Microservices

Monolithic?

- ➢ Traditional unified model
- ➢ Monolithic, means composed all in one piece
- ➢ A single-tiered software application in which the user interface and data access code are combined into a single program
- ➢ Tightly-coupled architecture

Advantages:

- Easy to build
- Easy to debugging and testing
- A physical monolith has no anti-pattern

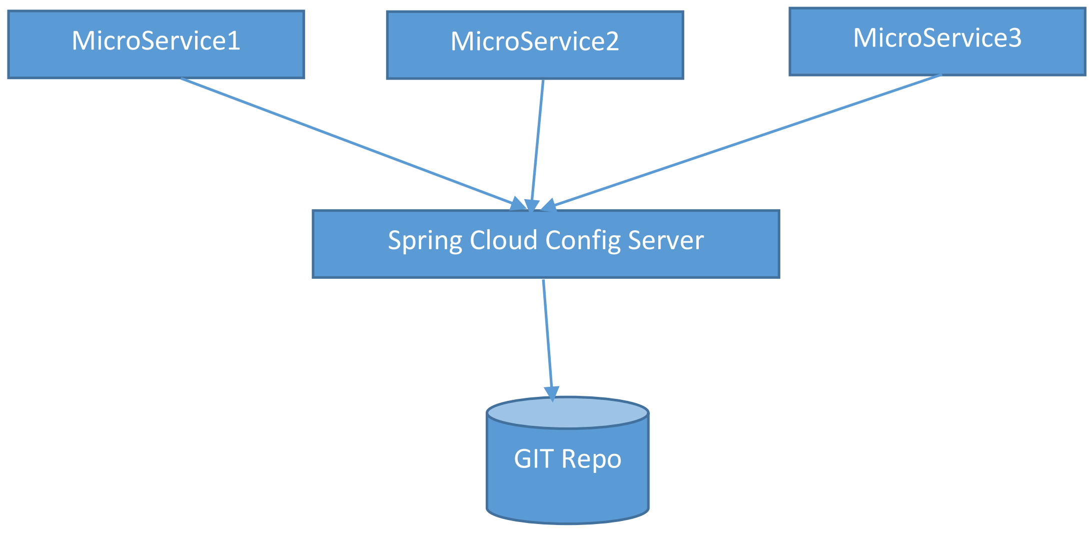
Disadvantages:

- Shared Codebase and shared data source
- Difficult to parallelize works among multiple teams.
- Large code-base.

Microservice?

- Small autonomous services that work together
- Approach to develop single application as a suite of small services
- Independently deployable
- Written in different programming languages and use different data storage

MicroServices Example:



MicroService – Limit Services:

Plan 1: Create a Micro Service (Config Client)

Plan 2: Create a Spring Cloud Config Server

Plan 3: Create GIT Repo with Shared Folder & establish connection with SCCS

Plan 4: Establish connection between MicroService & SCCS

Plan 1: To create a MicroService

Step 1: Create a Maven Project

Pom.xml

```
<?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.o
rg/POM/4.0.0
https://maven.apache.org/xsd/maven-
4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>

    <groupId>org.springframework.boot</groupI
d>
        <artifactId>spring-boot-starter-
parent</artifactId>
        <version>3.2.3</version>
        <relativePath/> <!-- lookup parent
from repository -->
    </parent>
    <groupId>com.limit.services</groupId>
    <artifactId>csd24sd1234-
limitservices</artifactId>
```

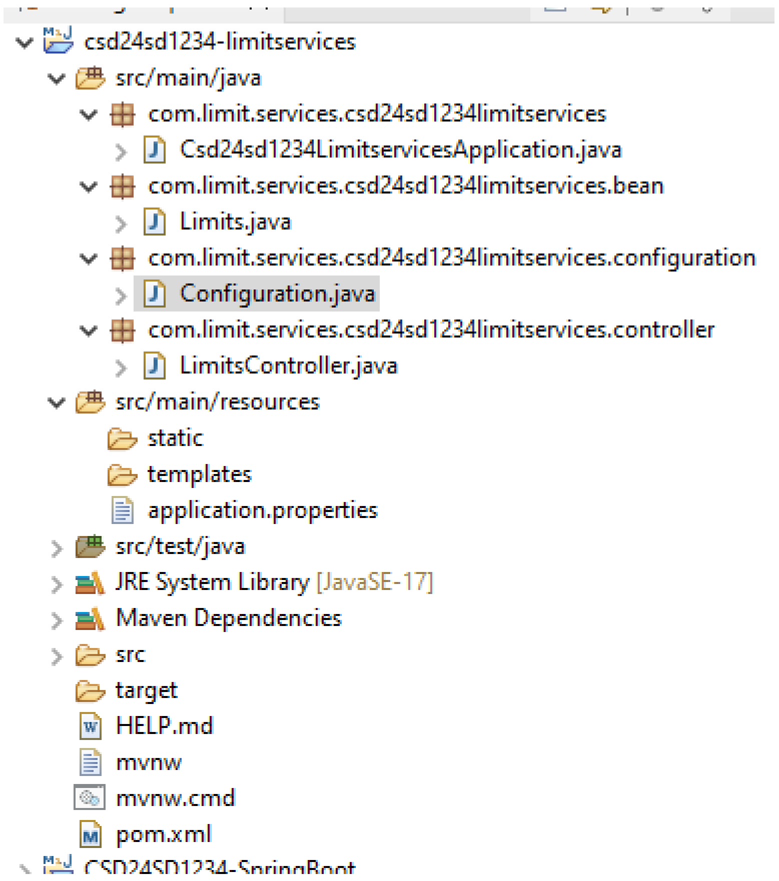
```
    <version>0.0.1-SNAPSHOT</version>
    <name>csd24sd1234-limitservices</name>
    <description>Demo project for Spring
Boot</description>
    <properties>
        <java.version>17</java.version>
        <spring-
cloud.version>2023.0.0</spring-cloud.version>
    </properties>
    <dependencies>
        <dependency>

    <groupId>org.springframework.boot</groupI
d>
            <artifactId>spring-boot-starter-
actuator</artifactId>
        </dependency>
        <dependency>

    <groupId>org.springframework.boot</groupI
d>
            <artifactId>spring-boot-starter-
web</artifactId>
        </dependency>
        <dependency>

    <groupId>org.springframework.cloud</group
Id>
            <artifactId>spring-cloud-starter-
config</artifactId>
        </dependency>

        <dependency>
    <groupId>org.springframework.boot</groupI
d>
            <artifactId>spring-boot-
devtools</artifactId>
            <scope>runtime</scope>
            <optional>true</optional>
        </dependency>
        <dependency>

    <groupId>org.springframework.boot</groupI
d>
            <artifactId>spring-boot-starter-
test</artifactId>
            <scope>test</scope>
        </dependency>
    </dependencies>
    <dependencyManagement>
        <dependencies>
            <dependency>

    <groupId>org.springframework.cloud</group
Id>
                <artifactId>spring-cloud-
dependencies</artifactId>
                <version>${spring-
cloud.version}</version>
                <type>pom</type>
                <scope>import</scope>
            </dependency>
        </dependencies>
    </dependencyManagement>
    <build>
        <plugins>
            <plugin>

    <groupId>org.springframework.boot</groupI
d>
                <artifactId>spring-boot-maven-
plugin</artifactId>
            </plugin>
        </plugins>
    </build>

</project>
```

Main Application

```
package
com.limit.services.csd24sd1234limitservices;

import
org.springframework.boot.SpringApplication;
import
org.springframework.boot.autoconfigure.Spring
BootApplication;

@SpringBootApplication
public class
Csd24sd1234LimitservicesApplication {

    public static void main(String[] args) {

    SpringApplication.run(Csd24sd1234Limitser
vicesApplication.class, args);
```

```
        }

}
```

Folder Structure



```
csd24sd1234-limitservices
  src/main/java
    com.limit.services.csd24sd1234limitservices
      Csd24sd1234LimitservicesApplication.java
    com.limit.services.csd24sd1234limitservices.bean
      Limits.java
    com.limit.services.csd24sd1234limitservices.configuration
      Configuration.java
    com.limit.services.csd24sd1234limitservices.controller
      LimitsController.java
  src/main/resources
    static
    templates
    application.properties
  src/test/java
  JRE System Library [JavaSE-17]
  Maven Dependencies
  src
  target
  HELP.md
  mvnw
  mvnw.cmd
  pom.xml
CSD24SD1234-SpringBoot
```

Run the application, see the logs reading that server not mapped & Tomcat server not started

```
***************************
APPLICATION FAILED TO START
***************************

Description:

No spring.config.import property has been defined

Action:

Add a spring.config.import=configserver: property to your configuration.
	If configuration is not required add
spring.config.import=optional:configserver: instead.
	To disable this check, set spring.cloud.config.enabled=false or
	spring.cloud.config.import-check.enabled=false.
```

Update Properties file

```
spring.config.import=optional:configserver:http://localhost:8888
```

Run the application, ensure Tomcat Server started in port 8080 & read the below logs:

```
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v3.2.3)

2024-03-19T10:59:04.664+05:30  INFO 11268 --- [csd24sd1234-limitservices] [
restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port
8080 (http) with context path ''
2024-03-19T10:59:04.931+05:30  INFO 11268 --- [csd24sd1234-limitservices] [
restartedMain] .s.c.Csd24sd1234LimitservicesApplication : Started
Csd24sd1234LimitservicesApplication in 9.73 seconds (process running for 10.825)
2024-03-19T11:01:06.955+05:30  INFO 11268 --- [csd24sd1234-limitservices] [   File
Watcher] rtingClassPathChangeChangedEventListener : Restarting due to 1 class path
change (1 addition, 0 deletions, 0 modifications)
```

Create “Limits” class:

```java
package com.limit.services.csd24sd1234limitservices.bean;

public class Limits {

    private int minimum;
    private int maximum;

    public Limits(){
        super();
    }

    public Limits(int minimum, int maximum) {
        super();
        this.minimum = minimum;
        this.maximum = maximum;
    }

    public int getMinimum() {
        return minimum;
    }
    public void setMinimum(int minimum) {
        this.minimum = minimum;
    }

    public int getMaximum() {
        return maximum;
    }
    public void setMaximum(int maximum) {
        this.maximum = maximum;
    }

}
```

Controller class:

```
package
com.limit.services.csd24sd1234limitservices.c
ontroller;

import
org.springframework.stereotype.Component;
import
org.springframework.web.bind.annotation.GetMa
pping;
import
org.springframework.web.bind.annotation.RestC
ontroller;
```

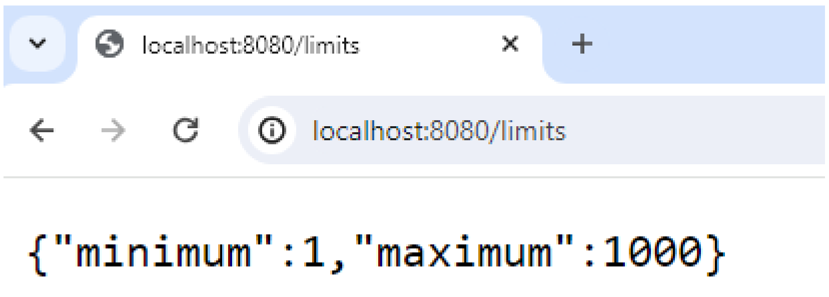
```java
import
com.limit.services.csd24sd1234limitservices.b
ean.Limits;

@Component
@RestController
public class LimitsController {

    @GetMapping("/limits")
    public Limits retrieveLimits() {
        return new Limits (1, 1000);
        }
}
```

Set up the URL:

http://localhost:8080/limits

Run the application & ensure Tomcat Server started in port 8080

Open the browser & give URL http://localhost:8080/limits to see the limit values displayed.



```
{"minimum":1,"maximum":1000}
```

Step 2: Add configuration

Make changes in Application.properties files:

```
spring.config.import=optional:configserver:ht
tp://localhost:8888
csd24sd1234-limitservices.minimum=2
csd24sd1234-limitservices.maximum=998
```

Configuration class:

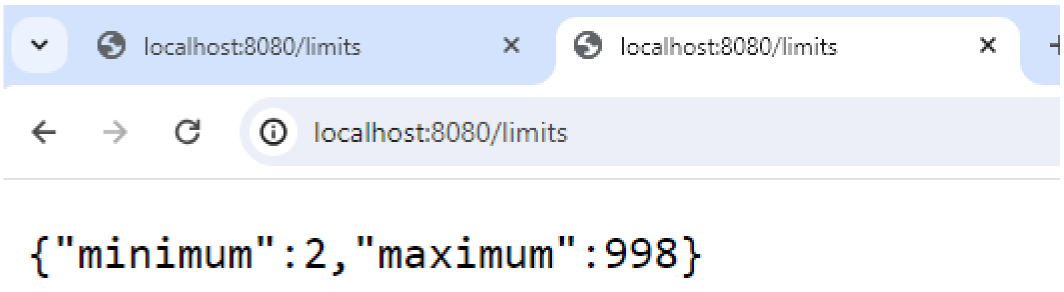
```
package
com.limit.services.csd24sd1234limitservices.c
onfiguration;

import
org.springframework.boot.context.properties.C
onfigurationProperties;
import
org.springframework.stereotype.Component;

@Component
@ConfigurationProperties("csd24sd1234-
limitservices")
public class Configuration {

    private int minimum;
    private int maximum;

    public int getMinimum() {
        return minimum;
    }
    public void setMinimum(int minimum) {
        this.minimum = minimum;
    }

    public int getMaximum() {
        return maximum;
    }
    public void setMaximum(int maximum) {
        this.maximum = maximum;
    }

}
```

Update Controller class:

```
package
com.limit.services.csd24sd1234limitservices.c
ontroller;

import
org.springframework.beans.factory.annotation.
Autowired;
import
org.springframework.stereotype.Component;
import
org.springframework.web.bind.annotation.GetMa
pping;
import
org.springframework.web.bind.annotation.RestC
ontroller;

import
com.limit.services.csd24sd1234limitservices.b
ean.Limits;
import
com.limit.services.csd24sd1234limitservices.c
onfiguration.Configuration;

@Component
@RestController
public class LimitsController {
    @Autowired
    private Configuration configuration;

    @GetMapping("/limits")
    public Limits retrieveLimits() {
//      return new Limits(1, 1000);
        return new 
Limits(configuration.getMinimum(), 
configuration.getMaximum());
    }
}
```

Ensure Tomcat Server restarted in port 8080

Refresh the browser with same URL http://localhost:8080/limits to see the new limit values displayed

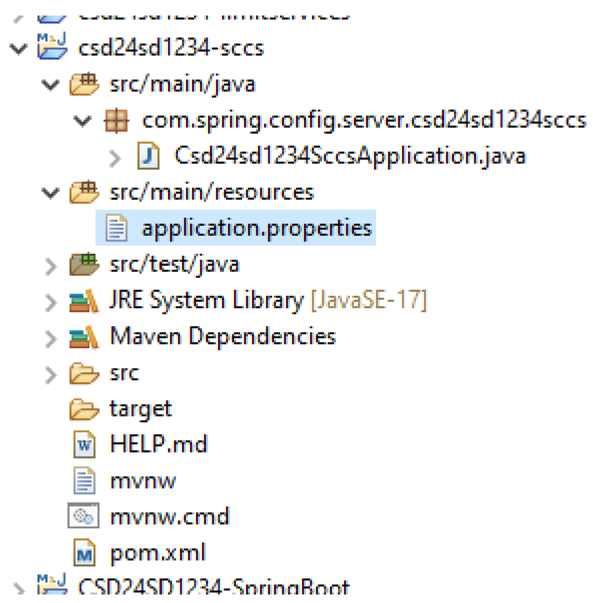


```
{"minimum":2,"maximum":998}
```

Plan 2: To create Spring Cloud Config Server

Step 1: Create a Maven Project using Spring Initializr

Folder structure:


```
csd24sd1234-sccs
  src/main/java
    com.spring.config.server.csd24sd1234sccs
      Csd24sd1234SccsApplication.java
  src/main/resources
    application.properties
  src/test/java
  JRE System Library [JavaSE-17]
  Maven Dependencies
  src
  target
  HELP.md
  mvnw
  mvnw.cmd
  pom.xml
CSD24SD1234-SpringBoot
```

Pom.xml

```xml
<?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 https://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>3.2.3</version>
		<relativePath/> <!-- lookup parent from repository -->
	</parent>
```

```
    <groupId>com.spring.config.server</groupI
d>
    <artifactId>csd24sd1234-sccs</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>csd24sd1234-sccs</name>
    <description>Demo project for Spring
Boot</description>
    <properties>
        <java.version>17</java.version>
        <spring-
cloud.version>2023.0.0</spring-cloud.version>
    </properties>
    <dependencies>
        <dependency>

    <groupId>org.springframework.cloud</group
Id>
            <artifactId>spring-cloud-config-
server</artifactId>
        </dependency>

        <dependency>

    <groupId>org.springframework.boot</groupI
d>
            <artifactId>spring-boot-
devtools</artifactId>
            <scope>runtime</scope>
            <optional>true</optional>
        </dependency>
        <dependency>
    <groupId>org.springframework.boot</groupI
d>
            <artifactId>spring-boot-starter-
test</artifactId>
            <scope>test</scope>
        </dependency>
    </dependencies>
    <dependencyManagement>
        <dependencies>
            <dependency>

    <groupId>org.springframework.cloud</group
Id>
                <artifactId>spring-cloud-
dependencies</artifactId>
                <version>${spring-
cloud.version}</version>
                <type>pom</type>
                <scope>import</scope>
            </dependency>
        </dependencies>
    </dependencyManagement>

    <build>
        <plugins>
            <plugin>

    <groupId>org.springframework.boot</groupI
d>
                <artifactId>spring-boot-maven-
plugin</artifactId>
            </plugin>
        </plugins>
    </build>

</project>
```

<u>Main Application</u>:

```
package
com.spring.config.server.csd24sd1234sccs;

import
org.springframework.boot.SpringApplication;
import
org.springframework.boot.autoconfigure.Spring
BootApplication;
import
org.springframework.cloud.config.server.Enabl
eConfigServer;

@SpringBootApplication
public class Csd24sd1234SccsApplication {

    public static void main(String[] args) {

    SpringApplication.run(Csd24sd1234SccsAppl
ication.class, args);
    }

}
```

Ensure that MicroService program is already running with Tomcat Server in port 8080

<u>Now run the Server application, see the logs reading that server not mapped & Tomcat server not started:</u>

```
Error starting ApplicationContext. To display the condition evaluation report re-
run your application with 'debug' enabled.
2024-03-20T09:42:46.276+05:30 ERROR 9272 --- [csd24sd1234-sccs] [  restartedMain]
o.s.b.d.LoggingFailureAnalysisReporter   :

***************************
APPLICATION FAILED TO START
***************************

Description:

Web server failed to start. Port 8080 was already in use.

Action:

Identify and stop the process that's listening on port 8080 or configure this
application to listen on another port.
```

Add below to Application.Properties:

```
spring.application.name=csd24sd1234-sccs
server.port=8888
```

Run the application & ensure Tomcat Server started in port 8888

```
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v3.2.3)

2024-03-20T09:45:05.847+05:30  INFO 7688 --- [csd24sd1234-sccs] [  restartedMain]
o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded
WebApplicationContext
2024-03-20T09:45:05.850+05:30  INFO 7688 --- [csd24sd1234-sccs] [  restartedMain]
w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext:
initialization completed in 4643 ms
2024-03-20T09:45:06.732+05:30  WARN 7688 --- [csd24sd1234-sccs] [  restartedMain]
o.s.b.d.a.OptionalLiveReloadServer       : Unable to start LiveReload server
2024-03-20T09:45:07.061+05:30  INFO 7688 --- [csd24sd1234-sccs] [  restartedMain]
o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port 8888 (http) with
context path ''
2024-03-20T09:45:07.231+05:30  INFO 7688 --- [csd24sd1234-sccs] [  restartedMain]
c.s.c.s.c.Csd24sd1234SccsApplication     : Started Csd24sd1234SccsApplication in
7.413 seconds (process running for 8.69)
2024-03-20T10:01:17.539+05:30  INFO 7688 --- [csd24sd1234-sccs] [   File Watcher]
rtingClassPathChangeChangedEventListener : Restarting due to 1 class path change
(0 additions, 0 deletions, 1 modification)
```

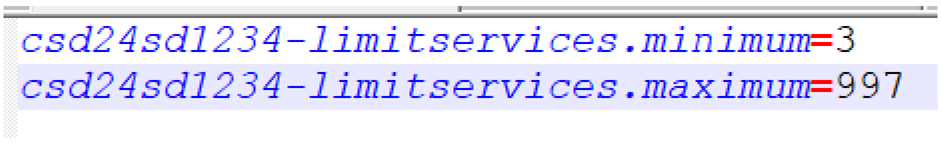
Plan 3: To create Shared Repository & Set up GIT

Step 1: Install GIT client:

Download & Install GIT client in local machine

Create a folder “`csd24sd1234-git-repo`” in one of the directory

Create a file “csd24sd1234-limitservices.properties” via a text editor & capture below statement:



```
csd24sd1234-limitservices.minimum=3
csd24sd1234-limitservices.maximum=997
```

Run below commands in GIT to configure & set up shared location:

```
windows@DESKTOP-O7DEUMC MINGW64 ~
$ pwd
/c/Users/windows

windows@DESKTOP-O7DEUMC MINGW64 ~
$ cd /e/java

windows@DESKTOP-O7DEUMC MINGW64 /e/java
$ pwd
/e/java

windows@DESKTOP-O7DEUMC MINGW64 /e/java
$ cd csd24sd1234-git-repo

windows@DESKTOP-O7DEUMC MINGW64 /e/java/csd24sd1234-git-repo
$ pwd
/e/java/csd24sd1234-git-repo

windows@DESKTOP-O7DEUMC MINGW64 /e/java/csd24sd1234-git-repo
$ git init
Initialized empty Git repository in E:/Java/csd24sd1234-git-repo/.git/

windows@DESKTOP-O7DEUMC MINGW64 /e/java/csd24sd1234-git-repo (master)
$ ls
csd24sd1234-limitservices.properties

windows@DESKTOP-O7DEUMC MINGW64 /e/java/csd24sd1234-git-repo (master)
$ git add *

windows@DESKTOP-O7DEUMC MINGW64 /e/java/csd24sd1234-git-repo (master)
$ git commit -m "adding csd24sd1234-limitservices.properties"
Author identity unknown

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'windows@DESKTOP-O7DEUMC.(none)')

windows@DESKTOP-O7DEUMC MINGW64 /e/java/csd24sd1234-git-repo (master)
$ git config user.email "maheswaran.s@cognizant.com"

windows@DESKTOP-O7DEUMC MINGW64 /e/java/csd24sd1234-git-repo (master)
$ git config user.name "Maheswaran"
```

```
windows@DESKTOP-O7DEUMC MINGW64 /e/java/csd24sd1234-git-repo (master)
$ git commit -m "adding csd24sd1234-limitservices.properties"
[master (root-commit) 4074df5] adding csd24sd1234-limitservices.properties
 1 file changed, 2 insertions(+)
 create mode 100644 csd24sd1234-limitservices.properties

windows@DESKTOP-O7DEUMC MINGW64 /e/java/csd24sd1234-git-repo (master)
$
```

Step 2: Connect Spring Cloud Config Server to GIT Repo

Make below changes in Application.Properties files:

```
spring.application.name=csd24sd1234-sccs
server.port=8888
spring.cloud.config.server.git.uri=file:///e:
/java/csd24sd1234-git-repo
```

Make below change in main application:

```
package
com.spring.config.server.csd24sd1234sccs;

import
org.springframework.boot.SpringApplication;
import
org.springframework.boot.autoconfigure.Spring
BootApplication;
import
org.springframework.cloud.config.server.Enabl
eConfigServer;

@EnableConfigServer
@SpringBootApplication
public class Csd24sd1234SccsApplication {

    public static void main(String[] args) {

    SpringApplication.run(Csd24sd1234SccsAppl
ication.class, args);
```

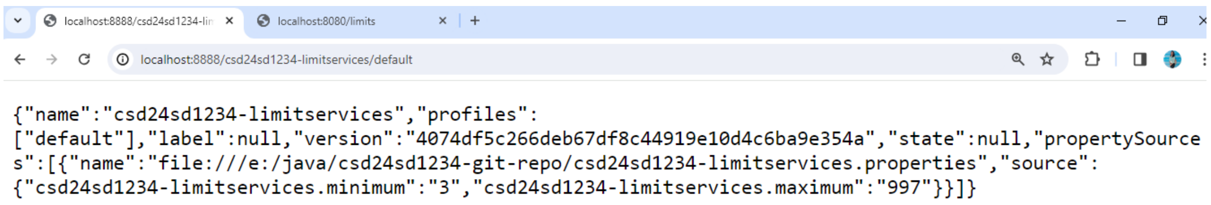
```
        }

}
```

Set up the URL:

http://localhost:8888/csd24sd1234-limitservices/default

Restart the Tomcat Server started in port 8888

Refresh the browser with URL http://localhost:8888/csd24sd1234-limitservices/default to see the new limit values displayed from git repository location



localhost:8888/csd24sd1234-lin
localhost:8080/limits
localhost:8888/csd24sd1234-limitservices/default

```
{"name":"csd24sd1234-limitservices","profiles":
["default"],"label":null,"version":"4074df5c266deb67df8c44919e10d4c6ba9e354a","state":null,"propertySource
s":[{"name":"file:///e:/java/csd24sd1234-git-repo/csd24sd1234-limitservices.properties","source":
{"csd24sd1234-limitservices.minimum":"3","csd24sd1234-limitservices.maximum":"997"}}]}
```

Plan 4: To connect Limits Services with Spring Cloud Config Server

Step 1: Updates to Limit Services application

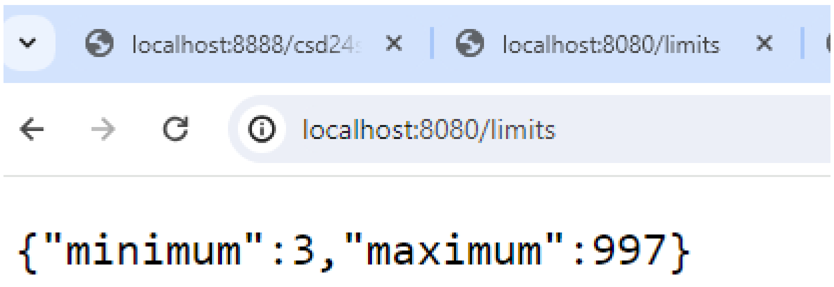
Make change to Application.Properties

```
spring.application.name=csd24sd1234-limitservices
spring.config.import=optional:configserver:http://localhost:8888
csd24sd1234-limitservices.minimum=2
csd24sd1234-limitservices.maximum=998
```

Save & Restart the server

Refresh the browser with URL: http://localhost:8080/limits

Check the values from GIT repository properties file displayed in the browser

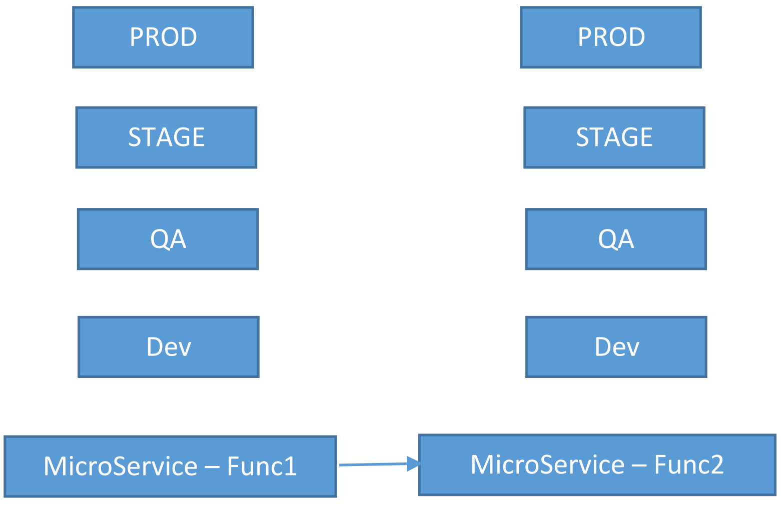


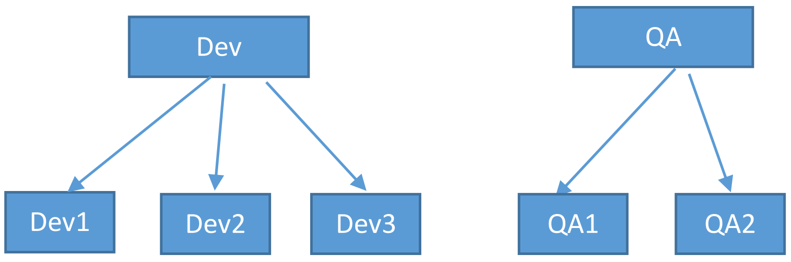
```
{"minimum":3,"maximum":997}
```

Path set up in log of config client:

```
http://localhost:8888/csd24sd1234-
limitservices/default
```

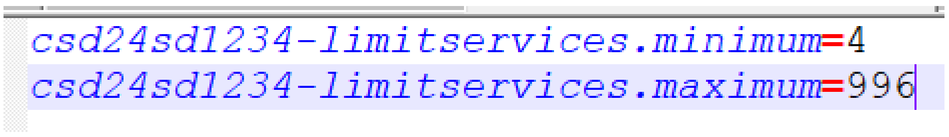
Multiple Environments & respective profiles:





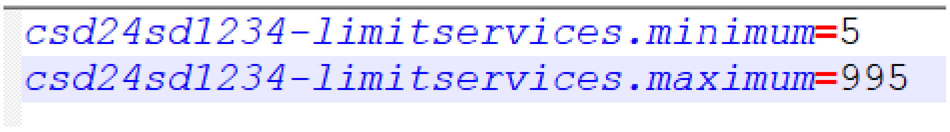
<u>Step 2: Create more properties files for other environments like DEV & QA</u>

<u>csd24sd1234-limitservices-dev.properties</u>



```
csd24sd1234-limitservices.minimum=4
csd24sd1234-limitservices.maximum=996
```

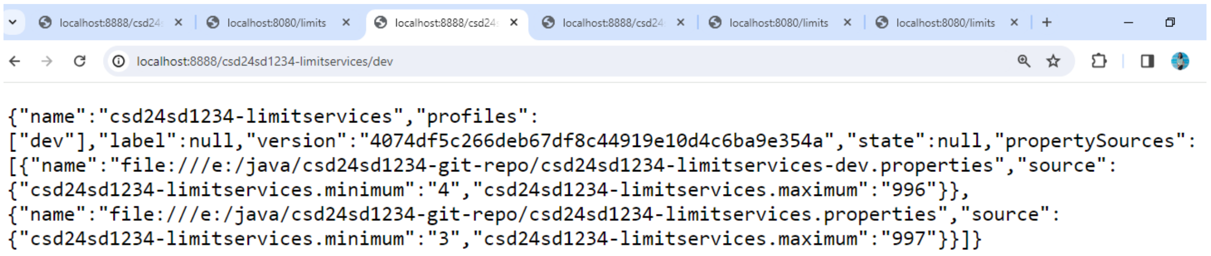
csd24sd1234-limitservices-qa.properties



```
csd24sd1234-limitservices.minimum=5
csd24sd1234-limitservices.maximum=995
```

Refresh the browser with URL with profile dev: http://localhost:8888/intqea23sd001-limitservices/dev

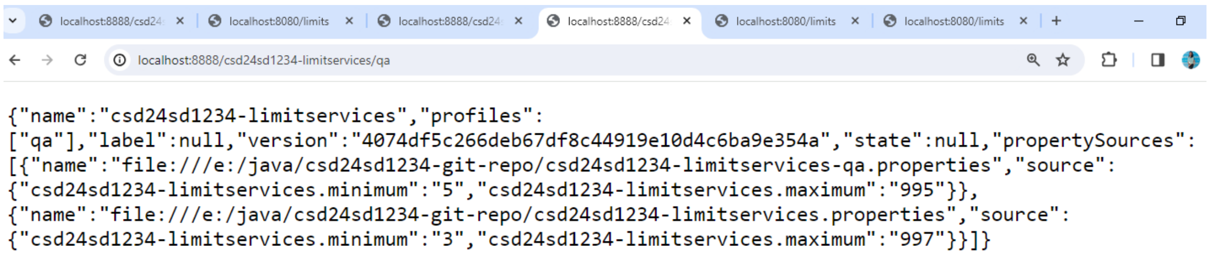
Check the values from GIT repository dev properties file displayed in the browser



```
{"name":"csd24sd1234-limitservices","profiles":
["dev"],"label":null,"version":"4074df5c266deb67df8c44919e10d4c6ba9e354a","state":null,"propertySources":
[{"name":"file:///e:/java/csd24sd1234-git-repo/csd24sd1234-limitservices-dev.properties","source":
{"csd24sd1234-limitservices.minimum":"4","csd24sd1234-limitservices.maximum":"996"}},
{"name":"file:///e:/java/csd24sd1234-git-repo/csd24sd1234-limitservices.properties","source":
{"csd24sd1234-limitservices.minimum":"3","csd24sd1234-limitservices.maximum":"997"}}]}
```

Refresh the browser with URL with profile qa : http://localhost:8888/intqea23sd001-limitservices/qa

Check the values from GIT repository qa properties file displayed in the browser



```
{"name":"csd24sd1234-limitservices","profiles":
["qa"],"label":null,"version":"4074df5c266deb67df8c44919e10d4c6ba9e354a","state":null,"propertySources":
[{"name":"file:///e:/java/csd24sd1234-git-repo/csd24sd1234-limitservices-qa.properties","source":
{"csd24sd1234-limitservices.minimum":"5","csd24sd1234-limitservices.maximum":"995"}},
{"name":"file:///e:/java/csd24sd1234-git-repo/csd24sd1234-limitservices.properties","source":
{"csd24sd1234-limitservices.minimum":"3","csd24sd1234-limitservices.maximum":"997"}}]}
```

To populate the config values from MicroService, do the following:

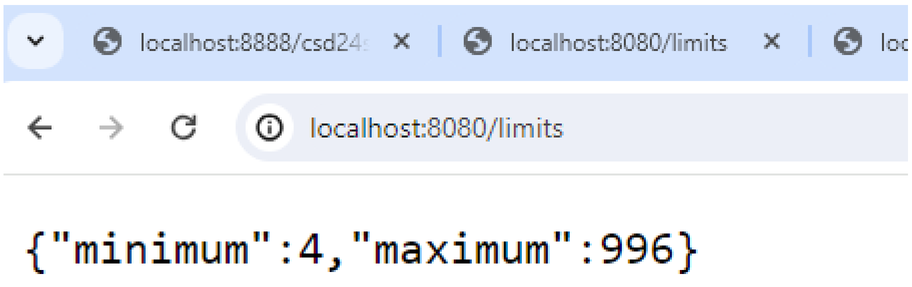
Make change to Application.Properties of Limits application to include profile "dev"

```
spring.profile.active=dev
spring.cloud.config.profile=dev
```

Save & Restart the 8080 server

Refresh the browser with URL: http://localhost:8080/limits

Check the values from GIT repository properties file for DEV is displayed in the browser



```
{"minimum":4,"maximum":996}
```

Path set up in log of config client:

```
http://localhost:8888/csd24sd1234-
limitservices/dev
```

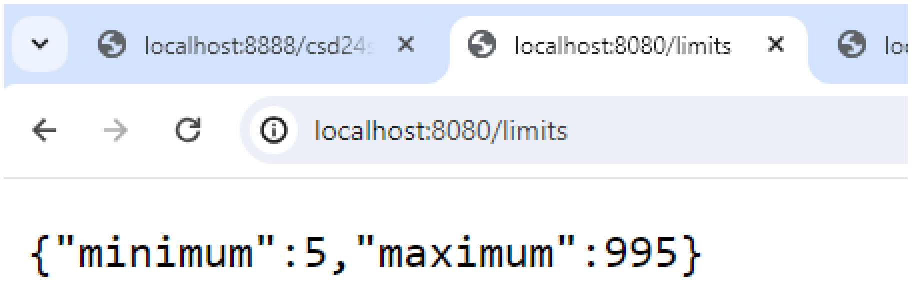
Make change to Application.Properties of Limits application to include profile “qa”

```
spring.profile.active=qa
spring.cloud.config.profile=qa
```

Save & Restart the 8080 server

Refresh the browser with URL: http://localhost:8080/limits

Check the values from GIT repository properties file for DEV is displayed in the browser



```
{"minimum":5,"maximum":995}
```

Path set up in log of config client:

```
http://localhost:8888/csd24sd1234-
limitservices/qa
```