



INSTALLATION DOCUMENT

Project of Software Engineering 2

WEATHER-CAL

Authors:

PAOLO POLIDORI

MARCO EDEMANTI

Contents

| | | |
|----------|--|----------|
| 1 | Introduction | 5 |
| 1.1 | System Overview | 5 |
| 1.2 | Purpose | 5 |
| 2 | Installation | 7 |
| 2.1 | Software Required | 7 |
| 2.1.1 | Install JDK 8 | 7 |
| 2.1.2 | Install Glassfish Server | 8 |
| 2.1.3 | Install PostgreSQL | 8 |
| 3 | Configuration | 9 |
| 3.1 | PostgreSQL configuration | 9 |
| 3.2 | Glassfish Server configuration | 10 |

Chapter 1

Introduction

1.1 System Overview

WeatherCal it's a Web application that allows the registered users to manage their schedule basing on the weather conditions. The users are also able to invite people to their event and to publicize their event. The system is even capable of notifying the users, as they log on to the system, if the weather forecast for the event is not the desired one. The target of this software is, therefore, to give the users a tool for schedule their events smartly, giving the possibility to change preferences as the weather forecast changes.

1.2 Purpose

This document describes how to correctly set up the system in order to be able to run this application. This document will guide you in order to perform the installation and the configuration of the needed components used by WeatherCal.

Chapter 2

Installation

2.1 Software Required

WeatherCal it's a Web application developed for the application server GlassFish Server and it use PostgreSQL Database to save the data. Therefore in order to execute it you need to download and install the following software:

- *Glassfish Open Source Edition 4.1*, download [here](#)
- *PostgreSQL >= 9.1*, download [here](#)
- *Java SE Development Kit 8*, download [here](#)

2.1.1 Install JDK 8

It is recommended to install JDK 8 from [here](#), but if you want to install JDK 7 you can do it but remember to switch the `maven.compiler.source` and `maven.compiler.target` version in your `pom.xml` from 1.8 to 1.7. JDK is needed in order to be able to use Glassfish Server. Please be careful and install the correct version of the JDK de-

pending on your system (32 bit or 64 bit) and your Operating System (Windows, Linux or OS X).

2.1.2 Install Glassfish Server

Once you get JDK 8 you can proceed with the installation of Glassfish Server. First of all download it from [here](#) and then unpacking it in the folder you desire, keep in mind that for the correct execution of our application it is important to execute it in the same context it is developed.

2.1.3 Install PostgreSQL

The next step is to complete the installation procedure is download PostgreSQL from [here](#), be care to get a version greater then 9.1.

Chapter 3

Configuration

3.1 PostgreSQL configuration

Set up PostgreSQL for accepting Unix-socket and TCP/IP connections from localhost without password for the user *postgres*, creating it if does not exists with the this query:

```
CREATE ROLE postgres  
WITH SUPERUSER LOGIN UNENCRYPTED  
PASSWORD 'postgres';
```

Now we have to establish a connection to the DBMS (psql, pgAdmin, DBC coding, ...) using this query:

```
DROP DATABASE IF EXISTS weathercal;  
CREATE DATABASE weathercal  
WITH OWNER = postgres
```

```
ENCODING = UTF8  
TABLESPACE = pg default  
LC_COLLATE = it_IT.UTF-8  
LC_CTYPE = it_IT.UTF-8  
CONNECTION LIMIT = 100;
```

The last step now is to connect weathercal database and create the schema for the application issue the next query:

```
CREATE SCHEMA development  
AUTHORIZATION postgres;
```

3.2 Glassfish Server configuration

In order to make the authorization and authentication system provided by Glassfish work, the configuration of a Realm is needed. To perform this operation you must start the server and open the Glassfish Admin Console accesible at

http://localhost:4848. From there, edit the service-config Configuration and, in particular, add a new Realm in the Security section. Set an arbitrary name for the realm and associate it to the JDBCRealm Class name. Now set the class-specific properties as follows:

| Parameter | Value |
|---------------------|----------------------------|
| JAAS Context | jdbcRealm |
| User Table | development.user_ |
| User Name Column | email |
| Password Column | password |
| Group Table | development.user_has_group |
| Group Name Column | id_group |
| Password Encryption | SHA-256 |
| Encoding | Hex |
| Charset | UTF-8 |

Optionally you can also use the command-line tool , issuing the next command:

```

1  $ asadmin create-auth-realm --classname com.sun.enterprise.security.
    auth.realm.jdbc.JDBCRealm --property 'datasource-jndi=java\:app/
    jdbc/pgResource:digestrealm-password-enc-algorithm=SHA-256:group-
    name-column=id_group:group-table=development.user_has_group:jaas-
    context=jdbcRealm:password-column=password:user-name-column=email:
    user-table=development.user_:charset=UTF-8:encoding=Hex' mpRealm
3

```

Or, you can edit the <glassfish_home>/domains/<domain_name>/config/domain.xml file and add the next node as a child of the security-service node:

```

2  <auth-realm classname="com.sun.enterprise.security.auth.realm.
    jdbc.JDBCRealm" name="mpRealm">
    <property name="jaas-context" description="null" value="jdbcRealm"></
    property>
    <property name="encoding" value="Hex"></property>
4  <property name="charset" value="UTF-8"></property>

```

```

6  <property name="password-column" description="null" value="password"
    ></property>
    <property name="datasource-jndi" value="java:app/jdbc/pgResource"></
        property>
    <property name="group-table" value="development.user_has_group"></
        property>
8  <property name="user-table" value="development.user_"></property>
    <property name="group-name-column" description="null" value="id_group
        "></property>
10 <property name="password-encryption-algorithm" value="SHA-256"></
    property>
    <property name="user-name-column" description="null" value="email"></
        property>
12 </auth-realm>
```

Time reporting

| | Paolo Polidori | Marco Edemanti |
|------------|-----------------------|-----------------------|
| DD writing | 35 hours | 28 hours |

List of Figures