# Flex ESD Inpasseringssystem – Installation och användarmanual

TDP017 – Grupp D  
Linköpings universitet  
2019-11-29

Innehåll

[Flex ESD Inpasseringssystem – Installation och användarmanual 1](#_Toc26797480)

[Innehåll 1](#_Toc26797482)

[Backend 1](#_Toc26797483)

[Installation – Windows 1](#_Toc26797484)

[Installation – Linux (Med Docker) 4](#_Toc26797485)

[Installation – Linux (Utan Docker) 6](#_Toc26797486)

[Android-app 8](#_Toc26797487)

[Administrativ frontend 10](#_Toc26797488)

[Användarmanual för frontend 10](#_Toc26797489)

[Administrera 10](#_Toc26797490)

[Tabeller 11](#_Toc26797491)

[Grafer 11](#_Toc26797492)

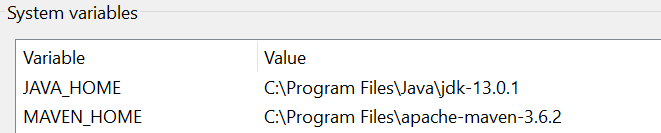
# Backend

## Installation – Windows

Installera senaste versionen av Docker Toolbox för att få tillgång till Docker Quickstart. Detta är en virtualiseringsmiljö för att enkelt installera och köra PostgreSQL, vilket är databasen som backend använder.

Installera en modern JDK-version och lägg till den i systemets miljövariabler. Backend är testad med JDK 13.0.1. Kontrollera att man i en kommandotolk (cmd) kan köra *java -version* och *javac -version*. Om de inte fungerar kan en omstart krävas.

Installera Apache Maven och lägg till det i systemets miljövariabler. Backend är testad med Maven 3.6.2. Kontrollera att man i en kommantotolk (cmd) kan köra *mvn -version*. Maven används för att kompilera och paketera backend.

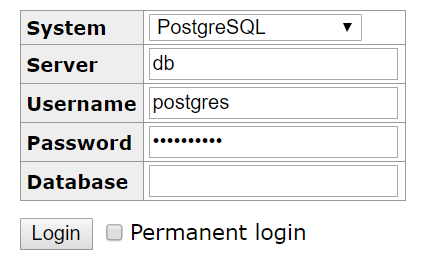


Kör Docker Quickstart och vänta på att den konfigurerats färdigt. Notera IP-addressen som visas. Default-addressen för docker bör vara 192.168.99.100.



Öppna en kommandotolk och navigera in i projektets /api-mapp, där ligger en *docker-compose*-fil.  
Kör kommandot *docker-compose up*. Detta startar backendens databas.

Första gången databasen sätts upp måste den initieras med en uppsättning tabeller och användare. Navigera in via en webbläsare till 192.168.99.100:8080, där 192.168.99.100 är er Docker-IP från ovan. Ställ in *System* till *PostgreSQL*, *Server* till *db*, *Username* till *postgres*, och *Password* till *verysecure*.

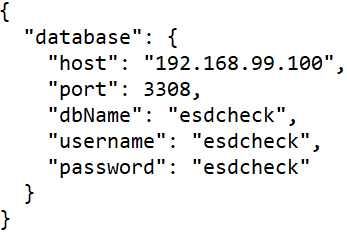


Till vänster på sidan, klicka på *SQL command*. Klistra därefter in innehållet från filen */api/src/main/resources/db/Create\_Role\_And\_Database.sql* och exekvera. Detta skapar en ny databas *esdcheck* och även ett nytt administrativt inlogg för den databasen.  
Det nya inlogget är användarnamn *esdcheck* och lösenord *esdcheck*.

Nu när databasen är konfigurerad kan vi kompilera och starta backend.

Öppna en kommandotolk och navigera in i mappen /api, där det ligger en pom.xml-fil. Kör därefter kommandot *mvn clean package*. Detta kompilerar backenden in i /api/target-mappen. Navigera därefter in i /api/target, och kör kommandot *java -jar esdcheck-api-0.0.1.jar*.

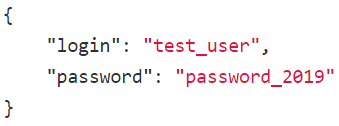
Om backenden inte lyckas att starta (något Java-fel kastas), så är det sannolikt något problem med anslutningen till databasen. Kontrollera att */api/src/main/resources/application.json* innehåller rätt information för inlogg mot databasen, kompilera om, och försök igen. Standardporten för databasen är 3308, och den ligger bakom Dockers IP.



Backenden öppnas som default på port 8099. För att kontrollera att den ligger uppe korrekt, öppna en webbläsare och navigera in på localhost:8099. Du bör se ett 404 felmeddelande från backend, men oroa dig inte, det är så den ska bete sig.



För att frontend ska fungera som den ska så behöver den ett inlogg för att autentisera sig mot backend. Denna användare skapas genom en PUT-request mot backend-URLen */auth/user* med en JSON-body som innehåller fälten *login* och *password*. Se en exempel-body nedan. Denna request kan utföras på valfritt sätt, vi rekommenderar programmet Postman.



Notera att backend skriver ut en speciell token vid start som används vid autentisering för *appen*.



Nu ska backend vara igång och de andra två delarna av projektet bör då bete sig som de ska.

## Installation – Linux (Med Docker)

Installera Docker, version 18.09.7 eller senare.

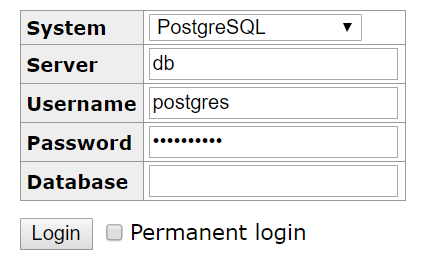
Installera docker-compose, version 1.25.0-rc4 eller senare.

Installera en modern version av JDK, version 12 eller senare.

Installera Maven, version 3.3.9 eller senare.

I terminalen, navigera in i projektets /api-mapp, där ligger en *docker-compose*-fil.  
Kör kommandot *docker-compose up (*alternativt *sudo docker-compose up)*. Detta startar backendens databas.

Första gången databasen sätts upp måste den inititieras med en uppsättning tabeller och användare. Navigera in via en webbläsare till localhost:8080. Ställ in *System* till *PostgreSQL*, *Server* till *db*, *Username* till *postgres*, och *Password* till *verysecure*.



Till vänster på sidan, klicka på *SQL command*. Klistra därefter in innehållet från filen */api/src/main/resources/db/Create\_Role\_And\_Database.sql* och exekvera. Detta skapar en ny databas *esdcheck* och även ett nytt administrativt inlogg för den databasen.  
Det nya inlogget är användarnamn *esdcheck* och lösenord *esdcheck*.

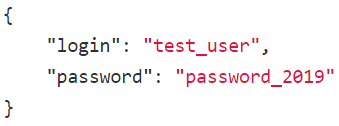
Nu när databasen är konfigurerad kan vi kompilera och starta backend.

Öppna en terminal och navigera in i mappen /api, där det ligger en pom.xml-fil. Kör därefter kommandot *mvn clean package*. Detta kompilerar backenden in i /api/target-mappen. Navigera därefter in i /api/target, och kör kommandot *java -jar esdcheck-api-0.0.1.jar*.

Om backenden inte lyckas att starta (något Java-fel kastas), så är det sannolikt något problem med anslutningen till databasen. Kontrollera att */api/src/main/resources/application.json* innehåller rätt information för inlogg mot databasen, kompilera om, och försök igen. Standardporten för databasen är 3308, och standard host är *localhost*.

Backenden öppnas som default på port 8099. För att kontrollera att den ligger uppe korrekt, öppna en webbläsare och navigera in på localhost:8099. Du bör se ett 404 felmeddelande från backend, men oroa dig inte, det är så den ska bete sig.



För att frontend ska fungera som den ska så behöver den ett inlogg för att autentisera sig mot backend. Denna användare skapas genom en PUT-request mot backend-URLen */auth/user* med en JSON-body som innehåller fälten *login* och *password*. Se en exempel-body nedan. Denna request kan utföras på valfritt sätt, vi rekommenderar programmet Postman.

Notera att backend skriver ut en speciell token vid start som används vid autentisering för *appen*.



Nu ska backend vara igång och de andra två delarna av projektet bör då bete sig som de ska.

## Installation – Linux (Utan Docker)

Börja med att uppdatera apt paketeringssystem.

$ sudo apt-get update

Installera PostgreSQL. Med -contribute-paketet får vi även några viktiga verktyg.

$ sudo apt install postgresql postgresql-contrib

Installera en modern version av JDK, version 12 eller senare.

Installera Maven, version 3.3.9 eller senare.

PostgreSQL har olika roller som kan bindas till respektive databas. Under installationen skapas en role (roll), ”postgres” som funkar som en admin. En Postgres role är associerad med ett Unix/Linux konto. Det vill säga man måste byta till det kontot för att logga in i databasen. Systemet måste ha ett Linux konto med namnet ”esdcheck”, och lösenord ”esdcheck”. För att skapa ett konto skriv in detta i terminalen.

$ useradd esdcheck

$ cat /etc/passwd

Logga in på detta konto via detta kommando.

$ sudo -i -u esdcheck

Prova att logga in på PostgreSQL

$ psql

Från postgres terminalen skapa en database ”esdcheck”

$ esdcheck@ CREATE DATABASE esdcheck;

Gå ur terminalen med \q. API kommer sedan att skapa de tabeller som behövs. Om det inte funkar kan man köra ett script från terminalen med detta kommando (filen måste vara i samma folder). Filen kommer från */api/src/main/resources/db*.

$ psql -U username -d database -f Create\_Role\_And\_Database.sql

Nu när databasen är konfigurerad kan vi kompilera och starta backend.

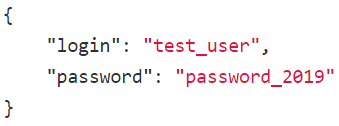
Öppna en terminal och navigera in i mappen /api, där det ligger en pom.xml-fil. Kör därefter kommandot *mvn clean package*. Detta kompilerar backenden in i /api/target-mappen. Navigera därefter in i /api/target, och kör kommandot *java -jar esdcheck-api-0.0.1.jar*.

Om backenden inte lyckas att starta (något Java-fel kastas), så är det sannolikt något problem med anslutningen till databasen. Kontrollera att */api/src/main/resources/application.json* innehåller rätt information för inlogg mot databasen, kompilera om, och försök igen. Standardporten för databasen är 3308, och standard host är *localhost*.

Backenden öppnas som default på port 8099. För att kontrollera att den ligger uppe korrekt, öppna en webbläsare och navigera in på localhost:8099. Du bör se ett 404 felmeddelande från backend, men oroa dig inte, det är så den ska bete sig.



För att frontend ska fungera som den ska så behöver den ett inlogg för att autentisera sig mot backend. Denna användare skapas genom en PUT-request mot backend-URLen */auth/user* med en JSON-body som innehåller fälten *login* och *password*. Se en exempel-body nedan. Denna request kan utföras på valfritt sätt, vi rekommenderar programmet Postman.



Notera att backend skriver ut en speciell token vid start som används vid autentisering för *appen*.



Nu ska backend vara igång och de andra två delarna av projektet bör då bete sig som de ska.

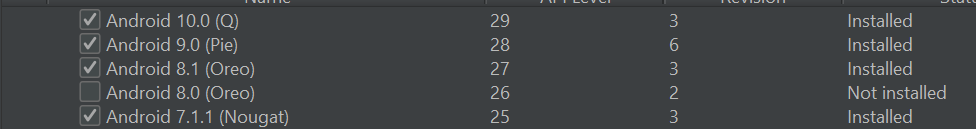
# Android-app

Android-appen är delen av projektet som ansluter sig till smartcard-skannern samt backend, och hanterar användarnas inpasseringar.

För att bygga vidare på eller kompilera appen till en .apk-fil så krävs Android Studio. Ladda ner och installera en modern version av Android Studio.

Öppna Android Studio. Klicka på öppna projekt och välj mappen *AndroidApp/Sample Code*.

För att kompilera projekt ordentligt krävs några Android SDKs att bygga mot. Navigera in till *File->Settings->Appearance & Behavior->Android SDK*. Kryssa i rutorna för Android 10.0 (Q), Android 9.0 (Pie), Android 8.1 (Oreo), Android 7.1.1 (Nougat). Det är möjligt att alla dessa versioner är överflödiga, men det är en konfiguration som fungerar för oss. Experimentera på egen risk.



Kör *File->Sync Project with Gradle Files*. Projektet bör bli synkat och byggt utan några felmeddelanden.

För att bygga appen till en ansluten surfplatta eller mobil, så kopplar ni in enheten via USB till datorn som kör Android Studio. Enheten som ansluts till datorn måste ha Developer Mode och USB Debugging aktiverat. Googla hur man aktiverar dessa funktioner för din specifika enhet och operativsystem. Om den utgivna surfplattan används, så bör ingen extra konfiguration behövas.

Om Android studio känner igen din enhet på rätt sätt, så kommer du se något likt detta högst upp i Android Studio:



Om Android Studio inte känner igen din enhet, även fast du har Developer Mode & USB Debugging aktiverat, så kan du behöva speciella USB-drivrutiner på din dator för att kunna kommunicera med enheten. (Exempelvis Samsung USB Driver om en Samsung-enhet används.).

När Android Studio känner igen din enhet, så bör du kunna bygga och installera Appen till enheten. Klicka på  längst upp i Android Studio så bör appen byggas på korrekt sätt.

Alternativt så kan en färdigbyggd APK-fil delas mellan enheter, för att slippa strulet med Android Studio. APK-filen efter kompilering hittas i */AndroidApp/Sample Code/app/build/outputs/apk/debug*.

Appen kräver två permissions vid uppstart, Bluetooth och Access Coarse Location. Aktivera Bluetooth och tillåt de båda.

När appen är startad så är det en mängd grejor som behöver konfigureras. Notera att backend bör vara igång. Öppna appens settings-meny (Klicka längst upp till höger, och sedan *Settings*.). Här finns tre fält som behöver fyllas i för att kunna ansluta till backend på korrekt sätt.

Till att börja med, fyll i IP-addressen och porten som backend körs på. Detta är sannolikt den externa IP:n för servern som backend körs på, samt standardporten 8099. Fyll även i den *token* som skrevs ut när backend startades, det ser ut någon likt detta:



Klicka på knappen ”Testa backend och auth” för att testa anslutningen mot backenden samt autentisering mot den. Om det lyckas skrivs ”Connected and authorized!” ut.

När vi kan ansluta och autentisera mot backend, så är det dags att koppla på NFC-kortläsaren. Appen är designad mot läsaren ACR1255U-J1.

När appens huvudfönster visas, aktivera kortläsarens Bluetooth-läge enligt bilden nedan.



Läsaren och Appen bör då paras. När de är parade så kan appen börja användas ordentligt.

När ett kort skannas som systemet inte har sett förut, så får man alternativet att registrera sig som en ny anställd (employee). Mata då in ditt namn och klicka på registrera.

När ett kort skannas som systemet har sett förut, så får man alternativet att skapa en ny ESD-kontroll, eller att se alla sina gamla ESD-kontroller. När man skapar en ny ESD-kontroll får man alternativet att klicka antingen *Pass* eller *Fail*, beroende på resultatet på er ESD-våg.

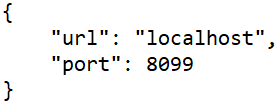
När man kollar på sina gamla ESD-kontroller så ser man en kalender, som kan ha röda eller gröna pluppar på varje datum. Röda pluppar innebär ett test med resultatet *Fail*, och gröna innebär ett test med resultatet *Pass*. Om man ett test med *Fail*, och ett test med *Pass* på samma dag, så visas grön plupp.

# Administrativ frontend

Stegen för installation och konfiguration för frontend bör vara nästan identiska oavsett system.

Dessa steg antar att du har backend installerad och konfigurerad. För att kunna logga in på frontend så behöver backend ha ett login registrerat.

Om backend ligger igång på samma maskin som frontends webbserver så bör standardkonfigurationen fungera. Om backend ligger på någon annan maskin, så behöver du ändra IP och port i */frontend/admin-webb-app/src/config.json*. Standardporten för backend är 8099.



Installera senaste versionen av Node. Frontend är testat med version 12.13.1.

Öppna en terminal/kommandotolk och navigera in i */frontend/admin-webb-app/*. Kör sedan kommandot *npm install* för att installera alla paket som frontend kräver.

För att bygga och starta frontend så körs sedan kommandot *npm start*. Som standard så hamnar frontend på port 3000.

Navigera in på localhost:3000 i en webbläsare och logga in med användaren som registrerades på backend för att få tillgång till webbsidan.

## Användarmanual för frontend

Det har skapats en frontend som främsta uppgift är att hantera data till/från databasen och dess användare. Hemsidan finns endast på engelska och består av tre huvudsektioner:

* En administrativ sida för hantering av användare och data i systemet.
* En sida för insamling av data i form av en tabell. Tabellen kan även exporteras till en JSON eller Excel-fil.
* En sida för insamling av data i form av en graf. Detta skall kunna användas för en snabb översyn över data. Datan kan visas för varje månad.

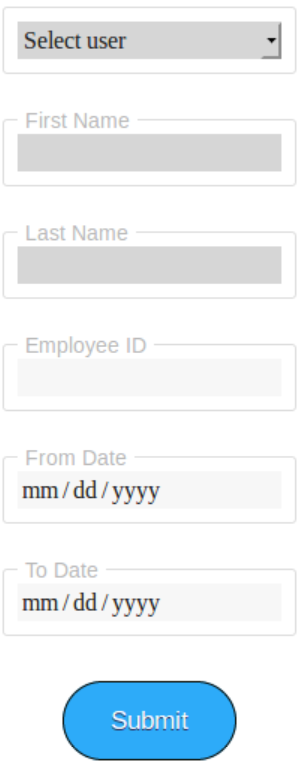
### Administrera

En ny användare för systemet måste först registrera sig. Registreringen av en användare sker vid Android Applikationen. Detta eftersom varje användare är bundet till det kort som läses vid Android applikationen. Denna registrering ligger i databasen som ”employee”. Om användaren också vill ha tillgång till sin data via hemsidan, måste detta registreras av en admin på hemsidan. I databasen ligger detta under ”user”.

Huvudändamålet med den administrativa sidan är att kunna göra ändringar i databasen. Det är till exempel möjligt att ta bort en användare från databasen. Om en användare byter kort och/eller skall inte kontrolleras för ESD måste denne avregistreras från systemet. Det är även möjligt att ta bort ESD kontroller från en användare eftersom det kan ske praktiska felaktiga registreringar som har en klar orsak. Den sista funktionen på hemsidan är att ange en ny användare för frontend/hemsidan.

### Tabeller

För att kunna mer specifikt inspektera ESD check data från en användare kan man få datan i form av en tabell. Då är det möjligt se direkt vid varje registrerat tillfälle om en användare har fått en godkänd ESD kontroll. För att få fram data måste man ange användarnummer och två datum under ”Select user”, ”From date”, ”To date”. Data hämtas efter man tryckt på knappen submit. Denna tabell kan sedan exporteras i form av JSON- eller Excel-fil.



### Grafer

Det är möjligt att få fram ESD check data från en användare genom att ange användarnummer och det tidsintervall som man vill undersöka. Detta görs under ”Select user”, ”From date”, och ”To date”. Datan kommer att hämtas från databasen och grupperas månadsvis. Uppvisning sker med två staplar, grön för godkända och röd för icke godkända. Om man misstänker att det finns felaktiga värden kan man ta bort dessa under den administrativa sidan. Det är inte möjligt att exportera grafer utan endast tabeller.

