

Lab RSMortgage Config Service

**Spring Cloud RSMortgage Config Service**

Udemy Exploring Spring Cloud Course

Presented By

Binit Datta

**Rolling Stone Technology**

**Formatted: December, 2016**

**Table of Content**

1.0 - Introduction 3

1.1 – Create a new Spring Starter Project 4

1.2 – Fill initial values 5

1.3 – Choose Nothing and click next 6

1.4 – Do not select any project references 7

1.5—Let it complete 8

1.6 – Open the pom.xml and verify 9

1.7 – Change parent project 9

1.8 – Add Dependency Management 9

1.8 – Add the following dependencies 10

1.9 – Add the following properties 10

1.10 – Add the following build section 10

1.11 – Add the application.yml property file under resources folder 11

1.12 – Add the bootstrap.yml property file under resources folder 11

1.13 – Add the following program under the package 12

1.14 – Conclusion 13

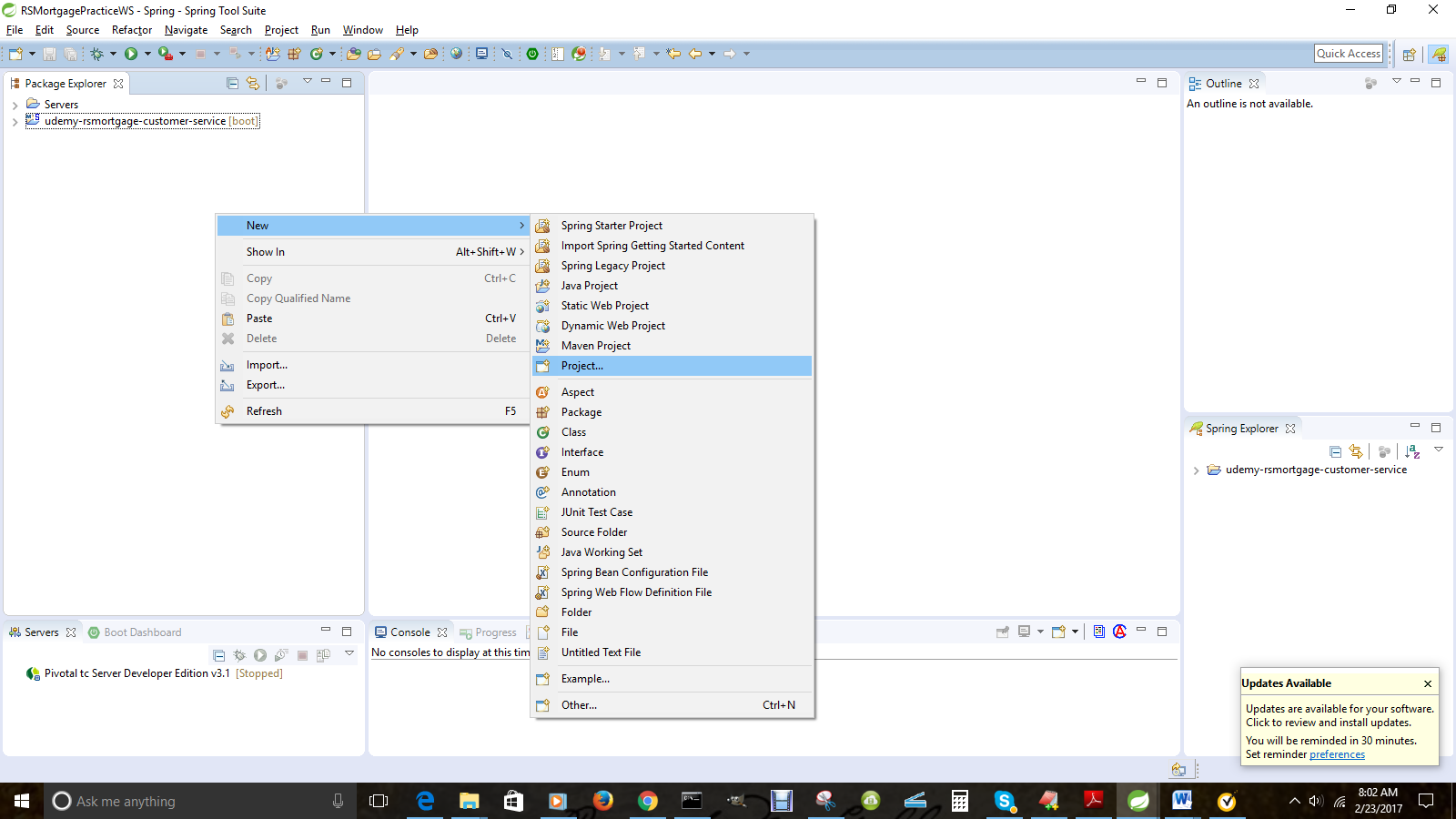
Chapter 1

*Spring Cloud RSMortgage Config Service Project Creation*

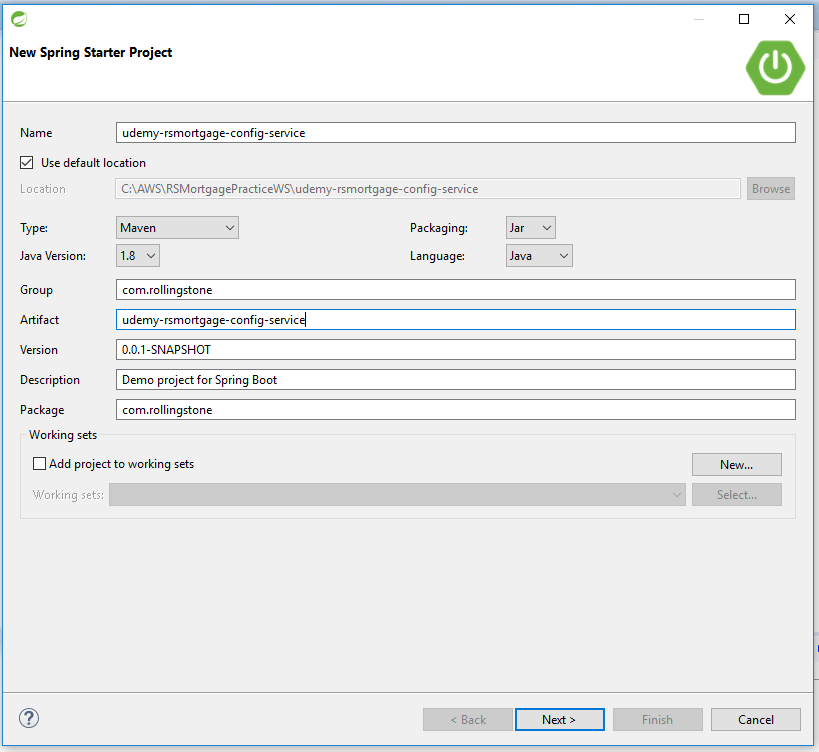
1.0 - Introduction

The following sections will lead us through creating the Spring Cloud Config Service creation, one step at a time.

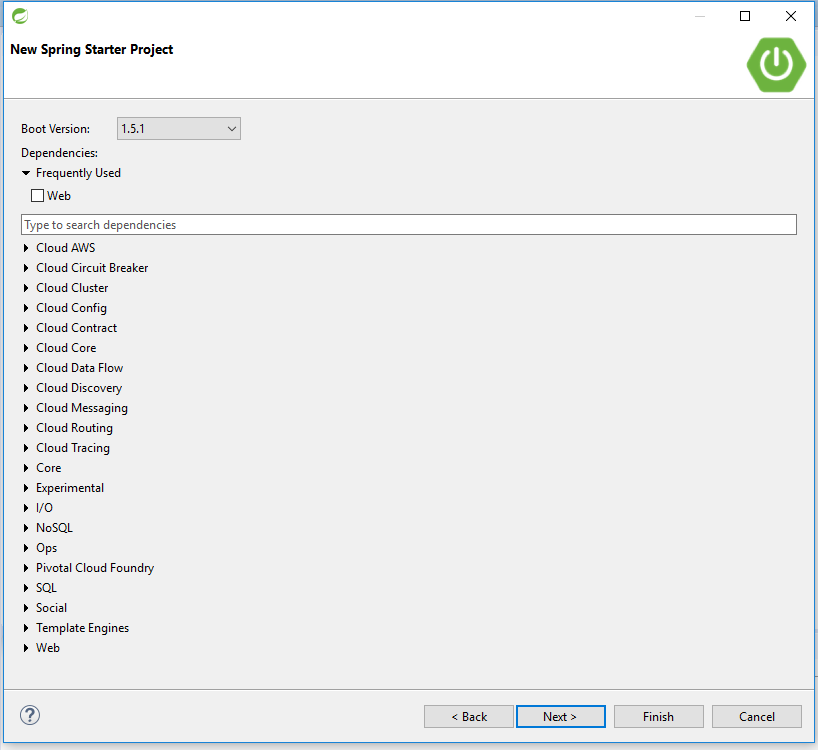
1.1 – Create a new Spring Starter Project



1.2 – Fill initial values

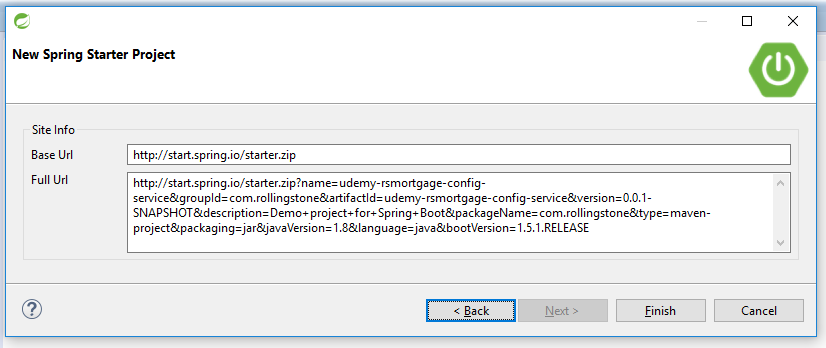


1.3 – Choose Nothing and click next

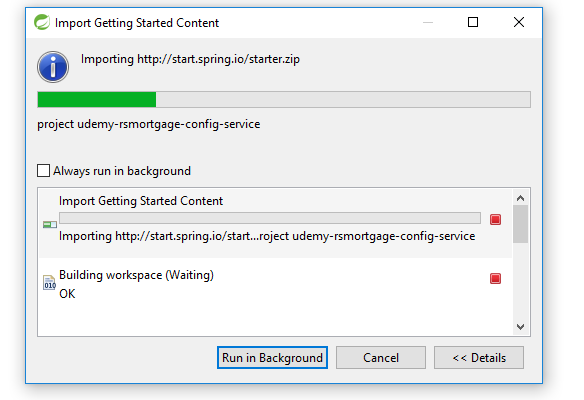


.

1.4 – Do not select any project references



1.5—Let it complete



1.6 – Open the pom.xml and verify

<groupId>com.rollingstone</groupId>

<artifactId>udemy-rsmortgage-customer-service</artifactId>

<version>1.0</version>

<packaging>jar</packaging>

<name>udemy-rsmortgage-config-service</name>

<description>Udemy RSMortgage Config Server</description>

1.7 – Change parent project

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>1.4.1.RELEASE</version>

</parent>

1.8 – Add Dependency Management

<dependencyManagement>

<dependencies>

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-dependencies</artifactId>

<version>Camden.SR1</version>

<type>pom</type>

<scope>import</scope>

</dependency>

</dependencies>

</dependencyManagement>

1.8 – Add the following dependencies

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-config-server</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-config</artifactId>

</dependency>

1.9 – Add the following properties

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<start-class>com.rollingstone.configserver.RSMortgageConfigServerApplication</start-class>

<java.version>1.8</java.version>

</properties>

1.10 – Add the following build section

<build>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.1</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

1.11 – Add the application.yml property file under resources folder

####################################

server:

port: 9000

spring:

cloud:

config:

server:

git:

uri: https://github.com/dattabinit/udemy-rsmortgage-config-repository

searchPaths: udemy-rsmortgage-customer-service

username: dattabinit

password: password123

1.12 – Add the bootstrap.yml property file under resources folder

spring:

application:

name: udemy-rsmortgage-configserver

1.13 – Add the following program under the package

package com.rollingstone.configserver;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.EnableAutoConfiguration;

import org.springframework.cloud.config.server.EnableConfigServer;

import org.springframework.context.annotation.Configuration;

@EnableAutoConfiguration

@EnableConfigServer

public class RSMortgageConfigServerApplication {

public static void main(String[] args) {

SpringApplication.run(RSMortgageConfigServerApplication.class, args);

}

}

1.14 – Conclusion

This document listed the steps as well as provided the explanation of creating a Spring Boot application based on Spring Cloud Service Discovery as well as JPA.