deployed website and its database. Any LTE or Fibre connection would be more than adequate.

Integrated Development Environment (IDE)

Most IDEs would work for project installation however we would recommend Visual Studio Code or Jetbrains' IntelliJ. Visual Studio Code detects and allows you to install plugins that will greatly help with the installation process and IntelliJ comes standard with most of the required features but also allows seamless plugin installation.

_____

Prerequisites and Supporting Software

As stated above, Infosafe is a Java Spring-Boot project on the backend side and utilises Javascript React on the front end. The following are needed to run the system.

Backend

The Backend runs in Java and is needed to make changes and compile the system software. A guide to install or update Java can be found here. Please make sure this is completed before moving onto the next step in the installation process.

As it is also a Spring-Boot application, Maven is required to install and manage the project dependencies for the Infosafe system. You can download and find an installation guide to Maven here.


Frontend

The frontend is created using Javascript React. React (formally ReactJS) builds the pages dynamically when accessed via the web. In order for ReactJS to work, you will need Node.js installed on your system. Node.js is a server which will help us to run the React code. Node.js windows installer can be downloaded here. Simply download the installer package and run it to install Node.js to your machine.

_____

Cloning and Creating Your Own Repository

Infosafe uses GitHub as its version control software and so it is available for cloning and local installation. You will need to download and install git if it is not currently on your machine, you can download git here. Please note you need to use chocolatey to install git. Chocolatey can be downloaded here. There is also a simple guide to help with the installation.
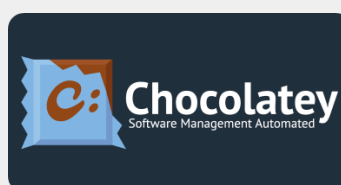
Git Installation

Once Chocolatey is installed you can open up Command Prompt or Powershell by clicking start on the windows taskbar and searching for either terminal. Make sure to right click on the application and click "Run as administrator". Once the terminal has opened, type the following:

```
> choco install git
```

Wait for the package to download and verify. You will be prompted to accept changes to be made, simply type **y** (for yes), and wait for the installation to complete. Git has now been installed and you can close the terminal.

Creating Your Own GitHub Repository

You are now able to create your own repository that you can then clone the Infosafe system into, this can be done 2 ways. Firstly it can be done using Command Prompt, Powershell or the terminal in the IDE you're using. Secondly it can be done using the GitHub Desktop Application. We will explore both these options. Please note you will

_____

need a GitHub account to create a repository. If you do not have an account you can create one here.

Terminal Repository Creation

Please note to create a repository this way you will need to download and install GitHub CLI from here. Once that is finished, open any terminal you are using and navigate to the directory where your local repositories will be. E.g:

```
> C: \Users\username\documents\GitHub\<your-repo-name>
```

Once there, type and execute the following commands in order to create and push your local repository to GitHub.

This will convert your current working directory to a GitHub Repository:

```
> git init
```

Next step is to stage the files so they can be loaded to the repository. If the directory is empty, add a .txt file with a description:

```
> git add .
```

Next step is to commit the files:

```
> git commit -m "first commit"
```

_____

You'll now need to authorise the repository to your account using GitHub CLI. Execute the the following command and follow the prompts to login and authorise the repository:

```
> gh auth login
```

Finally, this command will create a public GitHub repository with the name "My-First-Repo". It will create the repository using the the directory we have been working in:

```
> gh repo create my-first-repo --public --source=. --remote=upstream --push
```

## GitHub Desktop Repository Creation

The following steps will show you how to create a repository using the GitHub desktop application. If you do not have it installed you can download and install it from here. Once installation has completed, the first thing you will need to do is to login to your GitHub account. Click *File>Options*:
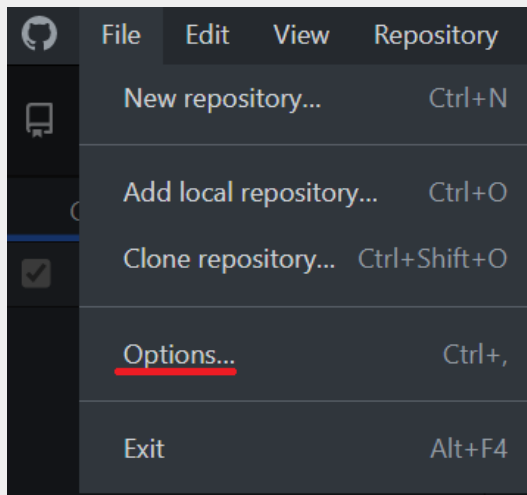
Under the Accounts tab click *Sign in* under the GitHub.com heading and fill in your credentials to authenticate GitHub Desktop to your account:
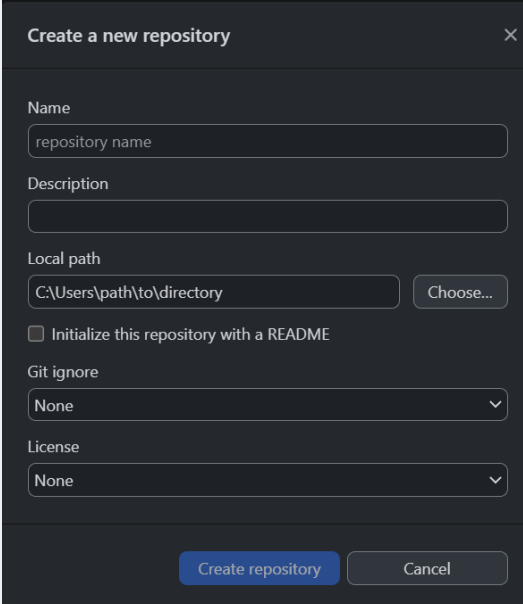
_____
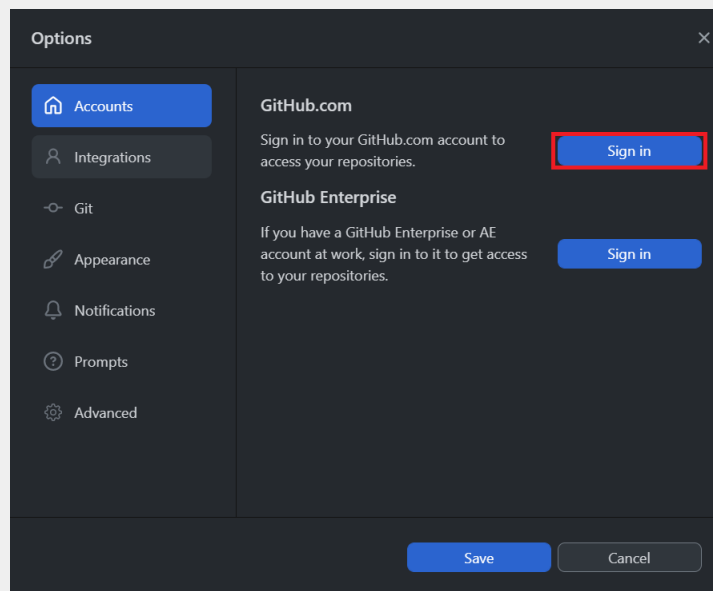
Now you are able to create a local repository. Click on *File> New repository...* :
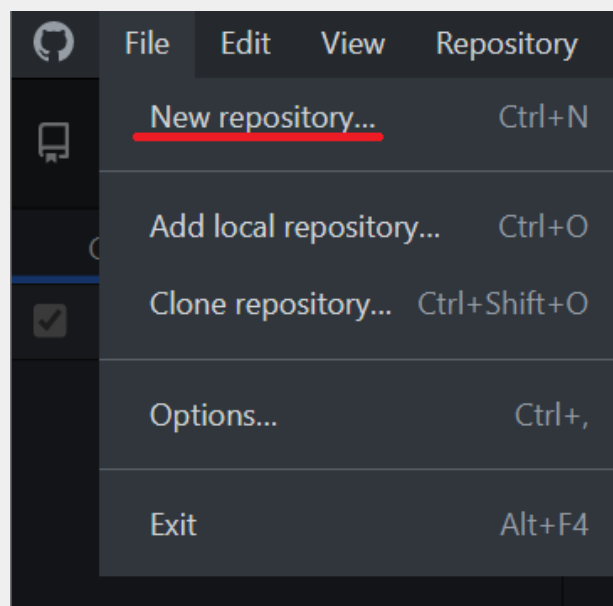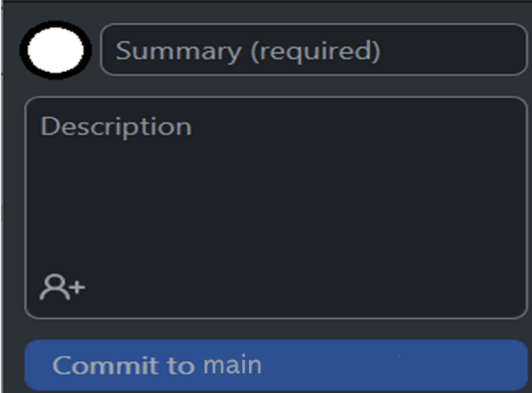

On the pop-up that opens, give your repository a name and description. Then specify the path to the directory you will be working in. Select the checkbox that says "initialise this

_____

repository with a README". Lastly leave the default options for the Git ignore and the licence. Then simply click *Create repository*:

GitHub desktop will now be working in this repository directory. Any changes you make to files and folders in this directory will be tracked on the application. To commit a file simply add a summary in the *summary* section on the bottom left and additionally you give more information in the *description* section beneath it. The click commit and your changes will be committed to the repository:

_____

Cloning Infosafe

Now that you have created your repository you are able to clone the Infosafe project to it. As the same as above we will show you how to do it in a terminal and using the GitHub desktop application.

Cloning Using A Terminal

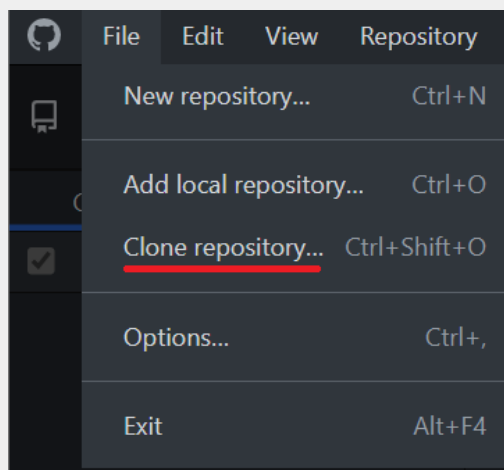Execute the following command to clone:

> git clone https://github.com/COS301-SE-2023/InfoSafe

You can then commit it to your repo and are then complete:
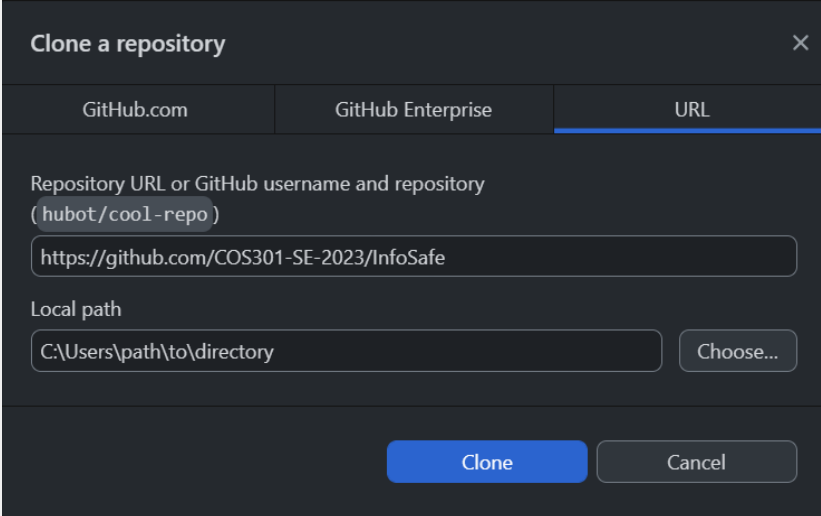
>git commit -m "cloned Infosafe"

Cloning Infosafe Using GitHub Desktop

To clone the infosafe repository all you need to is click *File> Clone repository...* on the main page of the desktop application:

On the opo-up that opens, under the URL tab, in the URl Section you must add the infosafe URL (https://github.com/COS301-SE-2023/InfoSafe), and then specify the directory where you wish this clone repository should be. Then simply click *clone.*

_____

Clone a repository   ✕

| GitHub.com | GitHub Enterprise | URL |

Repository URL or GitHub username and repository
(`hubot/cool-repo`)

https://github.com/COS301-SE-2023/InfoSafe

Local path

C:\Users\path\to\directory   Choose...

                                        Clone   Cancel

Any changes now made in this directory will be tracked by the application.

_____

Installation and Setup

Now that you have your local copy of Infosafe setup, you will need to run a few commands to install and then run. Please note the examples below use IntelliJ IDE and other IDEs may differ slightly.

Backend

Firstly you'll need to navigate to the backend application, which can be found at:

backend/infosafe_backend/src/main/java/com/fragile/infosafe
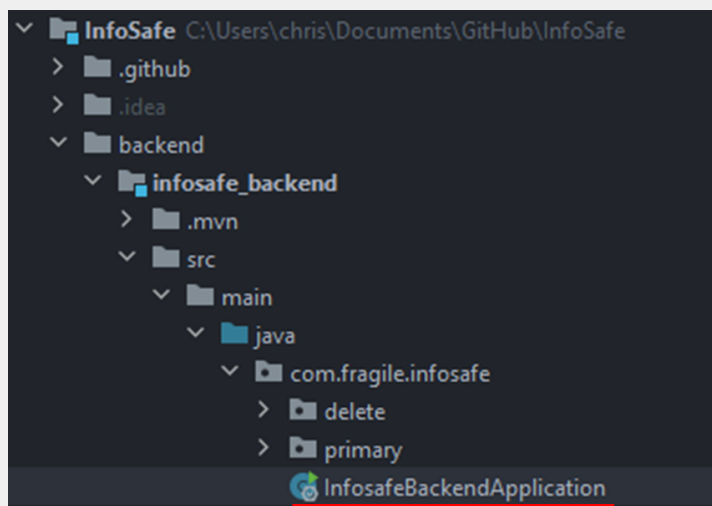
Double click on the file to open it in a window. Navigate to line 16 - public class InfosafeBackendApplication that is located underneath line 15 - @SpringBootApplication declaration and click the green arrow to the left to run the backend application.

The backend application is now running and provides functionality for the frontend.

_____

Frontend

To install and start the frontend, open a terminal of your choice, either in an IDE or Command Prompt or Powershell. If you're working in an IDE the current working directory will be the root folder. If you're not using an IDE make sure to change your directory to the Infosafe root folder.

Then type the following to change the directory to the frontend directory which will need to be installed and then run:

```
> cd frontend/infosafe_frontend
```

Now that you are in the correct working directory you can execute the following to install the frontend packages:

```
> npm install
```

When the installation completes, execute the following code to start the frontend side of the system. It will launch a web page in your default browser and you will be able to use the system. If a browser does not automatically open you can open a browser and type localhost:3030 into the address bar and hit enter to open the Infosafe login page:

```
> npm start
```

If you are a new user and are unfamiliar with the Infosafe system you can access the User Manual to learn how to effectively use the system.

_____

Deployment and Hosting

The Database, Deployment and Hosting is all provided and maintained by AWS (Amazon Web Services) and is thus all cloud based. You will need to create your own AWS account if you wish to host your system and database in a similar instance.

Database

The database is an instance of RDS (Relational Database Service), more specifically this instance is an MySQL Community engine database. You will need to create your own database on AWS RDS if you want your system's database to be in the cloud. You can also create a local database for testing purposes if you wish. Please follow the below instructions to set up a connection to your database instance and create the two virtual databases used to store the system data.


Using any database editing IDE like MySQL Workbench (download [here](#)) or DataGrip (download [here](#)), create a connection to your specific database and execute the following queries in a schema console. This will create the two databases that are needed for the infosafe system

```
> CREATE DATABASE infosafe_database;
```

```
> CREATE DATABASE secondary_database;
```

_____

Hosting

Hosting is made available using an EC2 (Elastic Compute Cloud) which keeps the current instance of the system online and runs it to be made available via website on a browser.
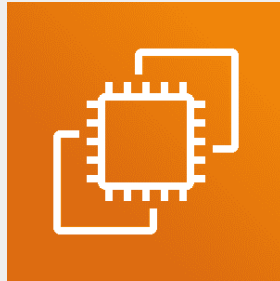
Deployment

As stated above hosting takes place on AWS. A deployment script was set up through GitHub actions to to automatically deploy the system to AWS. The deployment script specifically only triggers on a push to the dev branch or a pull-request into the main branch. The script first sets up Node.js for the frontend, it then installs all the dependencies for the frontend and then builds the frontend. It then sets up JDK 20 for the backend, builds the backend and finally executes the backend.

_____
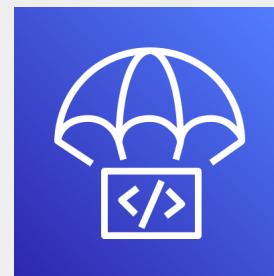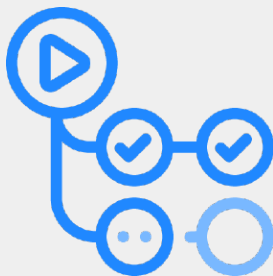
First Time Use

Due to the secure nature of the system and the importance to security, it's important to follow these steps for first time use.

Creating The Admin Role and your First Role

Creating the admin role and first profile is easy enough. Navigate to the backend application located at: backend/infosafe_backend/src/main/java/com/fragile/infosafe
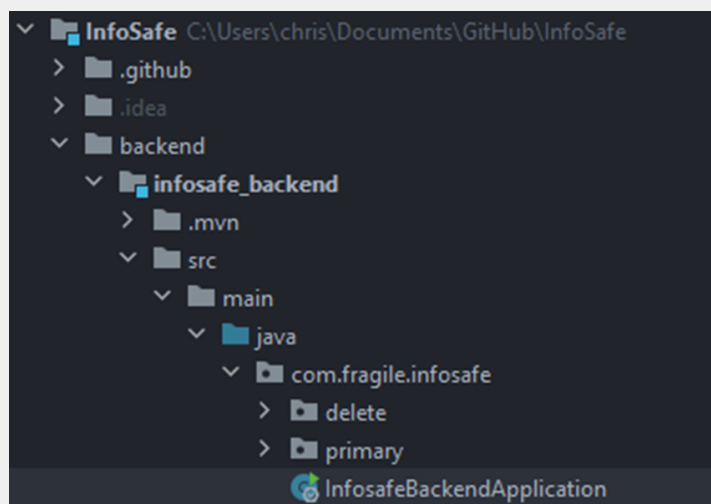
Uncomment all the lines beneath the main function and add your own personal details (name, surname, email address and password on lines 57-60) to the relevant sections and then run the backend application. This will create your profile in the database and allow you to then login to the system with Admin role permissions and use the system to its entirety.

Once this has run it will have created the role and your admin in the database you have selected. You can now use the system. Navigate to https://infosafe.live to login.

```java
package com.fragile.infosafe;

import com.fragile.infosafe.primary.auth.AuthenticationService;
import com.fragile.infosafe.primary.repository.RoleRepository;
import com.fragile.infosafe.primary.model.Role;
import com.fragile.infosafe.primary.requests.RegisterRequest;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.Bean;

import java.util.Collections;

// Chris Mittendorf +1
@SpringBootApplication
// CodiumAI: Test this class.
public class InfosafeBackendApplication {

    // Chris Mittendorf +1
    // CodiumAI: Test this method.
    public static void main(String[] args) {

        SpringApplication infosafe = new SpringApplication(InfosafeBackendApplication.class);
        infosafe.setDefaultProperties(Collections.singletonMap("server.port", "8080"));
        infosafe.run(args);
    }

    //Uncomment the below code if this is your first time use
    //This will create an admin role in the system and create your first profile

    @Autowired
    private RoleRepository roleRepository;

    // new
    @Bean
    // CodiumAI: Test this method.
    public CommandLineRunner cr (AuthenticationService service){
        return args -> {
            //Check if it exists
            Role admin = roleRepository.findByRole_name("ADMIN");

            //Create role if it doesn't exist
            if (admin == null){
                admin = new Role();
                admin.setRole_Name("ADMIN");
                admin.setPermissions(134217727);
                roleRepository.save(admin);
                System.out.println("Admin Role Created");
            }
            else{
                System.out.println("Admin Role Already Exists");
            }
        };
    }

    // new
    @Bean
    // CodiumAI: Test this method.
    public CommandLineRunner commandLineRunner(AuthenticationService service) {
        return args -> {
            Role adminRole = roleRepository.findByRole_name("ADMIN");

            var admin = RegisterRequest.builder()
                    .first_name("John")
                    .last_name("Smith")
                    .email("john.smith@gmail.com")
                    .password("Password@1234")
                    .role(adminRole)
                    .build();

            System.out.println("Admin token: " + service.register(admin).getAccessToken());
        };
    }
}
```

_____

Appendix A

Below is a detailed table of all dependencies required in order to run, update and use the Infosafe system.

Name	Required Version
Browsers	
Chrome	Latest
Firefox	Latest
Edge	Latest
Operating System	
Windows	10 or newer
Hardware	
CPU	Any modern Multi-core CPU Intel i3 or greater Ryzen3 or greater
Memory	4gb or more of RAM
Storage	At least 5gb (HDD or SSD)
Software	
Java	20 or newer
Maven (Apache Maven)[1]	3.9.3 or newer
Node.js	18.17.0 or newer
NPM[2]	9.6.7 or newer
Git	2.41.0 or newer

---

[1] Please see Appendix B that shows required Maven dependencies and versions required
[2] NPM is automatically installed and updated when Node.js is installed

_____

Appendix B

Below is a list of Maven dependencies required to run the Infosafe system. These should automatically be declared in the pom.xml file in the backend/infosafe_backend directory. However if there are any issues regarding the Maven dependencies you may refer to this appendix to see which dependencies are required as well as their version. Note dependency versions that are automatically applied at runtime by the software provider are denoted with a " - ".

Dependency Name	Required Version
Spring boot starter data jpa	-
Spring boot maven plugin	-
Spring boot starter mail	-
Spring boot starter web	-
Spring boot starter test	-
Spring boot starter mail	-
Spring boot starter security	-
Spring security test	-
Spring security core	-
Spring Cloud Starter AWS Secrets Manager Config	2.4.4
AWS SecretsManager JDBC	1.0.11
AWS Java SDK SecretsManager	1.12.411
AWS SDK Java	2.20.135
MySQL Connector Java	8.0.32
Junit	-
Javax persistence API	2.2

_____

JJWT API	0.11.5
JJWT IMPL	0.11.5
JJWT Jackson	0.11.5
GSON	-
Lombok	-