**palmquist**

This webapp is a rewrite of the Beinecke Library legacy application Palmquist.

The project can be built using Maven, and the resulting .war file can be dropped into Tomcat (or just launched with java -jar). The stack is Spring Boot/MVC/Data.

The HTML files are in src/main/resources/templates. All the backend logic is in package edu.yale.

**Installation**

* Install [Maven](https://maven.apache.org/) (and [JDK](http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html)).
* Download sqljdbc4 from [Microsoft] (<https://www.microsoft.com/en-us/download/details.aspx?id=11774>).
* If necessary, adjust the path to the database connection properties file in src/main/java/edu/yale/DatabaseInitializer class. By default, it points to AWS. Specify either db.prop for "brothers" or aws.prop for AWS RDS database called palmquist in class DatabaseInitializer.java.
* Build and Launch

# only the 1st time

mvn install:install-file -Dfile=sqljdbc4.jar -DgroupId=com.microsoft.sqlserver -DartifactId=sqljdbc4 -Dversion=4.0 -Dpackaging=jar

# go into the folder:

cd palmquist

# run the build and package it

mvn clean install

# to test it:

java -jar target/palmquist-0.0.1-SNAPSHOT.war

**Deployment to AWS**

Create aws.prop file. Just fill in the username and password:

connection = jdbc:sqlserver://dbpalmquist.ckao2ciychcj.us-east-1.rds.amazonaws.com:1433;databaseName=palmquist

user =

password =

'scp' the .war in target/ to /usr/share/tomcat/webapps, and the aws.prop file (which has AWS credentials) as follows. Don't forget to change the IP of AWS server in the following commands:

scp -i yul-palmquist-prod-app.pem palmquist/target/palmquist-0.0.1-SNAPSHOT.war ec2-user@10.5.68.174:/home/ec2-user

scp -i yul-palmquist-prod-app.pem aws.prop ec2-user@10.5.68.174:/opt/palmquist

ssh -i yul-palmquist-prod-app.pem ec2-user@10.5.68.174

rm -rf /usr/share/tomcat/webapps/ROOT

rm -rf /usr/share/tomcat/webapps/ROOT.war

sudo cp ~/ROOT.war /usr/share/tomcat/webapps/

The folder where you're running this command should have the .pem file for AWS authentication. Check paths to make sure you can reach the .war file.

We're copying the aws.prop file to /opt/palqmuist/, so that it can be pick up database properties on load. The file is named ROOT.war so that Tomcat can deploy it under the default context.

Once you've copied (using scp, as shown above) these files, Tomcat will automatically expand the .war file to full directory. The app should now be running at 8080.

The log file is in /tmp/palmquist-logs/debug.log. Use 'cat' or 'tail' to look at the log.

**Try it**

* Hit (for local testing): <http://localhost:8080/>
* For local Tomcat: <http://localhost:8080/palmquist-1.0.0-SNAPSHOT>
* For AWS Tomcat Instance: [http://AWS:8080](http://aws:8080/)