**Spring Cloud Config Server**

Optum

Student Exercise Manual

Contents

[Spring Cloud Config Server 3](#_Toc518587690)

# **Spring Cloud Config Server**

**Overview**

**Time: X Minutes**

In this exercise, you will:

* Generate and setup Spring Cloud Config Server to hold maximum and minimum configuration properties and will run on port 8888. This service will serve limits-service with properties.

**Step 1: Setup Git Local Repo with Config File**

* Create a folder anywhere on your machine and perform git init.
* Add this folder as an external resource (optional step).
* Create a properties file inside this folder, this is the configuration server. However, the file name has to be same as the application name for which we are creating the configuration i.e., limits-service.properties.
* Move the below entries from application.properties of limits-service to the new config project, now on limits-service will get configuration from spring cloud config.

limits-service.minimum=1

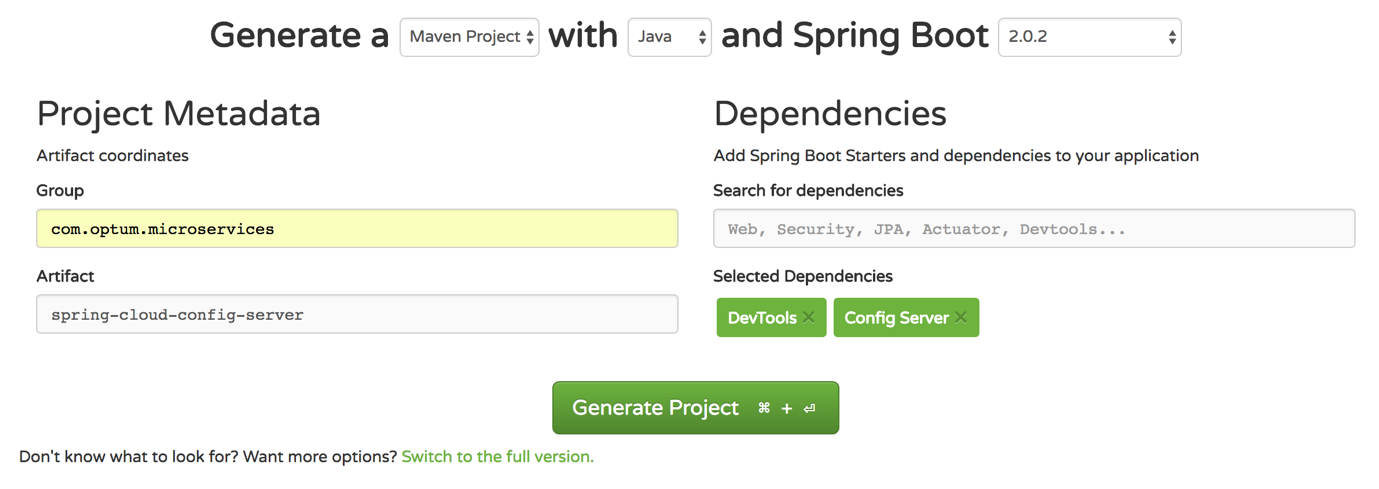
limits-service.maximum=999

* Run **git init** inside the new folder and add the newly created file.
* Commit this file locally, no need to push.

**Step 2: Generate and Setup Spring Cloud Config Server on Port 8888**

**Setup**

Create a new project from <https://start.spring.io/> and add Config Server as a dependency, as shown below.



* Click **Generate Project** and open with IDE.
* Make the below entries to **application.properties:**

spring.application.name=spring-cloud-config-server

server.port=8888

spring.cloud.config.server.git.uri=file://{git-localconfig-repo-path}

* Change git-localconfig-repo-path with full path on your machine, you can also use the github file.
* Add **@EnableConfigServer** to com.optum.microservices.springcloudconfigserver.SpringCloudConfigServerApplication

Run spring cloud config server and hit URL **<http://localhost:8888/limits-service/default>**

{"name":"limits-service","profiles":["default"],"label":null,"version":"d7bbc8de7b6d61e4da2f8f20dcfea8b2

**Step 3: Change to Limits-Service to Read from Spring Cloud Config Server**

* Rename **application.properties** inside limits-service to **bootstrap.properties**
* Add spring.cloud.config.uri=**[http://localhost:8888](http://localhost:8888/)** to **bootstrap.properties**
* Run the application and now, **limit service** will read property from spring-cloud-config-server running on port 8888.