

# **EU TRUST SERVICES DASHBOARD APP Manual**

Made by:  
**Tommaso Gabrieli**  
**Luca Fantin**  
**Francesco Colla**  
**Alessandro Viespoli**

# INDEX

|                                     |          |
|-------------------------------------|----------|
| <b>TECHNICAL OVERVIEW</b>           | <b>3</b> |
| <b>REUSED SOFTWARE</b>              | <b>4</b> |
| <b>INSTRUCTION MANUAL</b>           | <b>5</b> |
| Base install                        | 5        |
| Advanced install                    | 5        |
| Possible issues and troubleshooting | 6        |
| Usage                               | 6        |

# TECHNICAL OVERVIEW

The user has a sole access point to the program, which is a class with a main method that handles the various UIs. Each is controlled by its own subsystem, be it composed of one class or multiple classes interacting with each other. These subsystems can be identified as:

- The search subsystem (from now on referred to as SearchSys)
- The result screen subsystem (from now on referred to as ResultSys)
- The loading screen subsystem (from now on referred to as LoadingSys)
- The error messages subsystem (from now on referred to as ErrorSys)

During the program execution, the LoadingSys is the first one to be called, which makes the user wait for the building of the SearchSys. Once SearchSys is ready, it takes control of the program and allows the user to select the criteria, warns them of any anomalies with the selected criteria (nothing selected across all lists, nothing selected in some lists, invalid criteria selected) and performs the search. Once it has been completed, ResultSys takes over, displaying the search results and displaying the information about the services the User clicks on. Once the User moves on from this interface, the control can either go back to SearchSys directly or be transferred to LoadingSys while SearchSys is built anew. At any time, any alerts or errors thrown by these subsystems are displayed on screen by ErrorSys.

By identifying another partition of the system, we can find an instance of the Model-View-Controller (MVC) architectural pattern: in fact, we could divide the program into three main subsystems:

- The model: encapsulates the core information and data about services, providers, etc.;
- The view: displays the information to the user, through various UIs for selecting criteria and displaying search results;
- The controller: handles the user input and maps it to model updates, by creating the checkboxes to select the criteria, updating the displayed criteria depending on the selected ones, performing the searches and getting the data from the trust services API.

## REUSED SOFTWARE

Our program reuses a vast array of external pieces of software, most notably to create the graphical interfaces. They were built with JavaFX, a software platform used to create desktop and web applications. In particular, this program makes use of several classes and methods from its libraries to create all the graphical elements (CheckBox, Stage, Scene, etc.).

Across the whole program, we used a number of data structures, which provided more flexible containers with more useful methods than normal arrays, which were used anyways in some instances. These data structures were provided by the java.util library: Vector, Map, SortedMap, List.

Finally, the unit testing was performed with JUnit, a testing framework for Java which provides a whole toolbox to perform such tests, including automatically generated test reports and specific methods for verifying the conditions upon which the tests were based.

# INSTRUCTION MANUAL

## Base install

1. Check if you have the Java Runtime Environment installed in your system. In Windows by typing “Java” in your search bar, if a “Configure Java” app pops up your pc is ready; in Linux typing the command `java -version`
2. This step is required only if you don't have Java installed in your system. Download the installer from [here](#) and follow the instructions provided.
3. Install the latest version of the JDK needed to run the jar file at [this link](#) choosing your os.
4. Download the .jar application file from this [link](#).
5. Double click the EU Trust Service Dashboard icon to launch the application.

If you're still experiencing troubles please refer to this short [video tutorial](#) covering the installation process on Windows.

## Advanced install

The entirety of our app is open source and browsable. Another way to launch the program is to compile it from scratch even though this method is not recommended for users that are not interested in the source code. For this kind of launch refer to the same steps as the base install up to step 4, and then:

1. Open an IDE of your choice, like IntelliJ and Eclipse
2. Open the extracted folder as a project in your IDE
3. Follow the steps to build and run in your IDE
4. Make sure you include the json ([go here](#), click the *download json.jar* button, unzip the downloaded file and add the .jar file included) and guava ([go here](#), click the *download guava-gwt.jar* button, unzip the downloaded file and add the *guava-xxx-jre.jar* file) libraries in your project

These steps aren't very detailed as every development environment is different and may have different steps in order to run the source code.

## Possible issues and troubleshooting

We noticed that the zoom function on the operating system prevents the app to work correctly. If at launch you experience a half-blank window in order to solve this problem you have to set the zoom at 100% to do so:

- Open the Settings app on your pc;
- Click the "System" tab and enter the "Display" settings;
- Under scale and layout in the drop-down menu labelled "Change the size of text, apps, and other items" select the 100% option.

We have successfully tested the .jar file on Windows and Linux, so if you experience trouble opening and running it try checking your version of the JRE (it must be 17 or forward). We have not tested the .jar file on MacOS.

## Usage

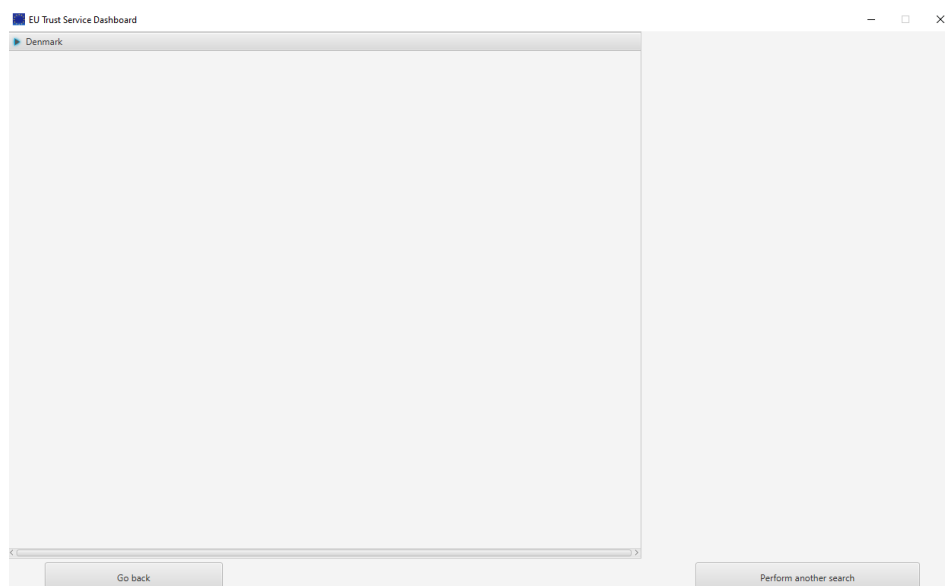
The screenshot shows the 'EU Trust Service Dashboard' application window. It features four main sections for selecting search parameters:

- Countries:** A list of countries and regions with checkboxes, including Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, European Union, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Liechtenstein, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, and United Kingdom.
- Providers:** A list of providers with checkboxes, including A-Trust Gesellschaft für Sicherheitssysteme im elektronischen Date, Bundesamt für Eich- und Vermessungswesen, Datakom Austria GmbH, PrimeSign GmbH, Rundfunk und Telekom Regulierungs-GmbH, Swisscom IT Services Finance S.E., e-commerce monitoring GmbH, Belgian Mobile ID SA/NV, CONNECTIVE, Certipost n.v./s.a., Connect Solutions, DigiCert Europe Belgium B.V., GlobalSign NV/SA, IPEX, Kingdom of Belgium - Federal Government, Portima s.r.l. c.v.b.a., SA UNIFIEDPOST, Society for Worldwide Interbank Financial Telecommunication SCR, Unversign, Zetes SA/N.V., BORICA AD, Bankservice PLC, Eurotrust Technologies JSC, and InfoNotary PLC.
- Types Of Service:** A list of service types with checkboxes, including QCertESeal, CertESeal, QCertESig, WAC, QWAC, CertESig, Timestamp, QTimestamp, QValQESig, QValQESeal, QeRDS, QPresQESig, QPresQESeal, NonRegulatory, GenESig, and CertUndefined.
- Service Statuses:** A list of service statuses with checkboxes, including withdrawn, granted, recognisedatnationallevel, and deprecatedatnationallevel.

A 'Search' button is located at the bottom right of the dashboard.

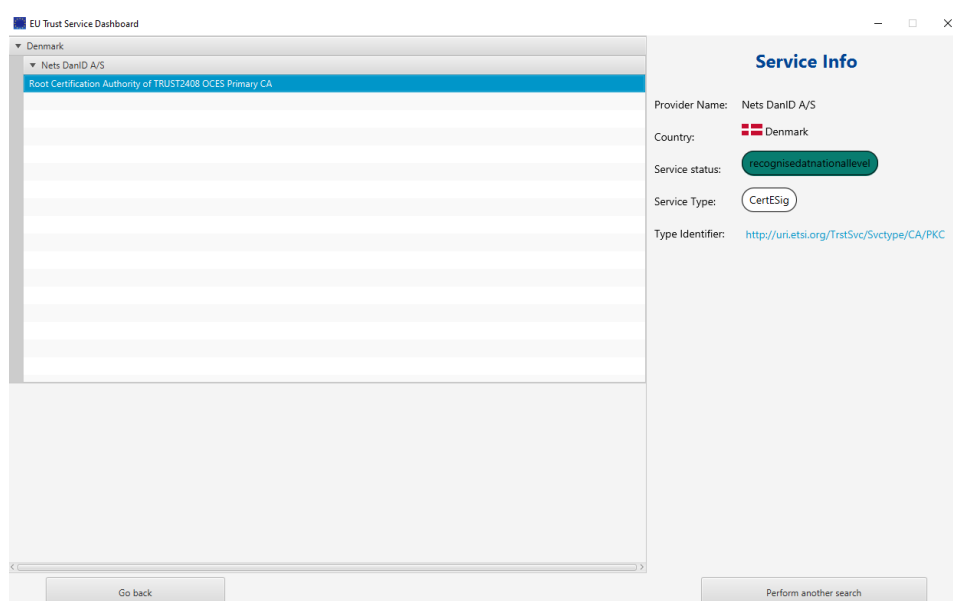
At launch, the app presents four lists with titles telling the user what the elements refer to. To select a search parameter simply click the checkbox at its left.

Once you're done adding search parameters click the "Search" button to perform the search.



At this point, the app changes the scene, presenting a list containing the counties and providers that match the selected criteria. When done consulting click the "go back" button to continue searching with your current parameters or "perform another search" to fetch new data from the APIs and reset all the filters. When first opened, all the lists are closed so you need to click the arrows near a name in order to open them.

If the selected filters conflict with each other the program will highlight the trouble, generating an alert telling the User that the conflicting criteria will automatically be discarded; at that point, the User can either go back and change the criteria or proceed with the search.



When a service is double-clicked its details will appear on the right side of this window as shown here.