Installationsguide for PasswordManager

Prerequisites

System requirements

To successfully install and run the Password Manager, ensure your system meets the following requirements:

Operating system: Windows 10/11
Java Runtime Environment (JRE): version 21 or above
.NET Core SDK: version 6.0 eller nyere
Web browser: Google Chrome

Required software and tools

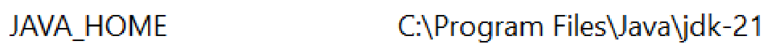
Install the following tools and software before proceeding with the installation:

JetBrains Rider (for frontend development)
JetBrains IntelliJ (for backend development)
Java Development Kit (JDK): Version 8 or newer
Apache Maven (for building the backend)
Git (optional but recommended for cloning and managing the repository)

Environment Configuration

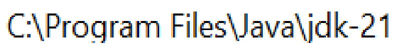
- Add the Java PATH to Windows environment variables to ensure Java is accessible

User Variables:



JAVA_HOME	C:\Program Files\Java\jdk-21

System Variables -> Path:



C:\Program Files\Java\jdk-21

- Add Maven to Windows environment variables to ensure maven is accessible through the terminal.

Installation and running the source code

To run the Password Manager, follow the appropriate steps depending on whether you are opening the project from a local file or cloning it from a repository. For opening the project from a local file continue to *Installing and Trusting SSL Certificates.*

Clone the repository with Git

1. Open a terminal or command prompt
2. Navigate to the directory where you want to clone the system
3. Clone the project repository using Git:

```
git clone https://github.com/KatAamand/SEP3_PasswordManager.git
```

Warning: Ensure that the cloning proceeds without error. It might throw a "Filename is too long" exception, if placed deep inside the file system.

Proceed to next step Installing and Trusting SSL Certificates

Installing and Trusting SSL Certificates

To ensure secure communication between the client and server, SSL certificates must be installed and trusted.

Steps to Install/Trust the Certificate

To enable the client to trust the server, the public key must be installed as a trusted certificate on the client machine:

1. Locate the `` `backend-cert.cer` `` file in the file explorer and double-click it, the file is located in the `` `src/main/resources` `` folder of the **SEP3_PasswordManager/PM_Backend/LoadBalancer** directory.
   If the file explorer asks you to choose an application, then select "Crypto Shell Extension" and press "Just once".
2. In the popup window, click **Install Certificate**.
3. Choose **Local Machine** (administrative rights required).
4. Select **Place all certificates in the following store** and choose:
   - ***Trusted Root Certification Authorities*** *(Rodnøglecentre, der er tillid til).*
5. Complete the installation process by clicking **Next** and the **Finish.**

First-Time Browser Access with Self-Signed Certificates

Since the certificates are self-signed, most browsers will initially show a security warning. Follow these steps to proceed:

1. When the browser shows a security warning, click the **Advanced** link.
2. Click **Proceed to [URL] (unsafe)**.
3. The browser will temporarily accept the certificate, allowing access to the Password Manager.

Running the Backend

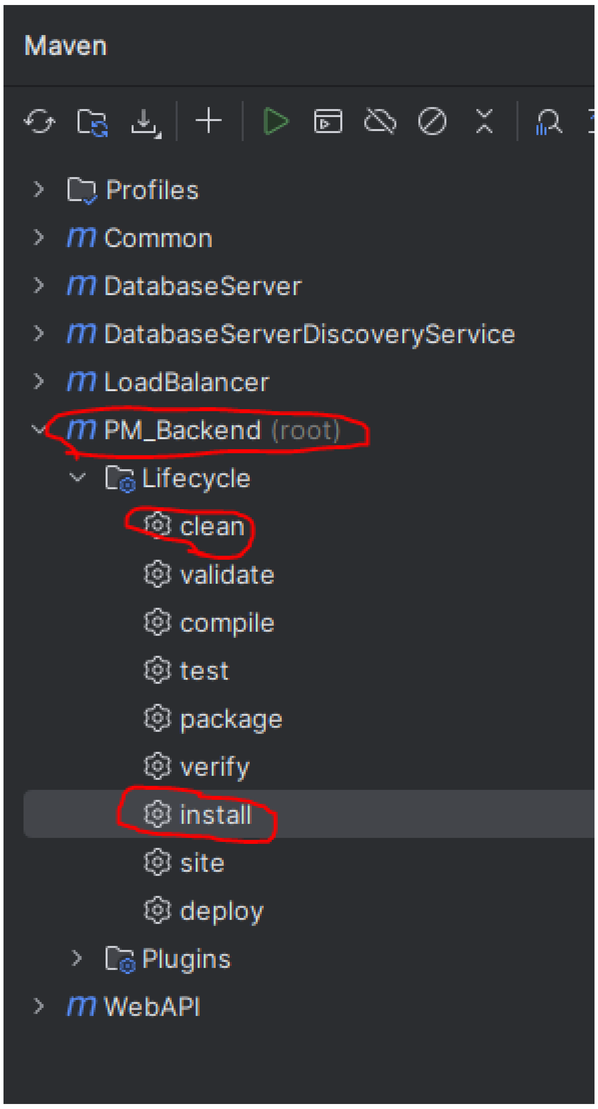
1. Open the cloned repository in JetBrains IntelliJ
2. In the terminal navigate to PM_Backend

```
cd PM_Backend
```

3. Build the backend using Maven

```
mvn clean install
```

(Optionally, use the built in Maven menu in IntelliJ)



4. Navigate to Services and choose Spring Boot
5. Run the following:
    - LoadBalancerApplication
    - DbDiscoveryServiceApplication

If DbServer is not properly started by DbDiscoveryService, run it manually

Warning: If an error is written in the loadBalancers console, relating to “Error cannot access” the .jar file, then move the entire installation to a folder within your local user (i.e. C:\Users\), deeper nested folder - since access rights at higher levels might interfere with the proper automatic launch of the .jar files in LoadBalancer and DbServerDiscoveryService.

Running the Frontend

1. Open the cloned repository in JetBrains Rider
2. When asked about “Select a Solution to Open”, choose “PM_frontend.sln” and press ‘Open’
3. In the terminal, run:

```
dotnet restore
```

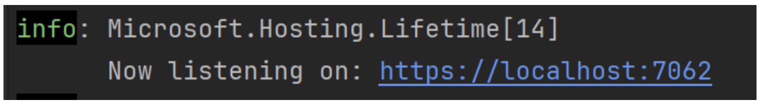
4. Navigate to BlazorUI

```
cd BlazorUI
```

5. Build and run the Blazor WebAssembly frontend

```
dotnet run --launch-profile https
```

6. Find the **https** link in the terminal and click it to open the application



```
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: https://localhost:7062
```

Certificates for Secure SSL Connections

This system relies on HTTPS encryption to securely transport data between the client and server. To enable secure communication, the system uses two types of certificates: a self-signed keystore for the server and a public certificate for the client.

Server-Side Certificate

The server uses a self-signed `` `keystore.jks` `` file, which contains the `` `mycert` `` certificate. This file is located in the `` `src/main/resources` `` folder of both the **LoadBalancer** and
WebAPI modules.

- The `` `keystore.jks` `` file includes both the private and public keys.
- The certificate details (file path, password, and alias) are configured in the `` `application.properties` `` file.

Client-Side Certificate

The client must trust the server's certificate by using the same public key. To achieve this, the public key is extracted from the `` `keystore.jks` `` file using the **Keytool** utility. The exported public key is stored in the `` `backend-cert.cer` `` file, located in the `` `src/main/resources` `` folder of the **LoadBalancer** module.