Installation and Configuration Guide

Table of Contents

1.Introduction

1.1. Overview

This guide provides detailed instructions on how to set up the environment for Timetable scheduling System. It lists all product requirements and guides you through the installation process. This guide can be used by all users, system testers and Evaluators.

1.2. How This Guide is Organized

Chapter 1 Introduction: the current chapter. It explains the purpose of this document, defines its audience and explains its structure.

Chapter 2 Installation Prerequisites: lists all Tools requirements, as well as how to set up a Spring Boot Application With PostgreSQL

Chapter 3 Installing the SpringSource Tool Suite

2. Installation Prerequisites

2.1 Tools and Technologies Used

- Spring Boot - 2.0.4.RELEASE
- JDK - 1.8 or later
- Spring Framework - 5.0.8 RELEASE
- IDE - Spring Tool Suite (STS)
- PostgreSQL - 42.2.5

2.2 Installing the SpringSource Tool Suite

The SpringSource Tool Suite (STS) is a development environment based on Eclipse that comes configured with all the plugins needed to work with dm Server and OSGi. This includes the latest version of dm Server Tools, so no updates are necessary. Although the steps are similar, the details of installing STS depend on the operating system.

Go to the STS download site (http://www.springsource.com/products/sts) and download the variant appropriate to the operating system being used. This guide is consistent with STS version 2.3.0.RELEASE. Previous versions may not work properly with the latest revision of GreenPages, currently 2.1.0.RELEASE.

Installing STS on Windows operating systems

Unzip the download of STS to the root directory of a drive (this will avoid possible problems with long pathnames).

```
prompt> cd C:\
prompt> "%JAVA_HOME%"\bin\jar xf \full…path…to\springsource-tool-suite-2.3.0.RELEASE-e3.4-win32.zip
```

To verify the installation, run the eclipse.exe executable in the unzipped directory and check that STS displays a welcome panel. The first time there may be a short delay due to the initial set-up of indexes.

Installing STS on UNIX operating systems

Unpack the download of STS to a suitable location on the file system, such as /opt or, if root access is not available, the home directory. (If the download was automatically unpacked by the operating system, simply move the unpacked directory to the chosen location.)

To verify the installation, run the STS executable (Eclipse.app on Mac OS X) in the unpacked directory and check that STS displays a welcome panel. The first time there may be a short delay due to the initial set-up of indexes.

Note about Java versions in STS

SpringSource Tool Suite runs on Eclipse using Java Version 1.5, and dm Server requires Java Version 1.6. The GreenPages application built here also requires Java Version 1.6. Alter the default Java compiler settings in STS before proceeding:

1. In SpringSource Tool Suite, click **Window > Preferences** from the menu.
2. In the **Preferences** window, click **Java > Compiler** in the left panel.
3. In the right panel, set the **Compiler compliance level** to 1.6.
4. Click **Apply**. You will get a message asking if you want a full rebuild; click **Yes**. The rebuild should take very little time to complete.
   You might also see a message similar to the following on the settings panel: *"When selecting 1.6 compliance, make sure to have a compatible JRE installed and activated (currently 1.5)."* A link to *Configure* this will appear. Select this link to open the *Java--*

Installed JREs panel. If not already selected, choose a JRE suitable for Java Version 1.6.x (for example JVM 1.6.0).

5. Click **OK**.

2.3 Set up a Spring Boot Application With PostgreSQL

In this part, we will see the steps to set up a Spring Boot application with PostgreSQL. We will have a simple CRUD operation in Postgres Database by exposing the application via Rest API. We will use POSTMAN to test the application.

2.3.1 Setting up Postgres Server

- Download the Postgres server from the link: https://www.postgresql.org/download/
- Run the installer. It will also ask the password for the superuser: *postgres*
- Click on pgAdmin4.exe located inside the PostgreSQL folder inside Program Files.

2.3.1.1 Requirements Overview

PostgreSQL 9.6 is certified on the following platforms:

32 bit Windows:

Windows 7, 8, and 10 Windows 2008 Server

64 bit Windows:

Windows 7, 8, and 10 Windows 2012 Windows 2008

32 bit Linux:

CentOS 6.x Oracle Enterprise Linux 6 Ubuntu 14.04

64 bit Linux:

CentOS 6.x and 7.x Debian 7 and 8 Oracle Enterprise Linux 6 and 7 SLES 12 Ubuntu 14.04

MAC OS X:

OS X Server 10.8, 10.9, and 10.10

2.3.1.2 Hardware Requirements

The following installation requirements assume you have selected the default options during the installation process. The minimum hardware required to install and run PostgreSQL are:

- A 1 GHz processor
- 1 GB of RAM
- 512 MB of HDD

2.3.2 Setting up Spring Boot Application

Prerequisite:

Have JDK 1.8 installed

- Download a sample Spring Boot project from https://start.spring.io/
- Update the pom.xml as below:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>
<parent>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>2.1.1.RELEASE</version>
<relativePath /> <!-- lookup parent from repository -->
</parent>
<groupId>com.sample</groupId>
<artifactId>postgress</artifactId>
<version>0.0.1-SNAPSHOT</version>
<name>postgress</name>
<description>Demo project for Spring Boot</description>
<properties>
<java.version>1.8</java.version>
</properties>
<dependencies>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter</artifactId>
</dependency>
<dependency>
```

```
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-web</artifactId>
</dependency>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-jdbc</artifactId>
</dependency>
<dependency>
<groupId>org.postgresql</groupId>
<artifactId>postgresql</artifactId>
<scope>runtime</scope>
</dependency>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-test</artifactId>
<scope>test</scope>
</dependency>
</dependencies>
<build>
<plugins>
<plugin>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-maven-plugin</artifactId>
</plugin>
</plugins>
</build>
</project>
```

spring-boot-starter-jdbc artifact will give all the spring jdbc related jars

org.postgresql.postgresql will have the dependency of postgres jdbc driver in runtime.

- Changes in application.properties to configure the data source with URL, username, and password of the Postgres DB. 5432 is the default port of Postgres. Hibernate will automatically pick up the postgressSQLDialect.

```
spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.PostgreSQLDialect
spring.jpa.hibernate.ddl-auto=none
spring.jpa.hibernate.show-sql=true
spring.datasource.url=jdbc:postgresql://localhost:5432/postgres
spring.datasource.username=postgres
spring.datasource.password=admin
spring.datasource.initialization-mode=always
spring.datasource.initialize=true
spring.datasource.schema=classpath:/schema.sql
spring.datasource.continue-on-error=true
```

spring.jpa.hibernate.ddl-auto will turn off the hibernate auto-creation of the tables from the entity objects. Generally, Hibernate runs it if there is an Entity defined. But we will be using a native SQL query with JdbcTemplate, hence, we can turn this off as we will not be creating an Entity.

spring.datasource.initialization-mode is marked as always as we want initialization of the database to happen on every startup. This is optional and made for this sample purpose.

spring.datasource.initialize=true will mark the initialization to be true.

spring.datasource.continue-on-error=true will continue application startup in spite of any errors in data initialization.

spring.datasource.schema is the schema path that needs to be initialized.

spring.datasource.url URL of the Postgres DB. It can be a remote DB as well.

spring.datasource.username username for the database.

spring.datasource.password password for the database.