

## Create a Microservice (Config Client)

- Create a Maven project, add following 4 dependencies and generate it.

The screenshot shows the start.spring.io web interface. On the left, there's a sidebar with a hamburger menu. The main area is divided into sections: 'Project' with radio buttons for Gradle - Groovy, Gradle - Kotlin, and Maven (selected); 'Language' with radio buttons for Java (selected), Kotlin, and Groovy; 'Spring Boot' with radio buttons for 3.3.0 (SNAPSHOT), 3.3.0 (M3), 3.2.5 (SNAPSHOT), 3.2.4 (selected), and 3.1.11 (SNAPSHOT); and 'Project Metadata' with input fields for Group (com.limit.services), Artifact (demo-limit-services), Name (demo-limit-services), and Description (Demo project for Spring Boot). On the right, the 'Dependencies' section lists 'Spring Boot DevTools' (DEVELOPER TOOLS), 'Spring Web' (WEB), 'Config Client' (SPRING CLOUD CONFIG), and 'Spring Boot Actuator' (OPS). At the bottom, there are buttons for 'GENERATE' (CTRL + G), 'EXPLORE' (CTRL + SPACE), and 'SHARE...'. A GitHub logo is in the bottom left corner.

- Extract downloaded project and import it in Eclipse.
- Open application.properties and following property to run tomcat server.

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

- Create a sub package in demolimitservices with name bean and create Limits class

```
package com.limit.services.demolimitservices.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;
    }
}

```

- Create another subpackage with name controller and create LimitController class.

```

package com.limit.services.demolimitservices.controller;

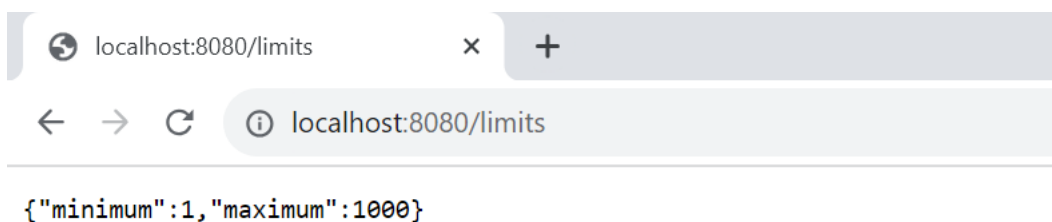
import org.springframework.stereotype.Component;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

import com.limit.services.demolimitservices.bean.Limits;

@Component
@RestController
public class LimitsController {
    @GetMapping("/limits")
    public Limits retrieveLimits() {
        return new Limits (1, 1000);
    }
}

```

- Open chrome browser and give url <http://localhost8080/limits> and see limit values



- Now add following configuration in application.properties file

```

demo-limit-services.maximum=998
demo-limit-services.minimum=2

```

- Create subpackage configuration and create Configuration class to fetch values from properties file

```
package com.limit.services.demolimitservices.configuration;

import org.springframework.boot.context.properties.ConfigurationProperties;
import org.springframework.stereotype.Component;

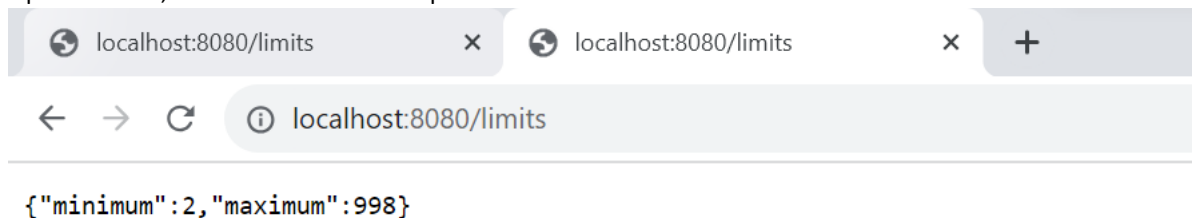
@Component
@ConfigurationProperties("demo-limit-services")
public class Configuration {

    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;
    }
}
```

- Change the retrieveLimits() as follows

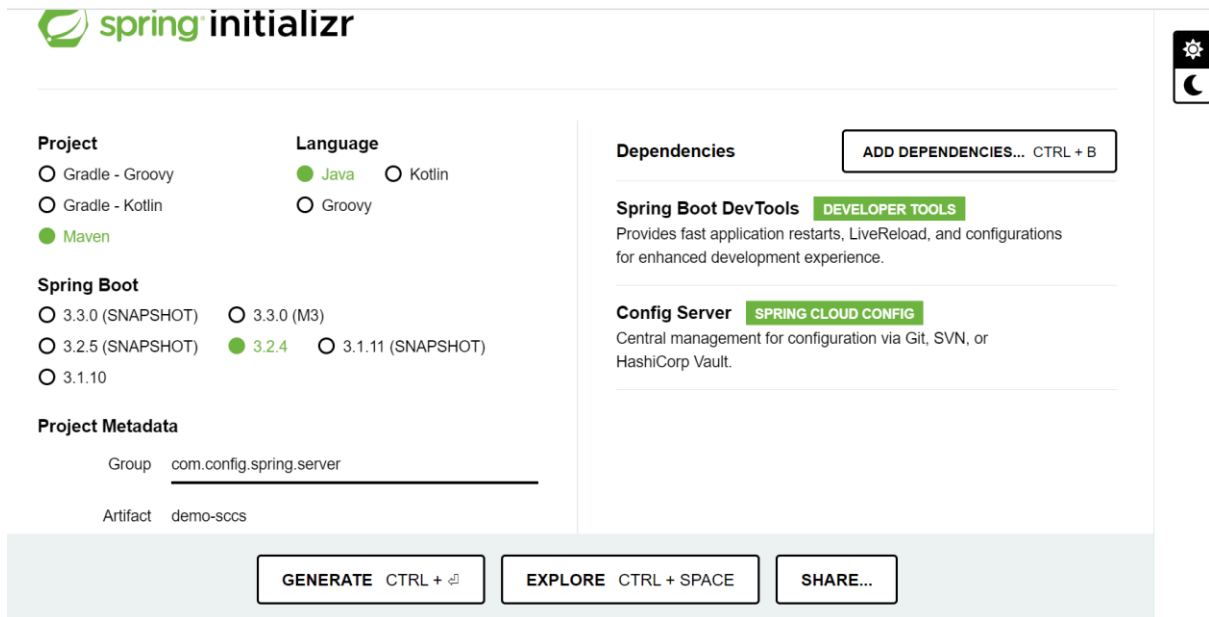
```
@GetMapping("/limits")
public Limits retrieveLimits() {
    //return new Limits (1, 1000);
    return new
Limits(configuration.getMinimum(), configuration.getMaximum());
}
```

- Open new tab, load same url and see updated values



## Create a Spring Cloud Server

- Create a maven project with Spring boot dev tools and config server as dependencies.



The Spring Initializr web interface is shown with the following configuration:

- Project:** Maven (selected)
- Language:** Java (selected)
- Spring Boot:** 3.2.4 (selected)
- Project Metadata:** Group: com.config.spring.server, Artifact: demo-sccs
- Dependencies:** Spring Boot DevTools (DEVELOPER TOOLS), Config Server (SPRING CLOUD CONFIG)

Buttons at the bottom: GENERATE CTRL + G, EXPLORE CTRL + SPACE, SHARE...

- Add following property to change server port.  
`server.port=8888`
- Create a folder “demo-git-repo” and inside it create a properties source file “demo-limit-services.properties” and add following properties.

```
demo-limit-services.minimum=3
demo-limit-services.maximum=997
```

- Open git bash and run following commands

```
MINGW64/c/demo-git-repo
CTS+2317616@LTIN522931 MINGW64 ~
$ pwd
/c/Users/2317616
CTS+2317616@LTIN522931 MINGW64 ~
$ ls -l
bash: ls -l: command not found
CTS+2317616@LTIN522931 MINGW64 ~
$ pwd
/c/Users/2317616
CTS+2317616@LTIN522931 MINGW64 ~
$ cd C:\demo-git-repo
CTS+2317616@LTIN522931 MINGW64 /c/demo-git-repo
$ git init
Initialized empty Git repository in C:/demo-git-repo/.git/
CTS+2317616@LTIN522931 MINGW64 /c/demo-git-repo (master)
$ ls
list-service-properties.txt
CTS+2317616@LTIN522931 MINGW64 /c/demo-git-repo (master)
$ git add *
CTS+2317616@LTIN522931 MINGW64 /c/demo-git-repo (master)
$ git commit -m "adding demo-list-service-properties"
[master (root-commit) 954db7b] adding demo-list-service-properties
Committer: Sen <2317616@cognizant.com>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 3 insertions(+)
create mode 100644 list-service-properties.txt
CTS+2317616@LTIN522931 MINGW64 /c/demo-git-repo (master)
$ |
```

- To connect spring cloud server to git repo, add following in application.properties

```
spring.cloud.config.server.git.uri=file:///C:/demo-git-repo
```

- Make following changes in main method

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

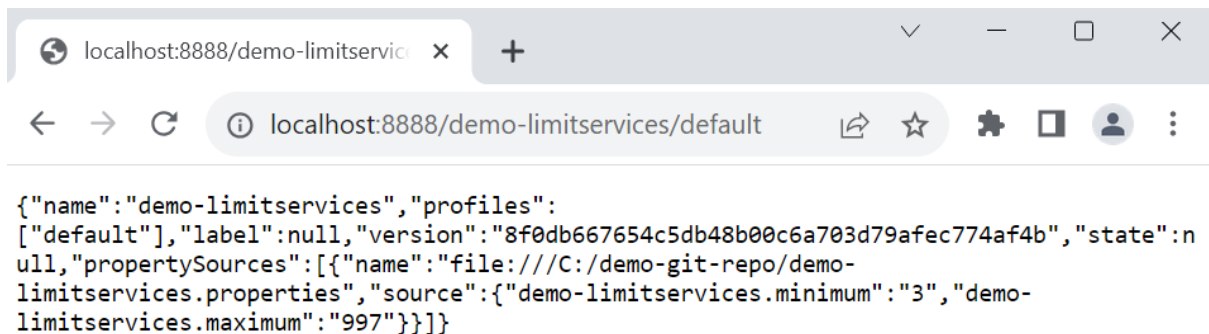
```
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.config.server.EnableConfigServer;
@EnableConfigServer
@SpringBootApplication
public class DemoSccsApplication {

    public static void main(String[] args) {
        SpringApplication.run(DemoSccsApplication.class, args);
    }

}
```

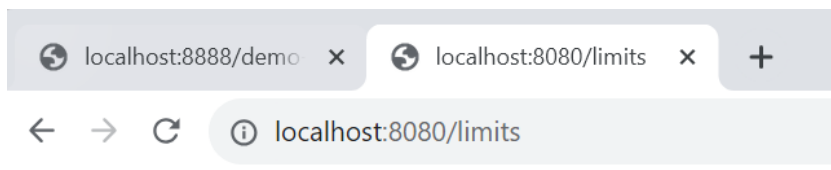
- Save all changes, restart tomcat server running at port 8888, open browser and add put following url.

<http://localhost:8888/demo-limitservices/default>



## Connect Limits service with Spring Cloud Config Server

- Add application name back to application.properties of demo-limitservices  
`spring.application.name=demo-limitservices`
- Save and restart server and enter following url in new tab.  
<http://localhost:8080/limits>  
Check if values is fetched from git repository



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

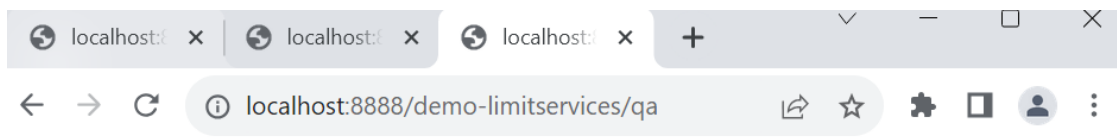
- Create new properties files for other environments like dev and qa  
`demo-limitservices-dev`

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

demo-limit**services-qa**

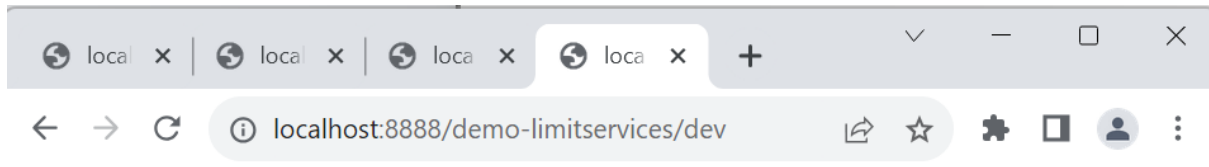
```
demo-limitservices.minimum=5  
demo-limitservices.maximum=995
```

- Enter the following url and check values from git repository qa properties are displayed.  
[http://localhost:8888/demo-limit\*\*services\*\*/qa](http://localhost:8888/demo-limit<b>services</b>/qa)



```
{"name":"demo-limitservices","profiles":  
["qa"],"label":null,"version":"8f0db667654c5db48b00c6a703d79afec774af4b","state":null,"  
propertySources":[{"name":"file:///C:/demo-git-repo/demo-limitservices-  
qa.properties","source":{"demo-limitservices.minimum":"5","demo-  
limitservices.maximum":"995"}},{"name":"file:///C:/demo-git-repo/demo-  
limitservices.properties","source":{"demo-limitservices.minimum":"3","demo-  
limitservices.maximum":"997"}}]}
```

- For dev properties, enter the following url  
<http://localhost:8888/demo-limitservices/dev>

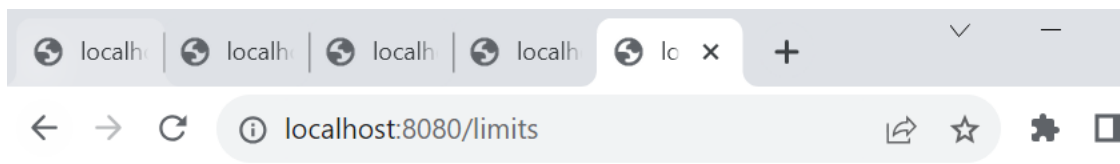


```
{
  "name": "demo-limitservices",
  "profiles": [
    {
      "dev": {
        "label": null,
        "version": "8f0db667654c5db48b00c6a703d79afec774af4b",
        "state": null,
        "propertySources": [
          {
            "name": "file:///C:/demo-git-repo/demo-limitservices-dev.properties",
            "source": {
              "demo-limitservices.minimum": "4",
              "demo-limitservices.maximum": "996"
            }
          },
          {
            "name": "file:///C:/demo-git-repo/demo-limitservices.properties",
            "source": {
              "demo-limitservices.minimum": "3",
              "demo-limitservices.maximum": "997"
            }
          }
        ]
      }
    }
  ]
}
```

- To populate config values from Microservice change application.properties of Limits application as follows to include profile “dev”

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

- Enter following url and check values from GIT repo properties file for dev is displayed.



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