



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

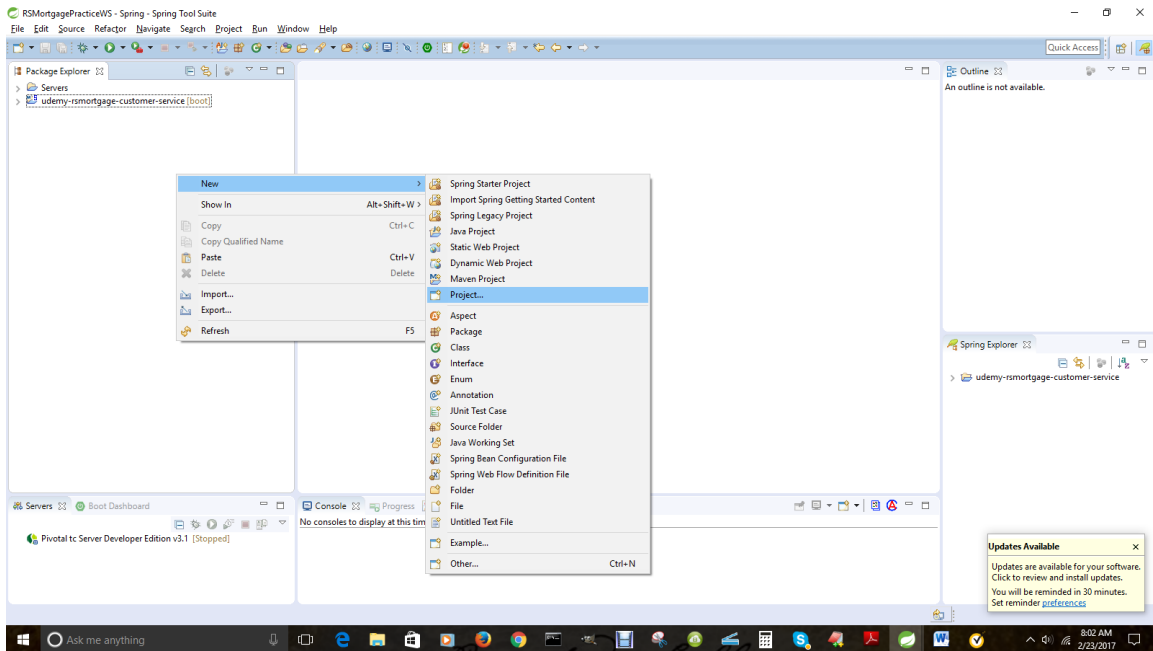
Chapter 1

Spring Cloud RSMortgage Config Service Project Creation

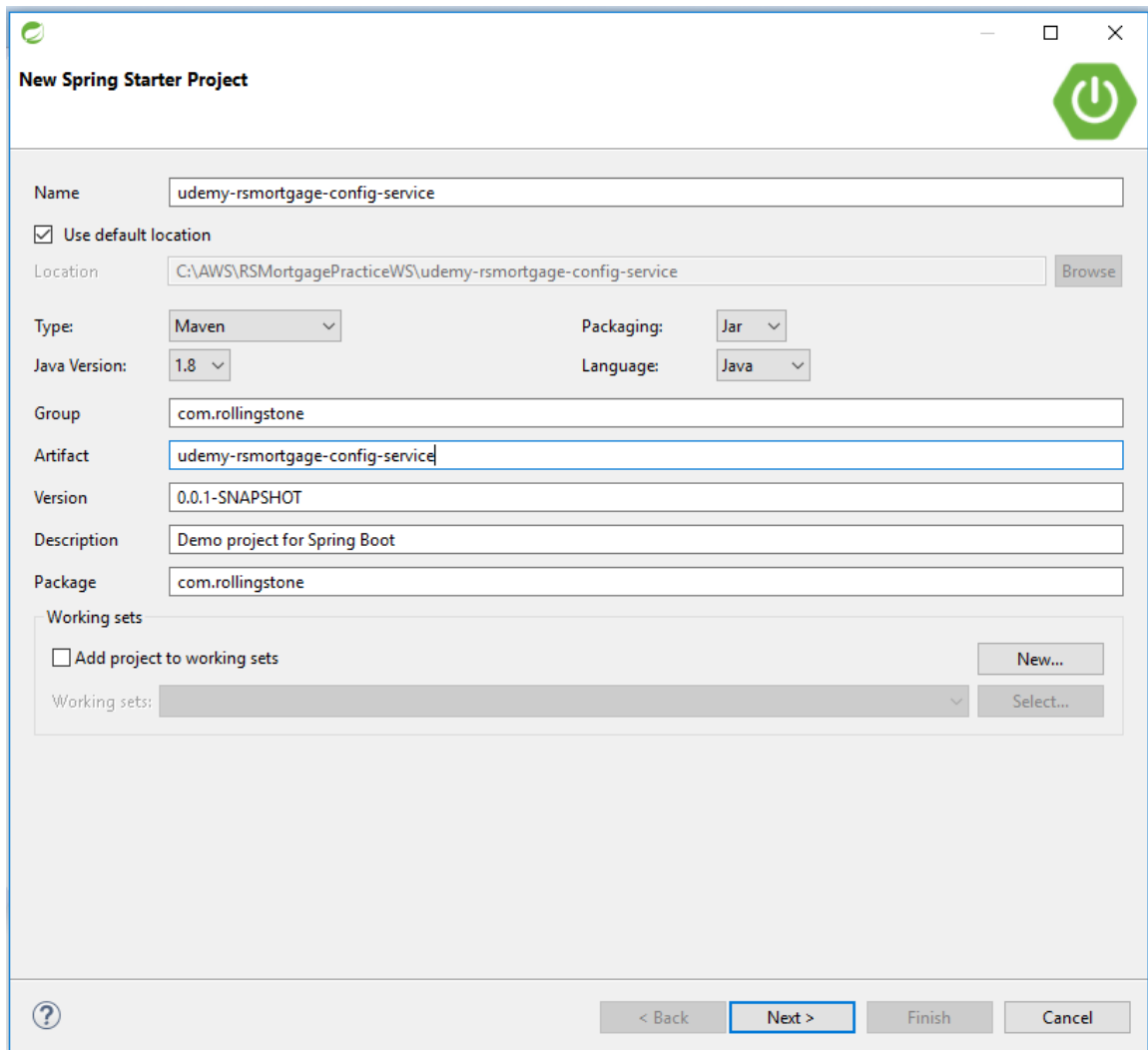
1.0 - Introduction

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



The image shows the 'New Spring Starter Project' dialog box in an IDE. The dialog is titled 'New Spring Starter Project' and features a green power button icon in the top right corner. The main area contains several input fields and checkboxes for configuring a new project. The 'Name' field is filled with 'udemy-rsmortgage-config-service'. The 'Use default location' checkbox is checked. The 'Location' field shows the path 'C:\AWS\RSMortgagePracticeWS\udemy-rsmortgage-config-service' with a 'Browse' button. The 'Type' is set to 'Maven', 'Packaging' to 'Jar', 'Java Version' to '1.8', and 'Language' to 'Java'. The 'Group' is 'com.rollingstone', 'Artifact' is 'udemy-rsmortgage-config-service', 'Version' is '0.0.1-SNAPSHOT', 'Description' is 'Demo project for Spring Boot', and 'Package' is 'com.rollingstone'. At the bottom, there is a 'Working sets' section with an unchecked 'Add project to working sets' checkbox, a 'Working sets:' dropdown menu, and 'New...' and 'Select...' buttons. The bottom of the dialog has a question mark icon on the left and navigation buttons: '< Back', 'Next >', 'Finish', and 'Cancel'.

New Spring Starter Project

Name:

☒ Use default location

Location:

Type: Packaging:

Java Version: Language:

Group:

Artifact:

Version:

Description:

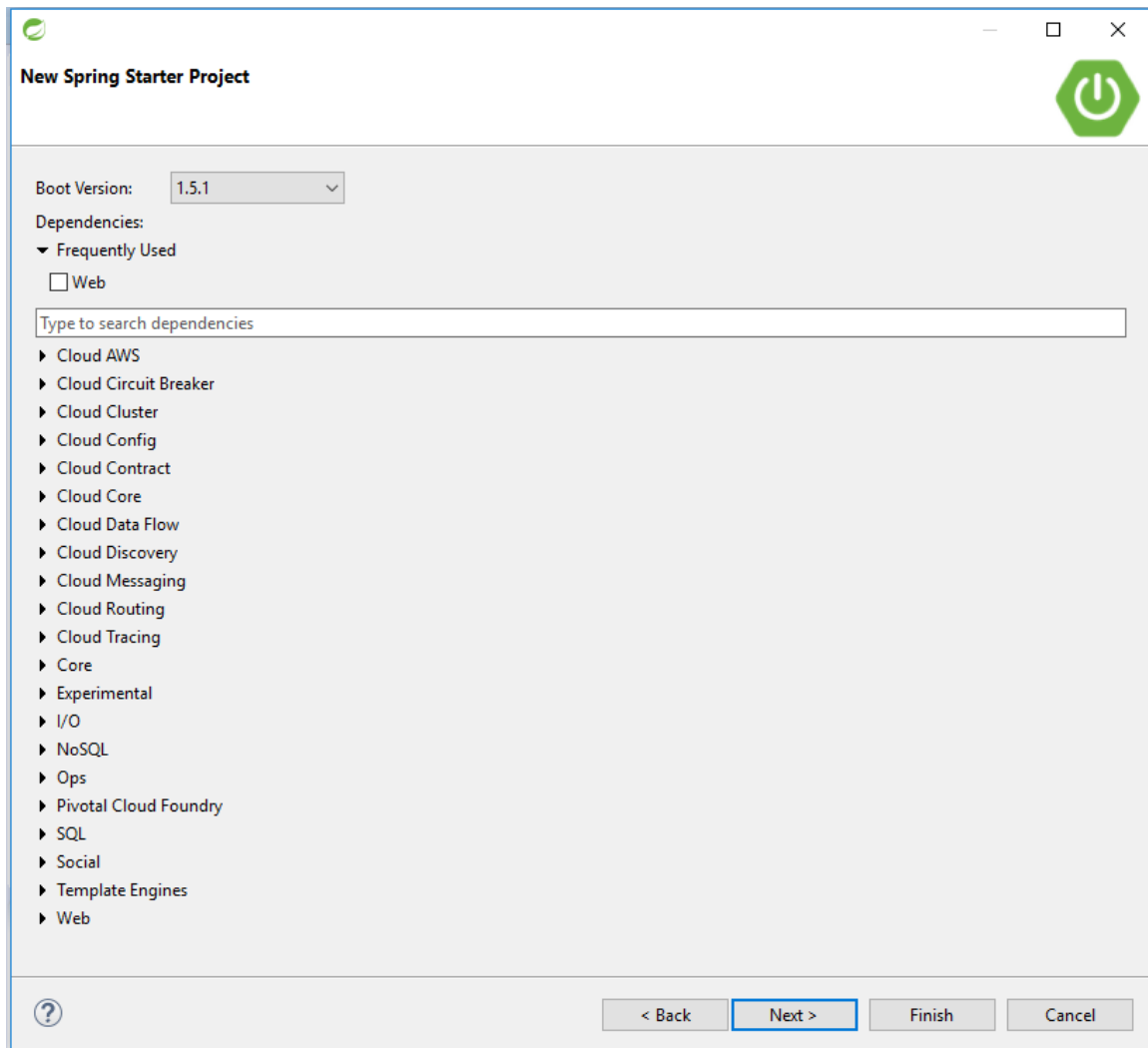
Package:

Working sets

☐ Add project to working sets

Working sets:

1.3 – Choose Nothing and click next



The image shows a 'New Spring Starter Project' dialog box. At the top, there's a title bar with a green icon and window controls. Below the title bar, the text 'New Spring Starter Project' is displayed. On the right side of the title bar, there's a green power button icon. The main area of the dialog box contains a 'Boot Version' dropdown menu set to '1.5.1'. Below this, there's a 'Dependencies' section with a 'Frequently Used' subsection. Under 'Frequently Used', there's a checkbox for 'Web' which is currently unchecked. Below the checkbox is a search bar with the placeholder text 'Type to search dependencies'. Below the search bar is a list of dependency categories, each preceded by a right-pointing triangle icon. The categories are: Cloud AWS, Cloud Circuit Breaker, Cloud Cluster, Cloud Config, Cloud Contract, Cloud Core, Cloud Data Flow, Cloud Discovery, Cloud Messaging, Cloud Routing, Cloud Tracing, Core, Experimental, I/O, NoSQL, Ops, Pivotal Cloud Foundry, SQL, Social, Template Engines, and Web. At the bottom of the dialog box, there's a row of buttons: a help button (question mark icon), a '< Back' button, a 'Next >' button (which is highlighted with a blue border), a 'Finish' button, and a 'Cancel' button.

New Spring Starter Project

Boot Version: 1.5.1

Dependencies:

▼ Frequently Used

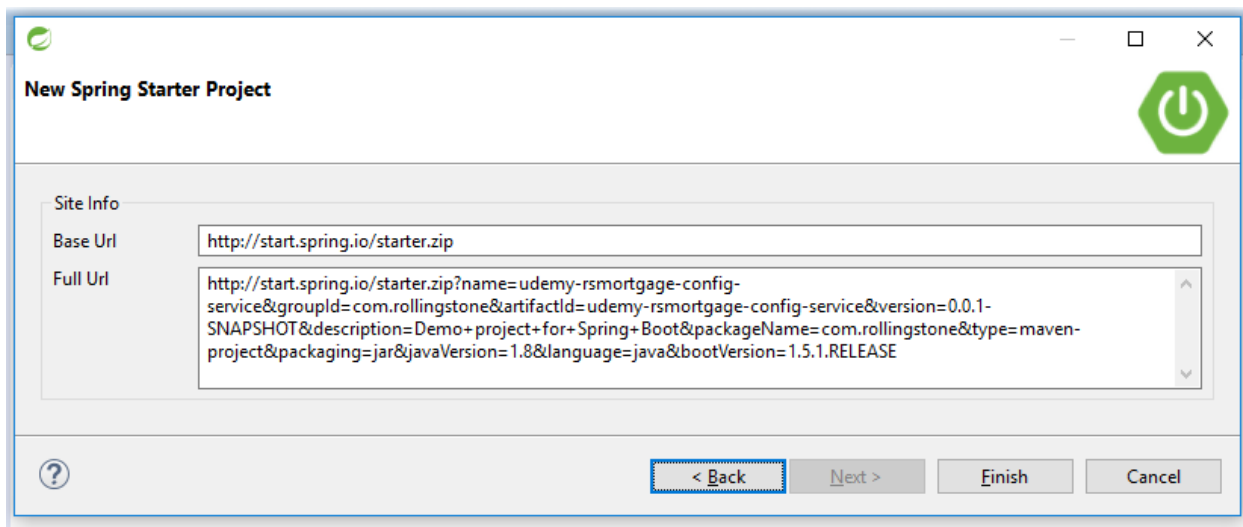
☐ Web

Type to search dependencies

- ▶ Cloud AWS
- ▶ Cloud Circuit Breaker
- ▶ Cloud Cluster
- ▶ Cloud Config
- ▶ Cloud Contract
- ▶ Cloud Core
- ▶ Cloud Data Flow
- ▶ Cloud Discovery
- ▶ Cloud Messaging
- ▶ Cloud Routing
- ▶ Cloud Tracing
- ▶ Core
- ▶ Experimental
- ▶ I/O
- ▶ NoSQL
- ▶ Ops
- ▶ Pivotal Cloud Foundry
- ▶ SQL
- ▶ Social
- ▶ Template Engines
- ▶ Web

? < Back Next > Finish Cancel

1.4 – Do not select any project references



The image shows a 'New Spring Starter Project' dialog box. It has a title bar with a green icon, a minus button, a maximize button, and a close button. The main area is divided into two sections: 'Site Info' and 'Project Info'. The 'Site Info' section contains two text fields: 'Base Url' with the value 'http://start.spring.io/starter.zip' and 'Full Url' with a long URL. The 'Project Info' section is empty. At the bottom, there are four buttons: a help button (question mark), '< Back', 'Next >', 'Finish', and 'Cancel'. The '< Back' button is highlighted with a blue border.

New Spring Starter Project

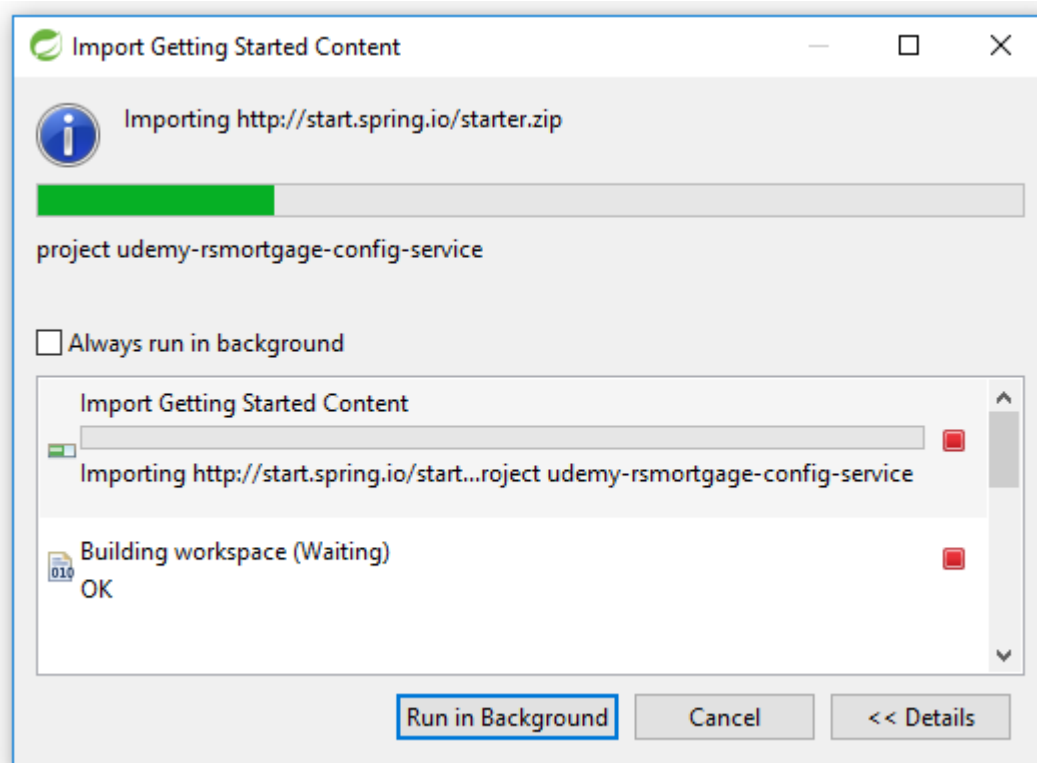
Site Info

Base Url:

Full Url:

< Back Next > Finish Cancel

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.