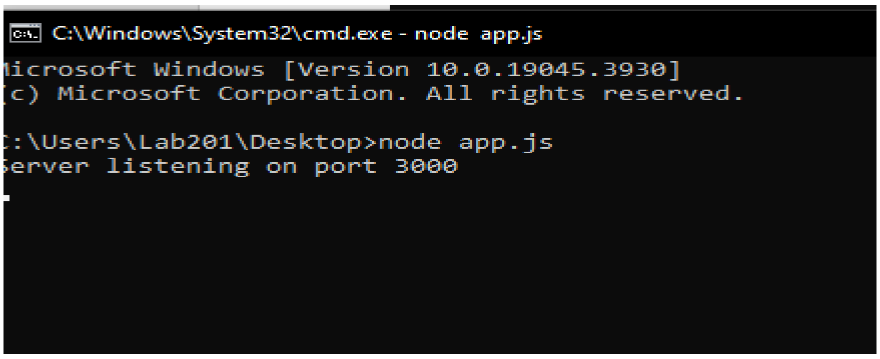
PROGRAM: (CONVERTING RS INTO DOLARS)

```
const express = require('express');
const axios = require('axios');

const app = express();

const exchangeRateApi = 'https://api.exchangerate-api.com/v4/latest/INR';

app.get('/convert', async (req, res) => {
  const { amount, to } = req.query;

  try {
    const response = await axios.get(exchangeRateApi);
    const exchangeRate = response.data.rates[to];

    if (!exchangeRate) {
      return res.status(400).json({ error: 'Invalid currency code' });
    }

    const convertedAmount = (amount * exchangeRate).toFixed(2);
    res.json({ convertedAmount });
  } catch (error) {
    console.error(error);
    res.status(500).json({ error: 'An error occurred while converting currency' });
  }
});

const PORT = process.env.PORT || 3000;
app.listen(PORT, () => {
  console.log(`Server listening on port ${PORT}`);
});
```

Go to desktop>cmd> type node filename.js (node app.js)

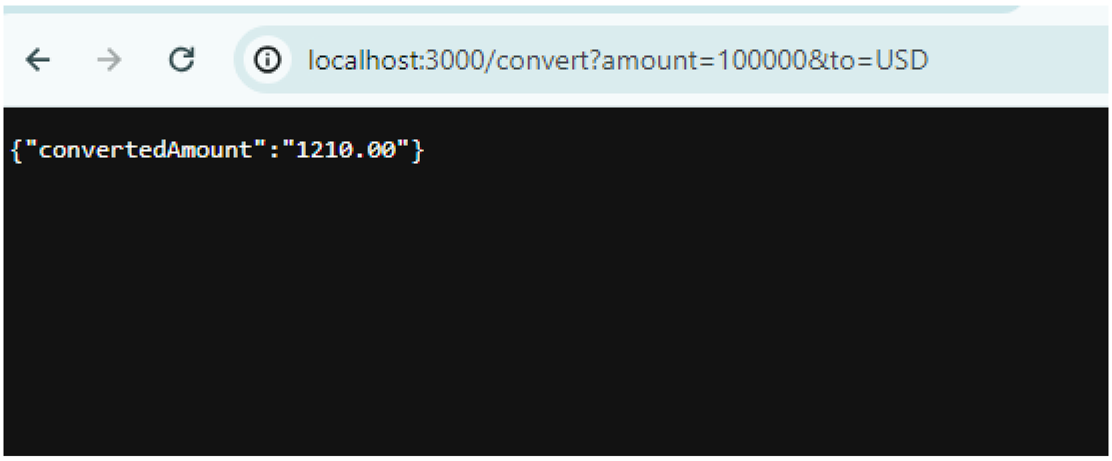
Copy port no (3000)



```
C:\Windows\System32\cmd.exe - node app.js
Microsoft Windows [Version 10.0.19045.3930]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Lab201\Desktop>node app.js
Server listening on port 3000
```

Go to browser type--- localhost:3000/convert?amount=100000&to=USD

OUTPUT:

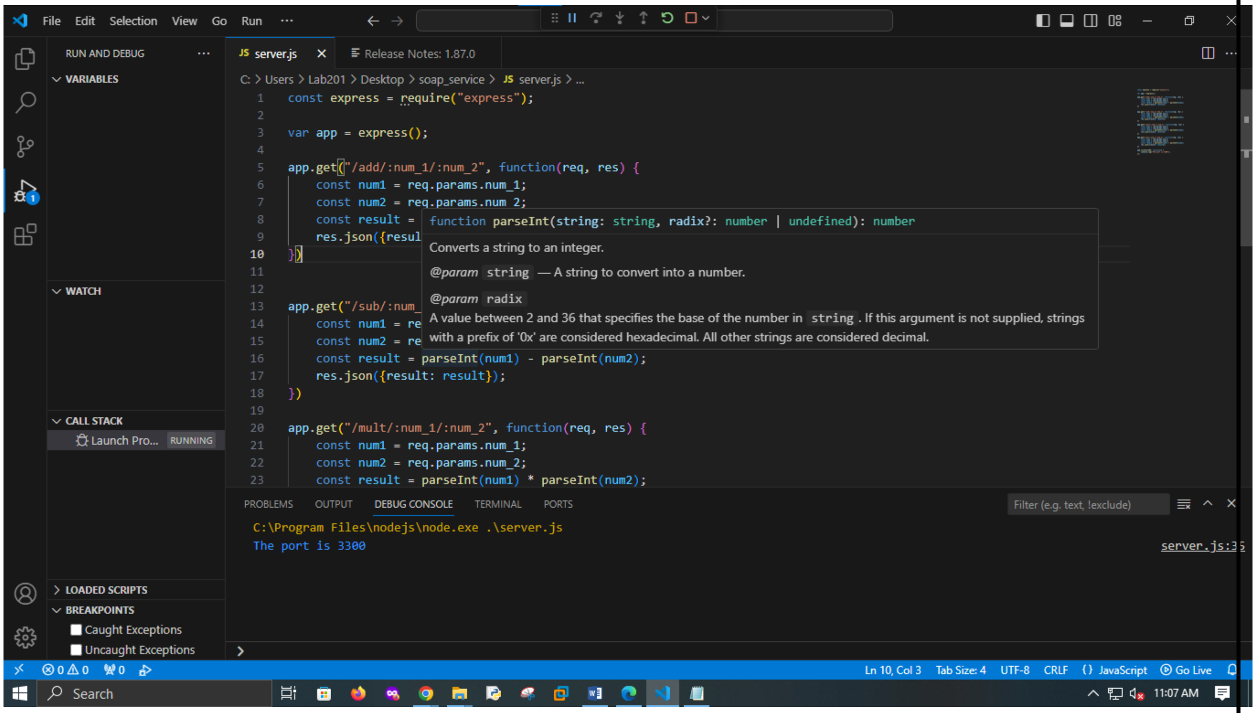


```
localhost:3000/convert?amount=100000&to=USD

{"convertedAmount":"1210.00"}
```

<u>PROGRAM:</u> (SIMPLE SOAP SERVICE USING NODE.js)

```
const express = require("express");

var app = express();

app.get("/add/:num_1/:num_2", function(req, res) {
        const num1 = req.params.num_1;
        const num2 = req.params.num_2;
        const result = parseInt(num1) + parseInt(num2);
        res.json({result: result});
})


app.get("/sub/:num_1/:num_2", function(req, res) {
        const num1 = req.params.num_1;
        const num2 = req.params.num_2;
        const result = parseInt(num1) - parseInt(num2);
        res.json({result: result});
})

app.get("/mult/:num_1/:num_2", function(req, res) {
        const num1 = req.params.num_1;
        const num2 = req.params.num_2;
        const result = parseInt(num1) * parseInt(num2);
        res.json({result: result});
})

app.get("/div/:num_1/:num_2", function(req, res) {
        const num1 = req.params.num_1;
        const num2 = req.params.num_2;
        const result = parseInt(num1) / parseInt(num2);
        res.json({result: result});
})

app.listen(3300, function() {
        console.log("The port is 3300");
});
```

Run in vs code copy the local host port number



File Edit Selection View Go Run ···
RUN AND DEBUG
VARIABLES
WATCH
CALL STACK
Launch Pro... RUNNING
LOADED SCRIPTS
BREAKPOINTS
Caught Exceptions
Uncaught Exceptions
server.js
Release Notes: 1.87.0
C: > Users > Lab201 > Desktop > soap_service > server.js > ...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Filter (e.g. text, !exclude)
Ln 10, Col 3 Tab Size: 4 UTF-8 CRLF {} JavaScript Go Live
Search
11:07 AM

```
const express = require("express");

var app = express();

app.get("/add/:num_1/:num_2", function(req, res) {
    const num1 = req.params.num_1;
    const num2 = req.params.num_2;
    const result =
    res.json({resul
})


app.get("/sub/:num_
    const num1 = re
    const num2 = re
    const result = parseInt(num1) - parseInt(num2);
    res.json({result: result});
})

app.get("/mult/:num_1/:num_2", function(req, res) {
    const num1 = req.params.num_1;
    const num2 = req.params.num_2;
    const result = parseInt(num1) * parseInt(num2);
```

```
function parseInt(string: string, radix?: number | undefined): number
Converts a string to an integer.
@param string — A string to convert into a number.
@param radix
A value between 2 and 36 that specifies the base of the number in string. If this argument is not supplied, strings
with a prefix of '0x' are considered hexadecimal. All other strings are considered decimal.
```

```
C:\Program Files\nodejs\node.exe .\server.js
The port is 3300                                                    server.js:35
```

Open google chrome type--

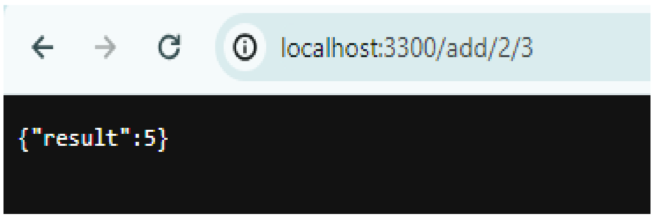
http://localhost:3300/add/2/3

http://localhost:3300/sub/2/3

http://localhost:3300/mult/2/3

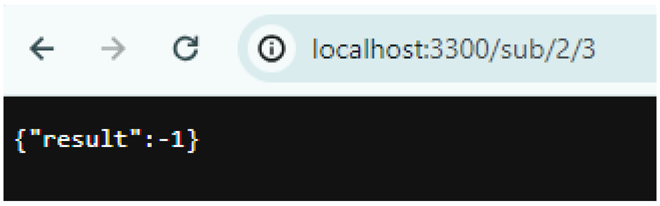
http://localhost:3300/div/2/3

OUTPUT:



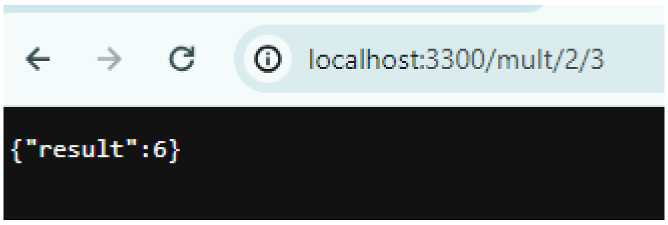
localhost:3300/add/2/3

```
{"result":5}
```



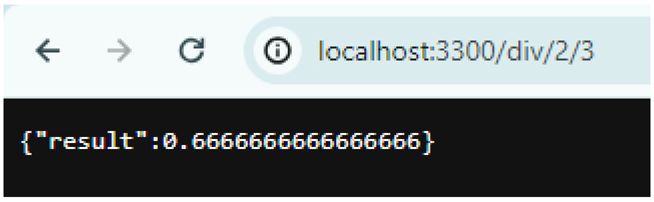
localhost:3300/sub/2/3

```
{"result":-1}
```



localhost:3300/mult/2/3

```
{"result":6}
```

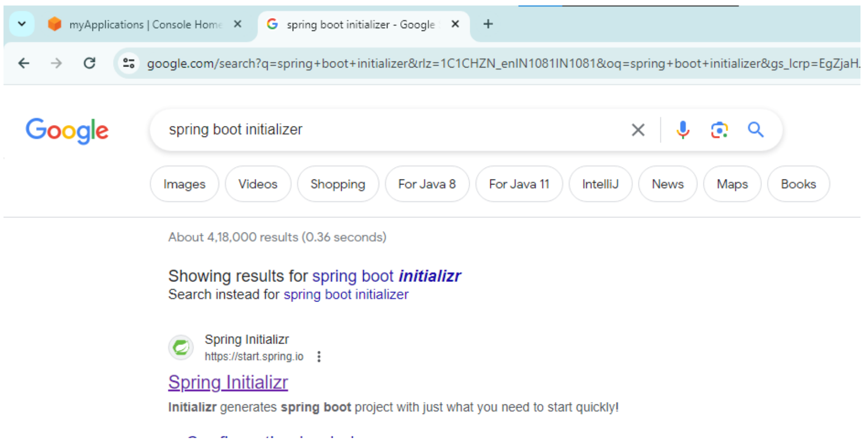


localhost:3300/div/2/3

```
{"result":0.6666666666666666}
```

PROGRAM:

OPEN GOOGLE CROME >search spring boot initializer


myApplications | Console Home ×   spring boot initializer - Google ×   +

google.com/search?q=spring+boot+initializer&rlz=1C1CHZN_enIN1081IN1081&oq=spring+boot+initializer&gs_lcrp=EgZjaH

spring boot initializer

Images   Videos   Shopping   For Java 8   For Java 11   IntelliJ   News   Maps   Books

About 4,18,000 results (0.36 seconds)

Showing results for spring boot ***initializr***
Search instead for spring boot initializer

Spring Initializr
https://start.spring.io

Spring Initializr

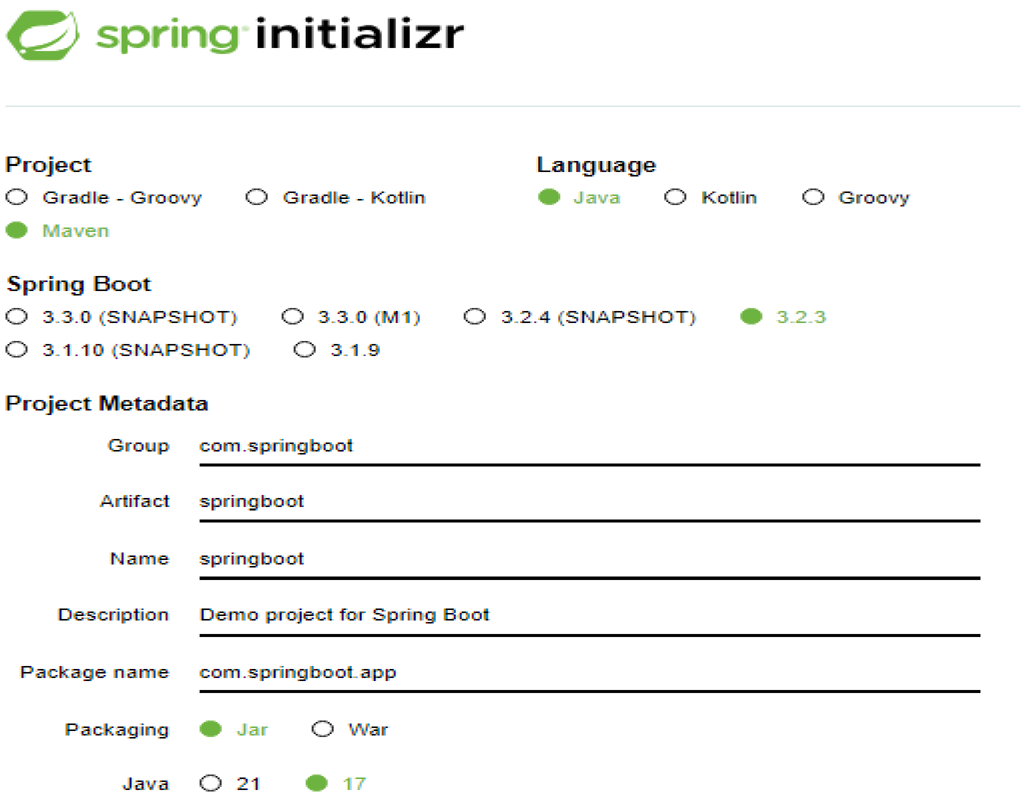
Initializr generates **spring boot** project with just what you need to start quickly!

SELECT>PROJECT:MAVEN

LANGUAGE:JAVA

SPRING BOOT :3.2.3

FILL UP PROJECT METADATA


spring initializr

Project
Gradle - Groovy   Gradle - Kotlin   [✓ Maven]

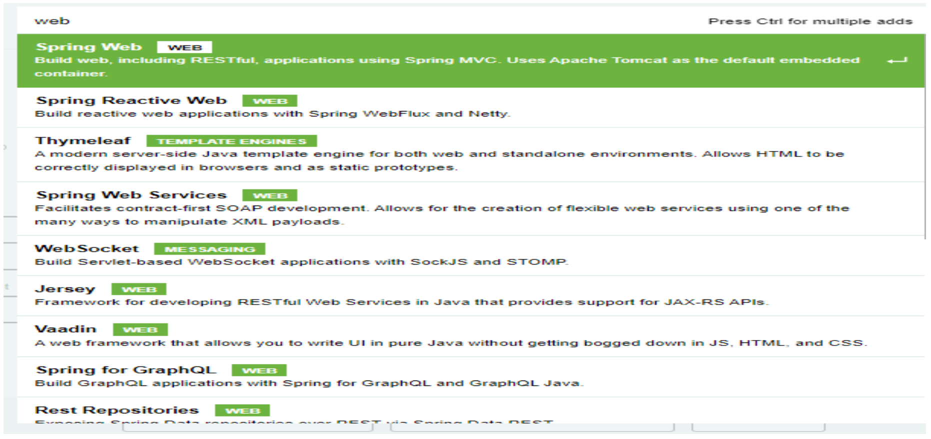
Language
[✓ Java]   Kotlin   Groovy

Spring Boot
3.3.0 (SNAPSHOT)   3.3.0 (M1)   3.2.4 (SNAPSHOT)   [✓ 3.2.3]
3.1.10 (SNAPSHOT)   3.1.9

Project Metadata

Group	com.springboot
Artifact	springboot
Name	springboot
Description	Demo project for Spring Boot
Package name	com.springboot.app
Packaging	[✓ Jar] War
Java	21 [✓ 17]

GO TO DEPENDENCIES >SEARCH WEB



web — Press Ctrl for multiple adds

Spring Web WEB
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

Spring Reactive Web WEB
Build reactive web applications with Spring WebFlux and Netty.

Thymeleaf TEMPLATE ENGINES
A modern server-side Java template engine for both web and standalone environments. Allows HTML to be correctly displayed in browsers and as static prototypes.

Spring Web Services WEB
Facilitates contract-first SOAP development. Allows for the creation of flexible web services using one of the many ways to manipulate XML payloads.

WebSocket MESSAGING
Build Servlet-based WebSocket applications with SockJS and STOMP.

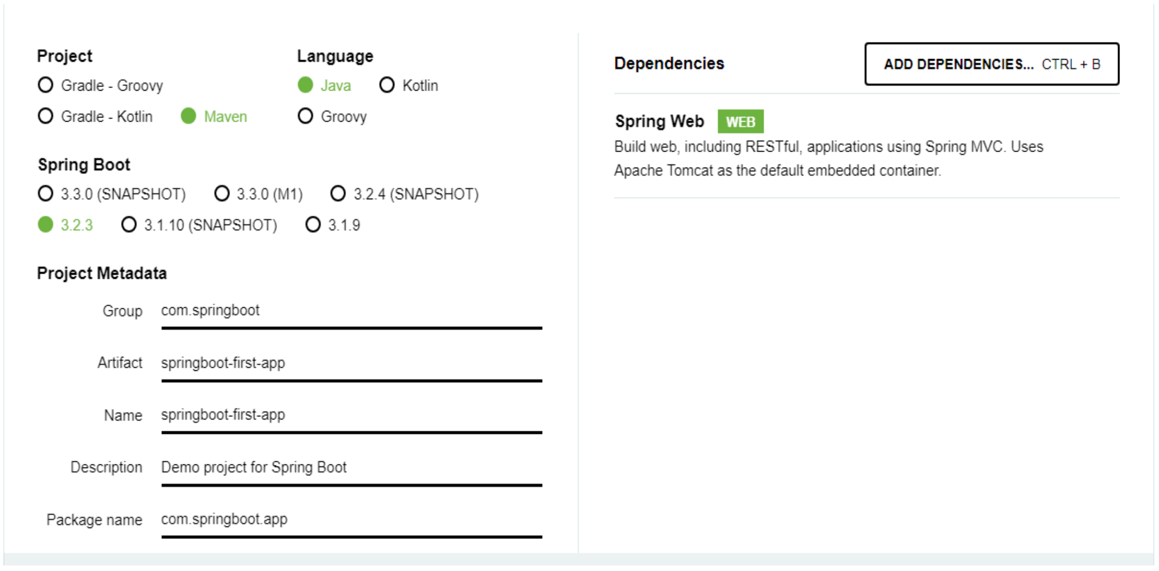
Jersey WEB
Framework for developing RESTful Web Services in Java that provides support for JAX-RS APIs.

Vaadin WEB
A web framework that allows you to write UI in pure Java without getting bogged down in JS, HTML, and CSS.

Spring for GraphQL WEB
Build GraphQL applications with Spring for GraphQL and GraphQL Java.

Rest Repositories WEB

ADD DEPENDENCIES



Project
Gradle - Groovy
Gradle - Kotlin
[✓ Maven]

Language
[✓ Java]
Kotlin
Groovy

Spring Boot
3.3.0 (SNAPSHOT)
3.3.0 (M1)
3.2.4 (SNAPSHOT)
[✓ 3.2.3]
3.1.10 (SNAPSHOT)
3.1.9

Project Metadata

Group	com.springboot
Artifact	springboot-first-app
Name	springboot-first-app
Description	Demo project for Spring Boot
Package name	com.springboot.app

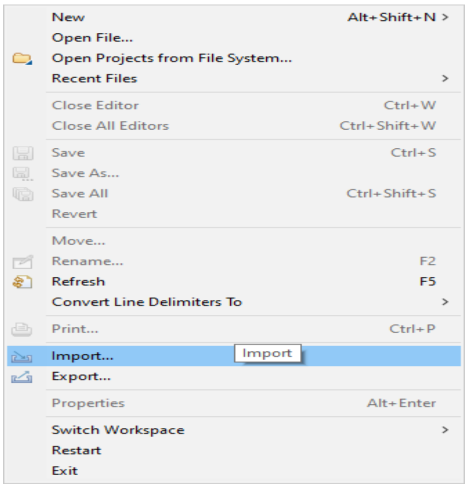
Dependencies — ADD DEPENDENCIES... CTRL + B

Spring Web WEB
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

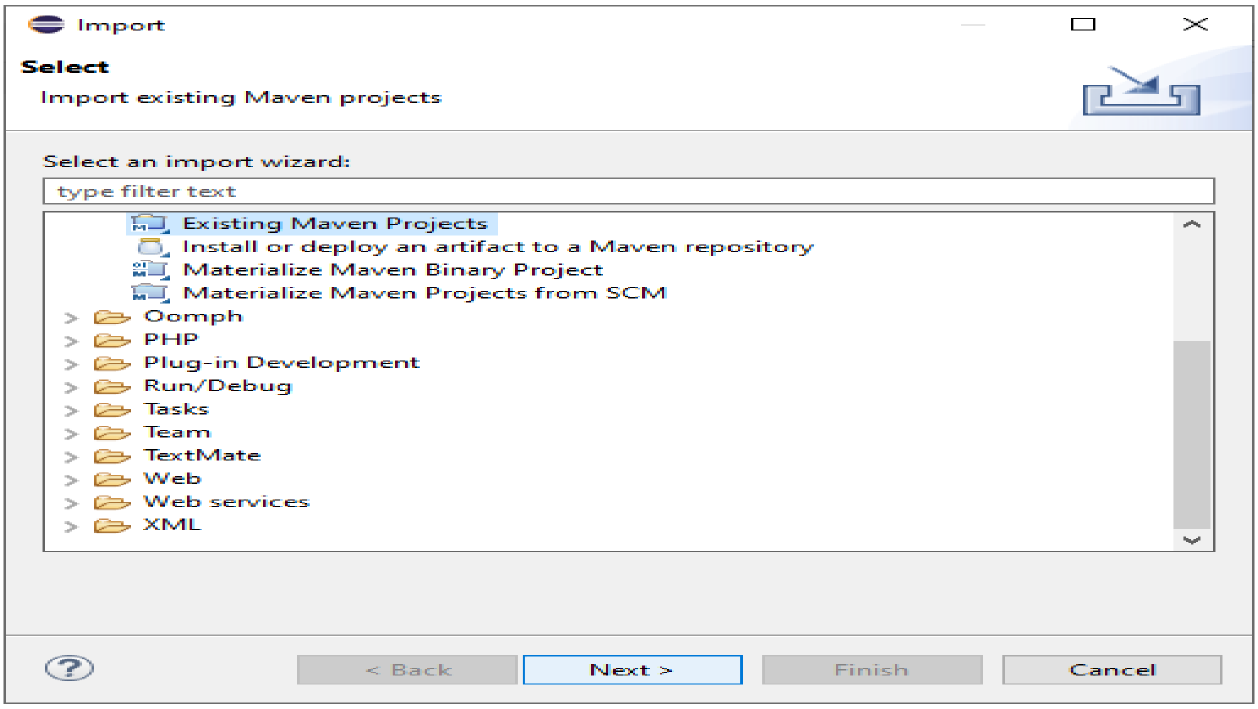
CLICK ON GENERATE BUTTON (BELOW) TO DOWNLOAD SPRING BOOT PROJECT AS ZIP FILE

EXTRACT THE ZIP FILE

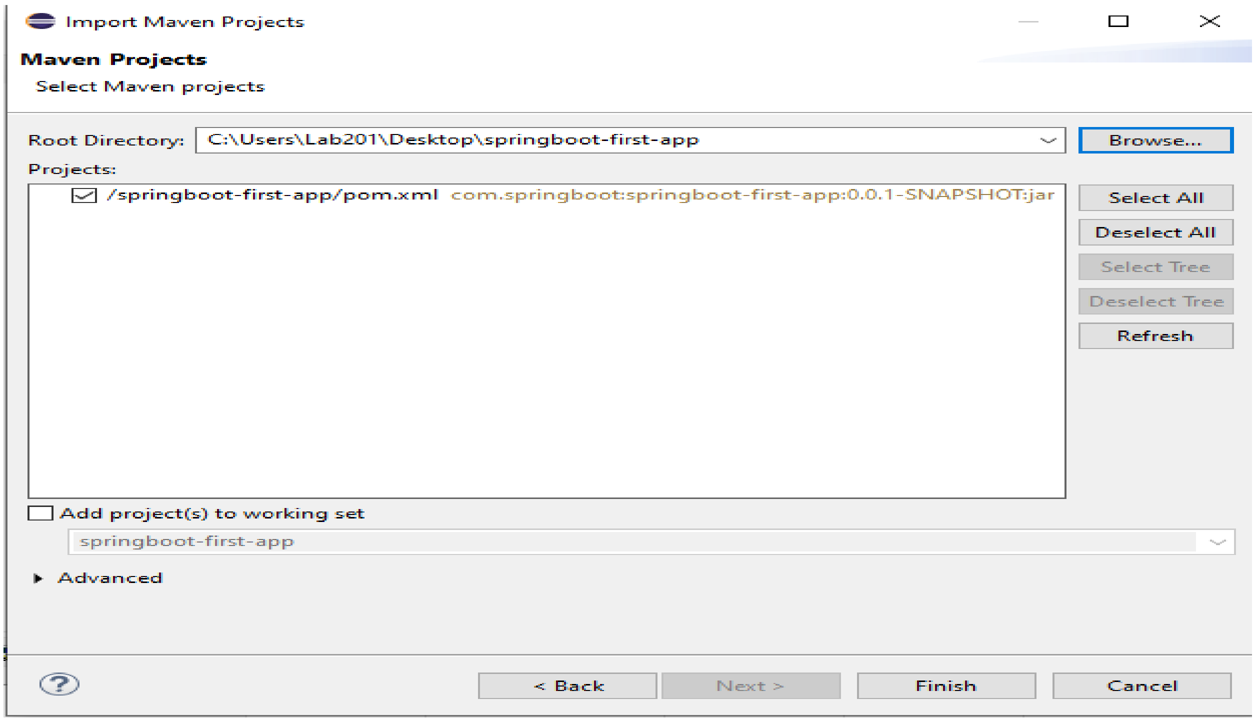
OPEN ECLIPSE>GO TO FILE>IMPORT



SELECT EXISTING MAVEN PROJECT



BROWSE SPRINGBOOT-FIRAT-APP >CLICK ON FINISH


Import Maven Projects

Maven Projects

Select Maven projects

Root Directory: C:\Users\Lab201\Desktop\springboot-first-app

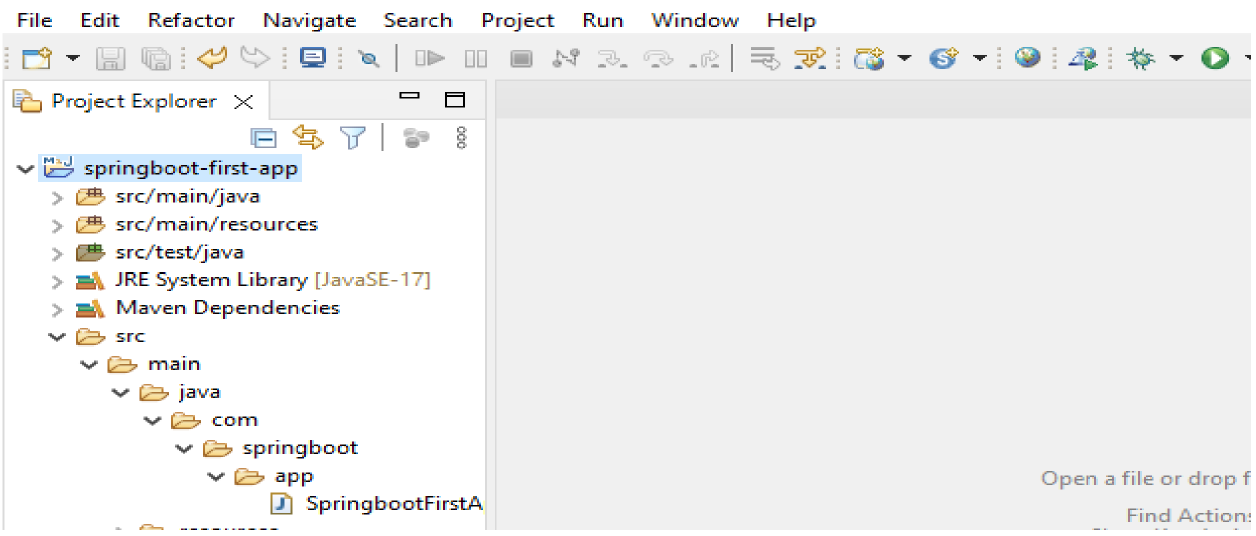
Projects:

☑ /springboot-first-app/pom.xml com.springboot:springboot-first-app:0.0.1-SNAPSHOT:jar

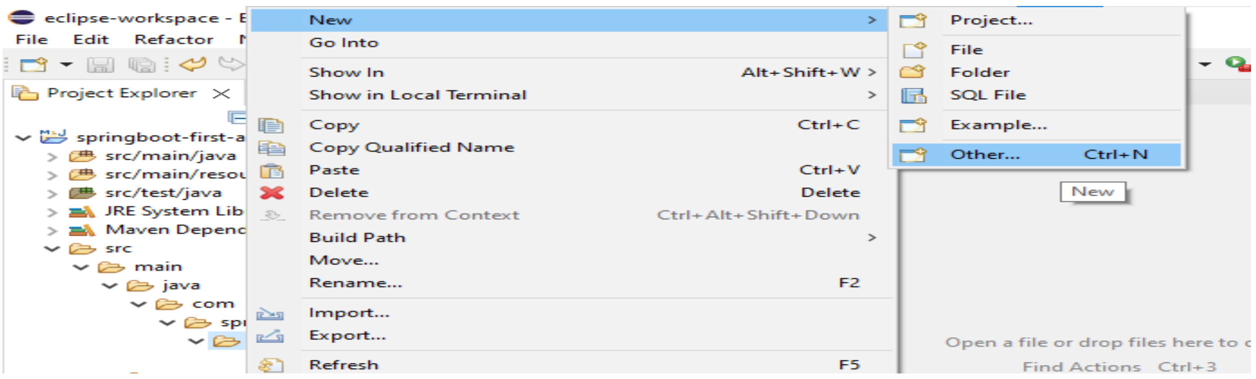
☐ Add project(s) to working set

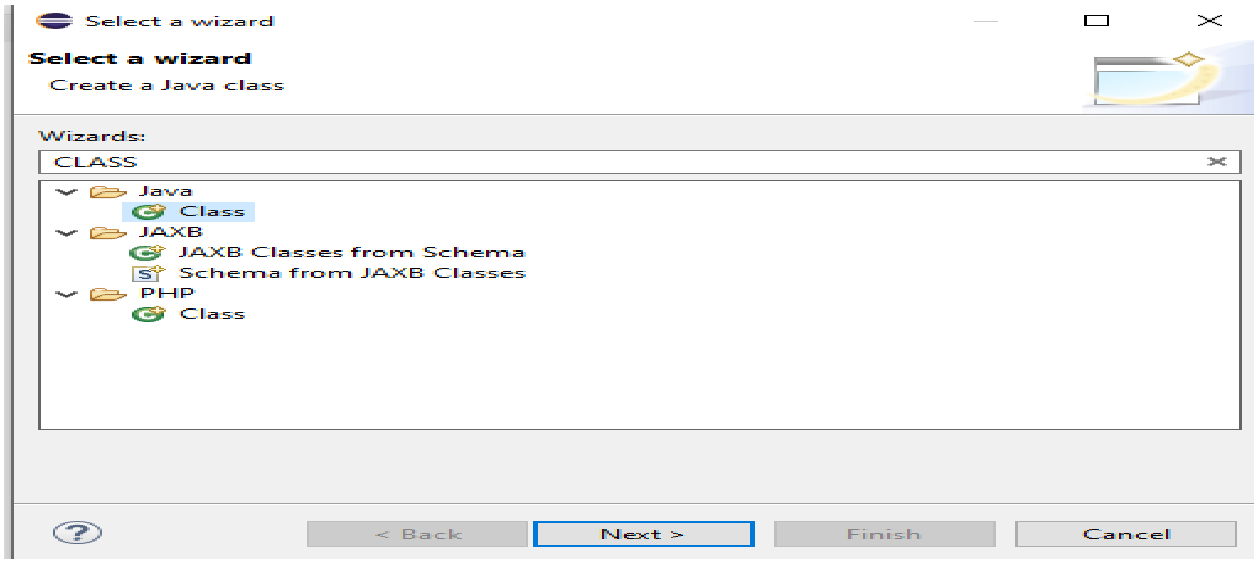
springboot-first-app

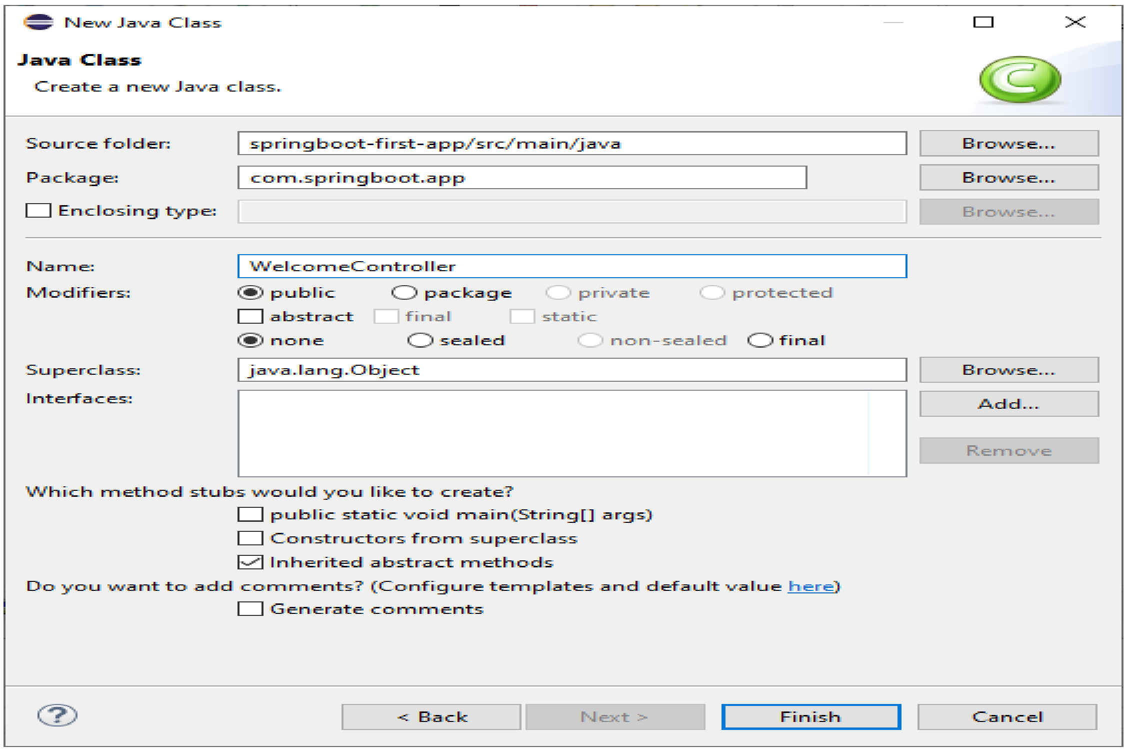
SELECT SPRINGBOOT-FIRST-APP>SRC>MAIN>JAVA>COM>SPRINGBOOT>APP



RIGHT CLICK ON APP>NEW>OTHERS>SELECT CLASS>NEXT






New Java Class

Java Class

Create a new Java class.

Source folder: springboot-first-app/src/main/java

Package: com.springboot.app

☐ Enclosing type:

Name: WelcomeController

Modifiers: [✓ public] package private protected

☐ abstract ☐ final ☐ static

[✓ none] sealed non-sealed final

Superclass: java.lang.Object

Interfaces:

Which method stubs would you like to create?

☐ public static void main(String[] args)

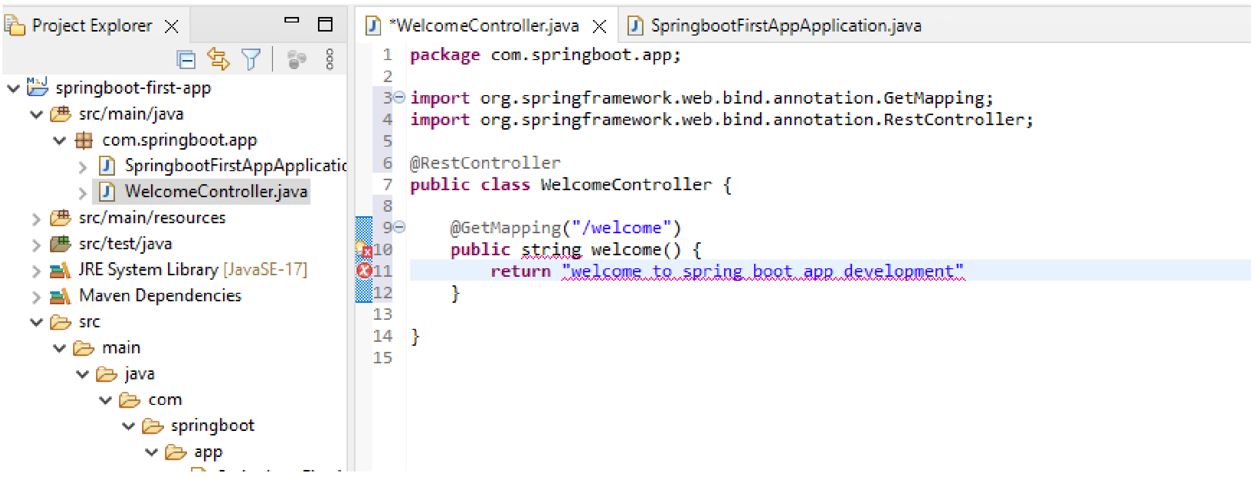
☐ Constructors from superclass

☑ Inherited abstract methods

Do you want to add comments? (Configure templates and default value here)

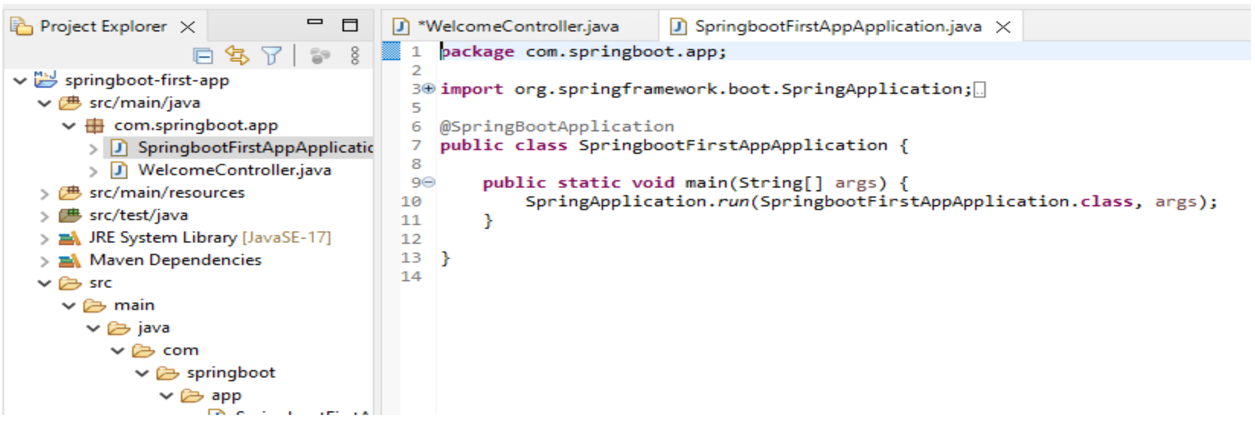
☐ Generate comments

WRITE CODE IN WELCOMECONTROLLER.JAVA FILE



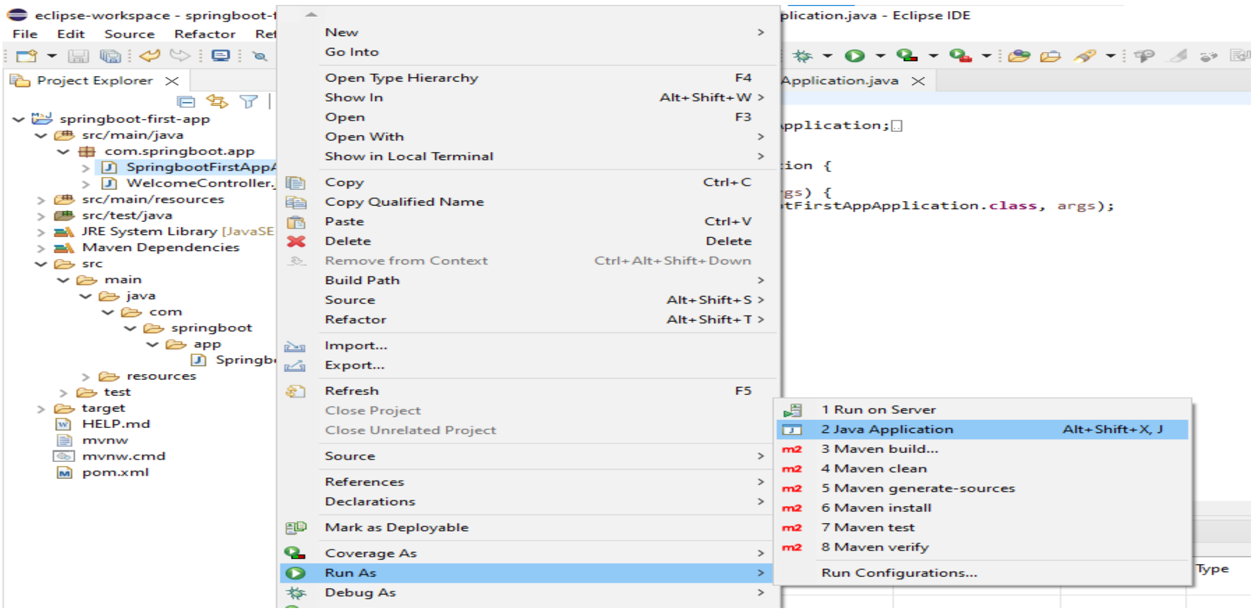
```
package com.springboot.app;

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

@RestController
public class WelcomeController {

    @GetMapping("/welcome")
    public string welcome() {
        return "welcome to spring boot app development"
    }

}
```

NOW GO TO springbootfirstappapplication.java



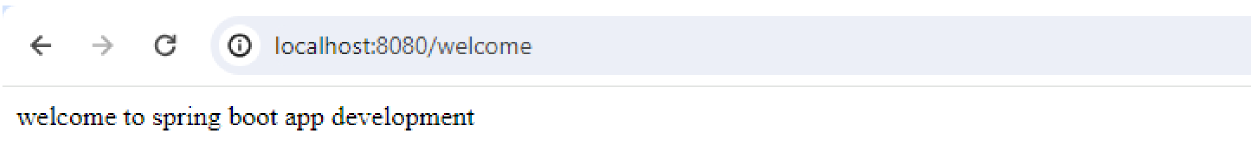
```
package com.springboot.app;

import org.springframework.boot.SpringApplication;

@SpringBootApplication
public class SpringbootFirstAppApplication {

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

}
```

RUN THE PROGRAM (AS JAVA APPLICATION)



GO TO BROWSER TYPE>localhost:8080/welcome

<u>OUTPUT:</u>



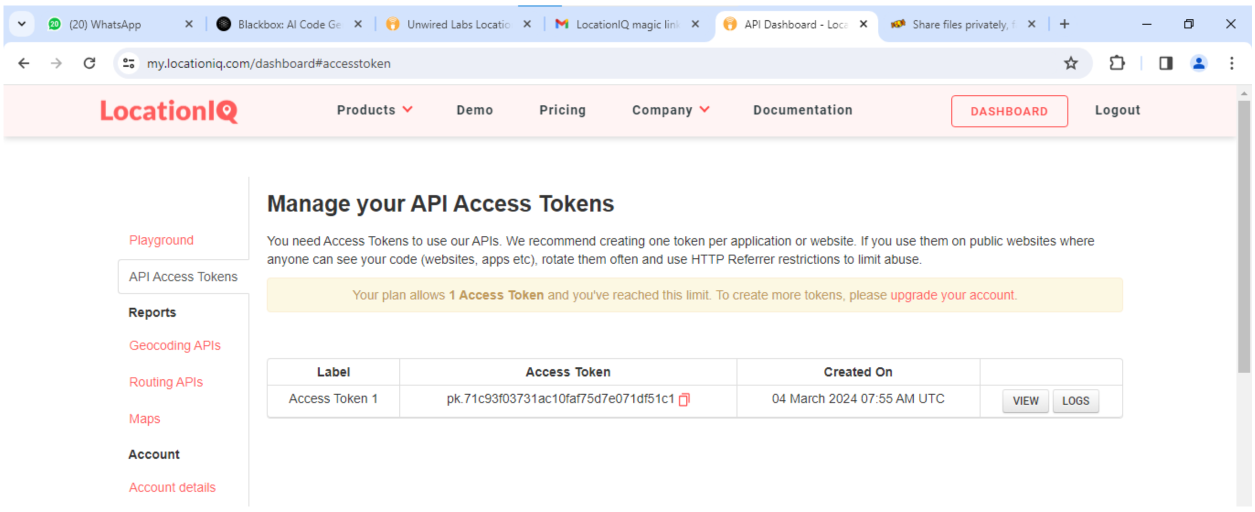
welcome to spring boot app development

<u>PROGRAM:</u> **(GOOGLE's MAP RESTFUL WEB SERVICE)**

```
import requests

def get_geolocation(api_key, search_string):
    base_url = "https://us1.locationiq.com/v1/search"
    params = {
        'key': api_key,
        'q': search_string,
        'format': 'json',
    }

    response = requests.get(base_url, params=params)
    data = response.json()

    if response.status_code == 200 and data:
        result = {
            'place_id': data[0].get('place_id', ''),
            'lat': data[0].get('lat', ''),
            'lon': data[0].get('lon', ''),
            'display_name': data[0].get('display_name', ''),
        }
        return result
    else:
        print(f"Error: {response.status_code} - {data.get('error', 'No error message')}")
        return None

api_key = 'pk.71c93f03731ac10faf75d7e071df51c1 '
search_string = input("Enter the location : ")
```

```
result = get_geolocation(api_key, search_string)


if result:
    print("Output:")
    for key, value in result.items():
        print(f"{key}: {value}")
```

BROWSE LOCATION IQ>SIGNUP WITH EMAIL> CLICK ON THE LINK PROVIDED BY LOCATION IQ (on your email)>YOU WILL GET YOUR ACCESS TOKEN COPY THE KEY AND PASTE IN THE PYTHON CODE:



Manage your API Access Tokens

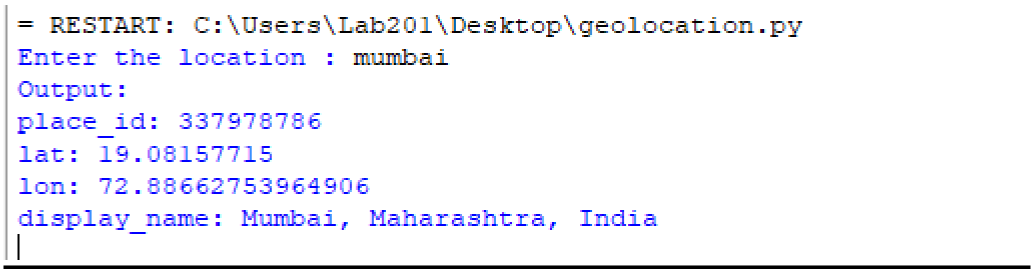
You need Access Tokens to use our APIs. We recommend creating one token per application or website. If you use them on public websites where anyone can see your code (websites, apps etc), rotate them often and use HTTP Referrer restrictions to limit abuse.

Your plan allows 1 Access Token and you've reached this limit. To create more tokens, please upgrade your account.

Label	Access Token	Created On	
Access Token 1	pk.71c93f03731ac10faf75d7e071df51c1	04 March 2024 07:55 AM UTC	VIEW LOGS

RUN THE CODE

Output:



```
= RESTART: C:\Users\Lab201\Desktop\geolocation.py
Enter the location : mumbai
Output:
place_id: 337978786
lat: 19.08157715
lon: 72.88662753964906
display_name: Mumbai, Maharashtra, India
```

PROGRAM: **(INSTALLATION AND CONFIGURATION OF VIRTUALIZATION USING KVM)**

COMMANDS:

1.sudo grep-c"svm\|vmx"/proc/cpuinfo

2.sudo apt install qemu-kvm libvirt-daemon-system virt-manager brid

3.sudo apt-get update

4.sudo apt-get install qemu-kvm libvirt-daemon-system virt-manager bridge-utils

5.sudo apt install qemu-kvm libvirt-clients libvirt-daemon-system bridge-utils

6.sudo systemctl start libvirtd

7.sudo usermod -aG kvm $USER

8.sudo systemctl is-active libvirtd

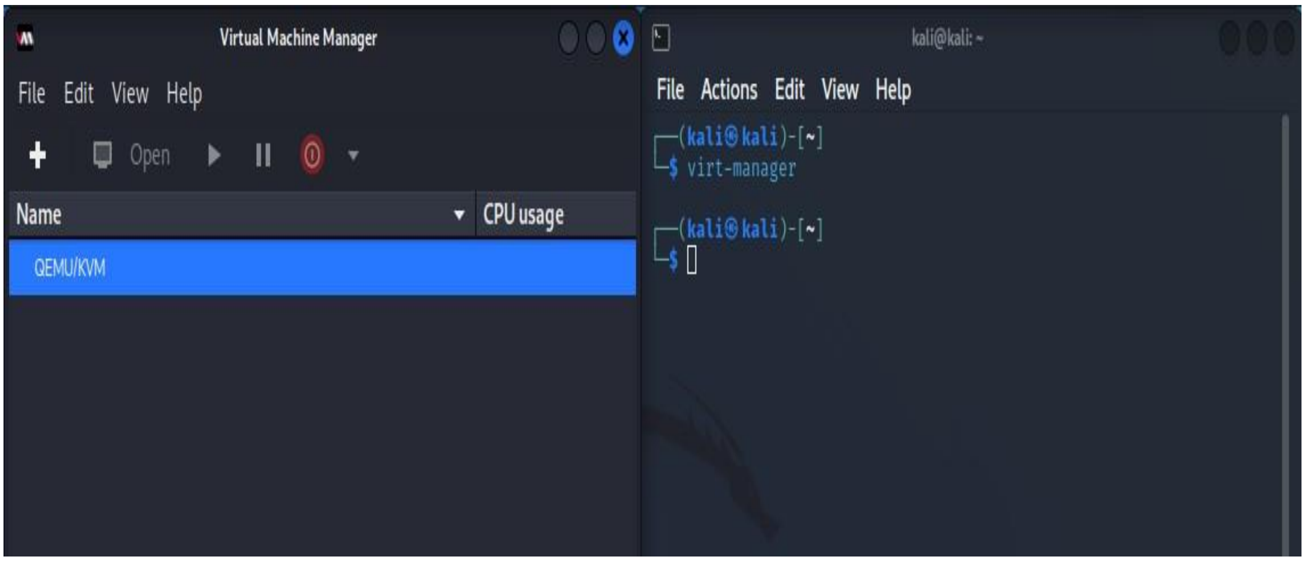
9.sudo usermod -aG libvirt $USER

sudo usermod -aG kvm $USER

10.virt-manager

11.kvm-ok

OUTPUT:



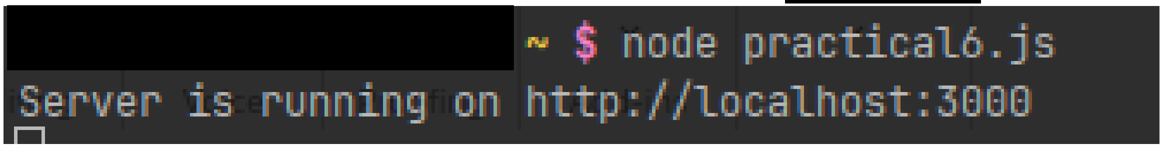
```
┌──(kali㉿kali)-[~]
└─$ virt-manager

┌──(kali㉿kali)-[~]
└─$ ▯
```

PROGRAM: (application to download image/video from server or upload image/video to server using MTOM techniques)

(node.js) code:

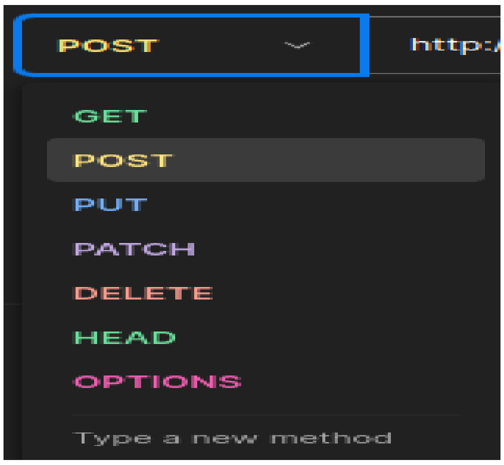
```
const express = require('express');

const multer = require('multer');

const path = require('path');

const fs = require('fs');

const app = express();

const port = 3000;

// Define storage using multer.diskStorage

const storage = multer.diskStorage({

  destination: (req, file, cb) => {

    // Set the destination folder where the file will be saved

    const uploadFolder = 'uploads';

    fs.mkdirSync(uploadFolder, { recursive: true });

    cb(null, uploadFolder);

  },

  filename: (req, file, cb) => {

    // Set the filename to the original filename

    cb(null, file.originalname);

  },

});


const upload = multer({ storage: storage });

app.post('/upload', upload.single('file'), (req, res) => {

  const file = req.file;
  // Check if file is present
  if (!file) {
    return res.status(400).json({ success: false, message: 'No file uploaded.' });
  }
  // Process the file as needed (save to disk, database, etc.)
  res.json({ success: true, message: 'File uploaded successfully.' });
});
app.get('/download/:filename', (req, res) => {
  const filename = req.params.filename;
  const filePath = path.join(__dirname, 'uploads', filename);
  // Check if file exists
  if (fs.existsSync(filePath)) {
    // Implement logic to send the file as a response
    res.sendFile(filePath);
  } else {
    res.status(404).json({ success: false, message: 'File not found.' });
  }
});
app.listen(port, () => {
  console.log(Server is running on http://localhost:${port});
});
```

RUN THE CODE: IT WILL START THE SERVER AT <u>localhost:3000</u>



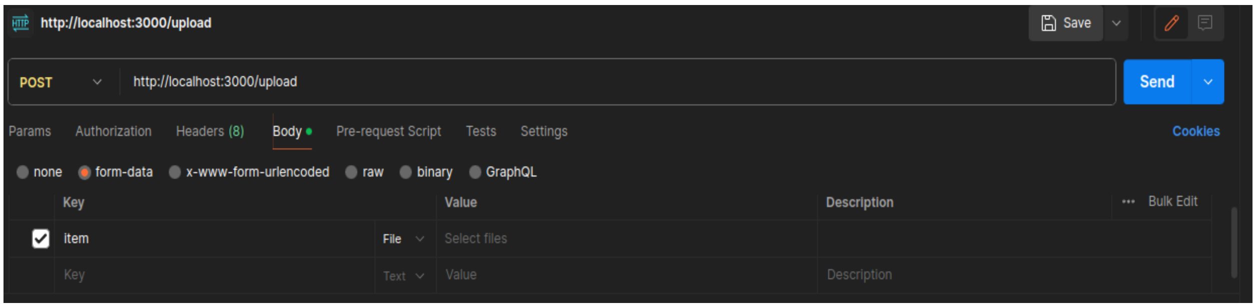
```
[redacted] ~ $ node practical6.js
Server is running on http://localhost:3000
```

OPEN POSTMAN OPEN A NEW PAGE & CHOOSE POST METHOD



ENTER THE URL OF THE SERVER: http://localhost:3000/upload

THEN CLICK ON BODY>FORM DATA>NAME THE KEY ITEM AND FILE TYPE: FILE



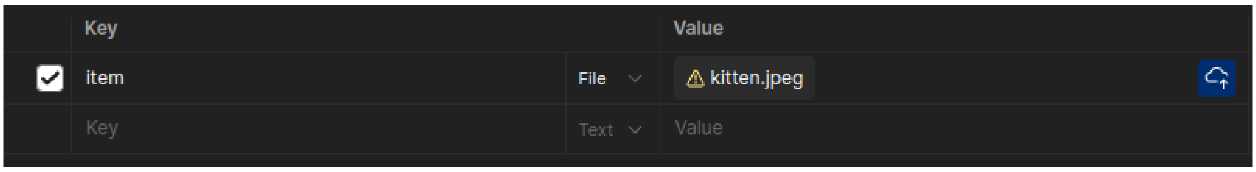
http://localhost:3000/upload

POST http://localhost:3000/upload

Body: form-data

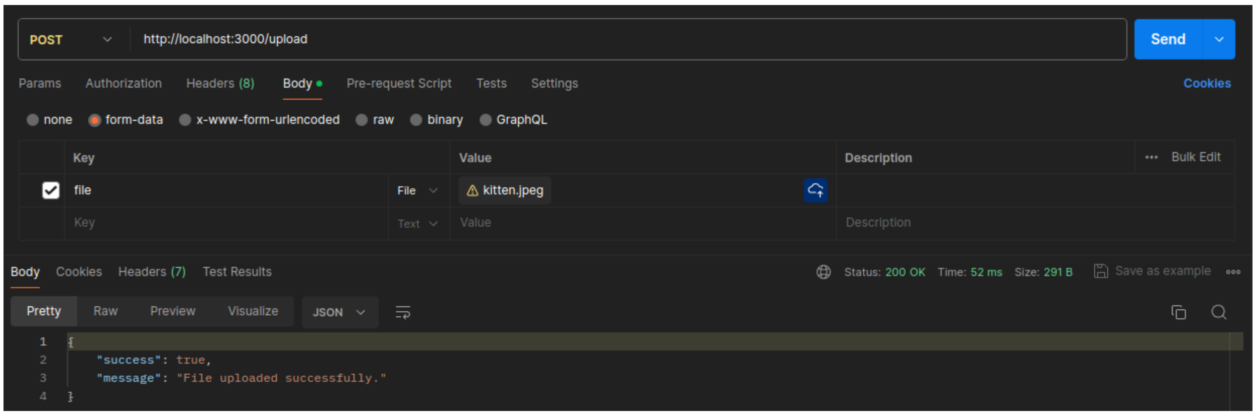
Key		Value	Description
item	File	Select files	
Key	Text	Value	Description

IN VALUE TAB ENTER THE FILE YOU WANT TO UPLOAD TO THE SERVER



Key		Value
item	File	kitten.jpeg
Key	Text	Value

CLICK ON SEND



POST http://localhost:3000/upload

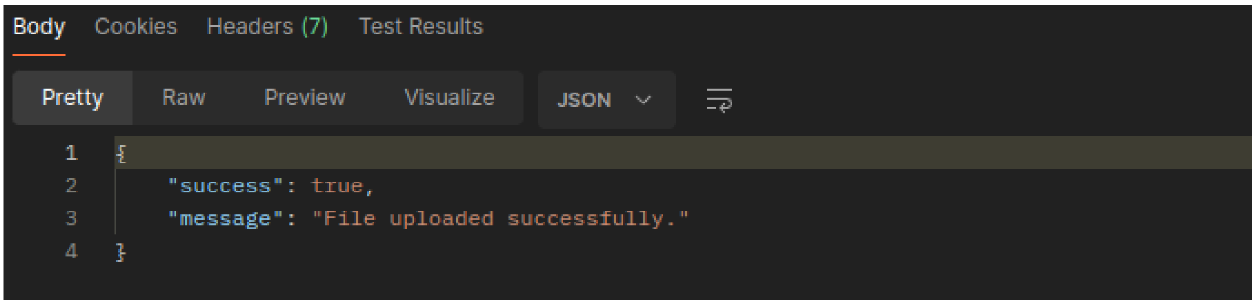
Body: form-data

Key		Value	Description
file	File	kitten.jpeg	
Key	Text	Value	Description

Status: 200 OK Time: 52 ms Size: 291 B

```
{
    "success": true,
    "message": "File uploaded successfully."
}
```

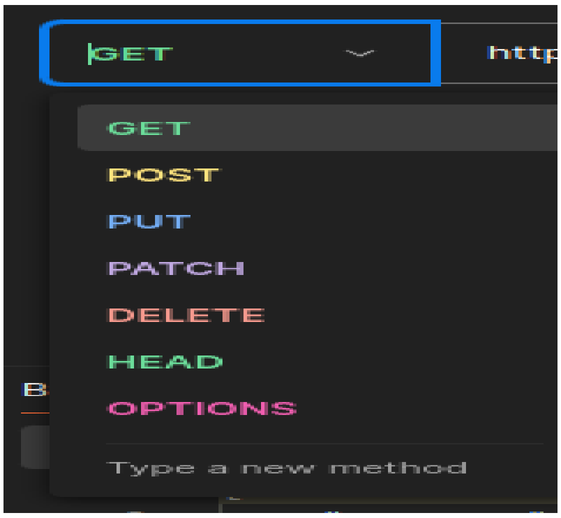
Upload Output:



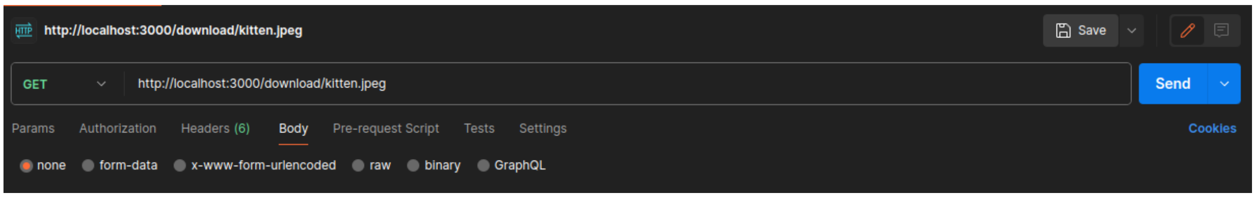
```
{
    "success": true,
    "message": "File uploaded successfully."
}
```

DOWNLOAD:

CHOOSE GET METHOD



ENTER THE URL OF THE SERVER: http://localhost:3000/download/kitten.jpg

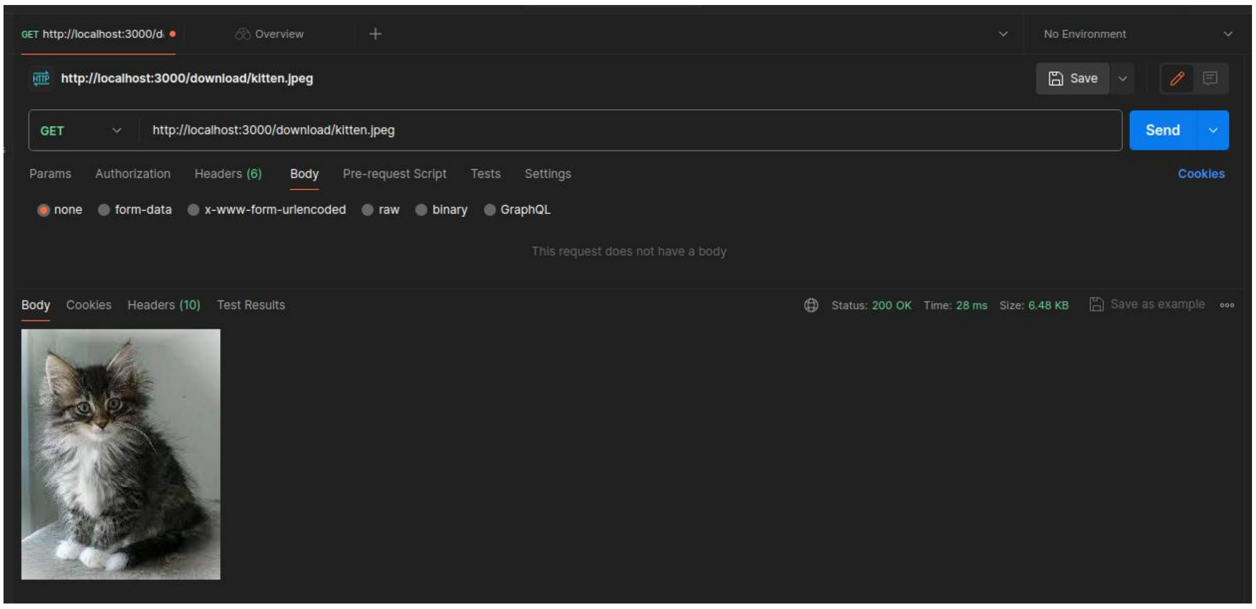


http://localhost:3000/download/kitten.jpeg

GET http://localhost:3000/download/kitten.jpeg

CLICK ON SUBMIT

DOWNLOAD OUTPUT :

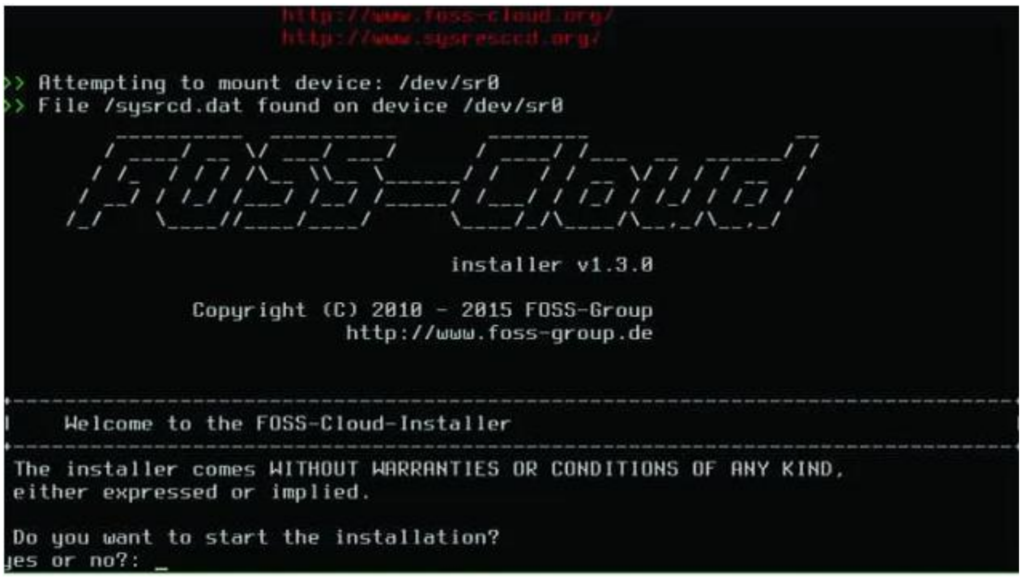


GET http://localhost:3000/download/kitten.jpeg

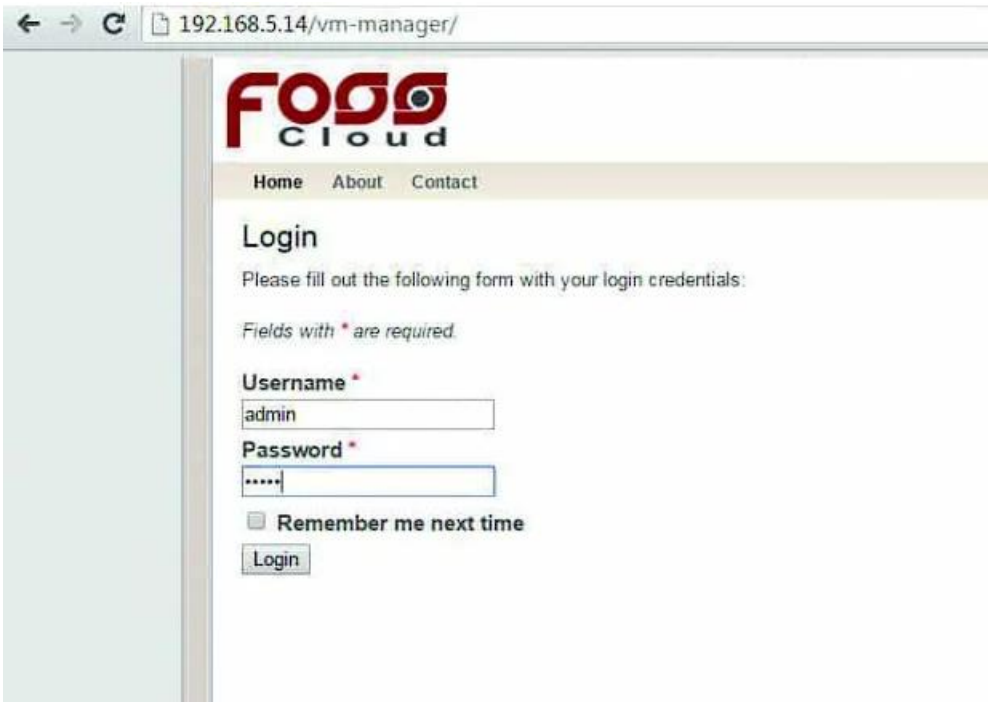
Status: 200 OK Time: 28 ms Size: 6.48 KB

PROGRAM: **(Cloud Functionality VSI (Virtual Server Infrastructure) Infrastructure as a Service (IaaS), Storage)**

AFTER INSTALLATION, IT WILL SHOW YOU AN IP ADDRESS. PUT IT IM YOUR BROWSER TO ACCESS YOUR ADMINISTRATOR PAGE THE DEFAULT USER CREDENTIALS ARE USER: ADMIN AND PASSWORD: ADIMIN. FOR ROOT LOGIN - USERNAME- ROOT AND PASSWORD - PASSWORD.



```
                    http://www.foss-cloud.org/
                    http://www.sysresccd.org/

>> Attempting to mount device: /dev/sr0
>> File /sysrcd.dat found on device /dev/sr0

                          installer v1.3.0

          Copyright (C) 2010 - 2015 FOSS-Group
                    http://www.foss-group.de

|    Welcome to the FOSS-Cloud-Installer
 The installer comes WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
 either expressed or implied.

 Do you want to start the installation?
yes or no?: _
```



192.168.5.14/vm-manager/

Home About Contact

Login

Please fill out the following form with your login credentials:

*Fields with * are required.*

Username *
admin

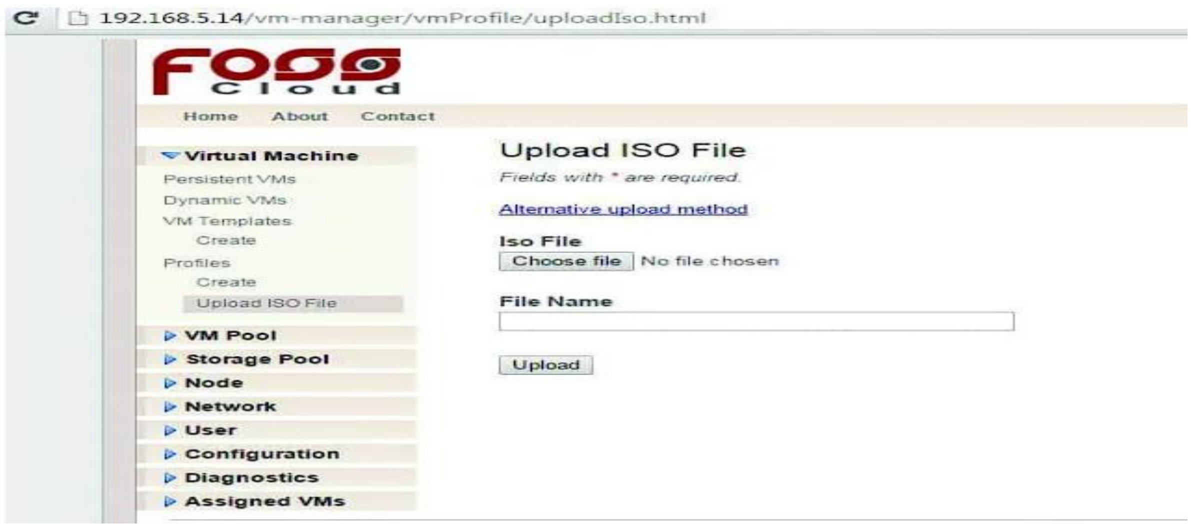
Password *
•••••

Remember me next time

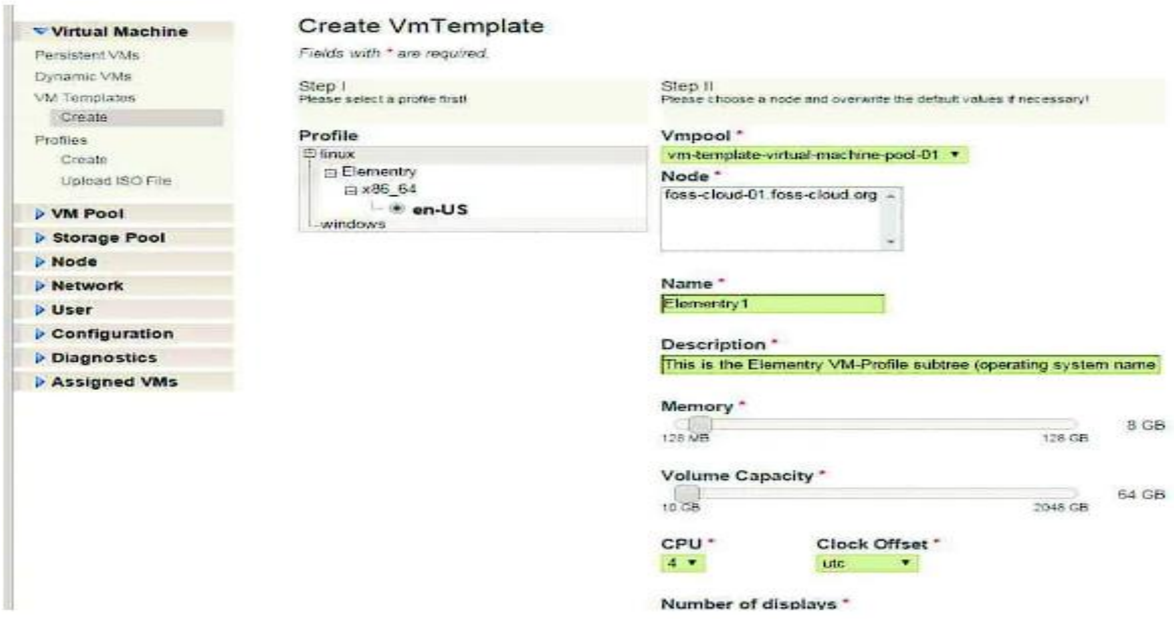
Login

THE FIRST SCREEN AFTER LOGIN SHOWS MANY OPTIONS TO INSTALL AND DEPLOY ANY VIRTUAL MACHINE. TO INSTALL A VIRTUAL MACHINE CLICK ON VIRTUAI MACHINE-> UPLOAD ISO FILE OPTION AND UPLOAD

THE BOOTABLE ISO FILE. HERE, WE ARE GOING TO UPLOAD LINUX ELEMENTARY 0S IS0.



ONCE YOU UPLOADED THE FILE, CREATE VIMTEMPLATE. IN THIS OPTION YOU ARE BASICALLY CONFIGURING YOUR VIRTUAL MACHINE'S STORAGE LOCATION, CPU, MEMORY, NODE ETC. HERE, YOU WILL FIND SINGLE NODES ANA VW PO01 1 RESPECTIVE OPTIONS BECAUSE EVERYTHING WAS INSTALLED AT THE SINGLE SERVER.


Create VmTemplate

Fields with * are required.

Step I
Please select a profile first!

Step II
Please choose a node and overwrite the default values if necessary!

Profile: linux / Elementry / x86_64 / en-US; windows

Vmpool *: vm-template-virtual-machine-pool-01

Node *: foss-cloud-01.foss-cloud.org

Name *: Elementry1

Description *: This is the Elementry VM-Profile subtree (operating system name

Memory *: 8 GB (128 MB – 128 GB)

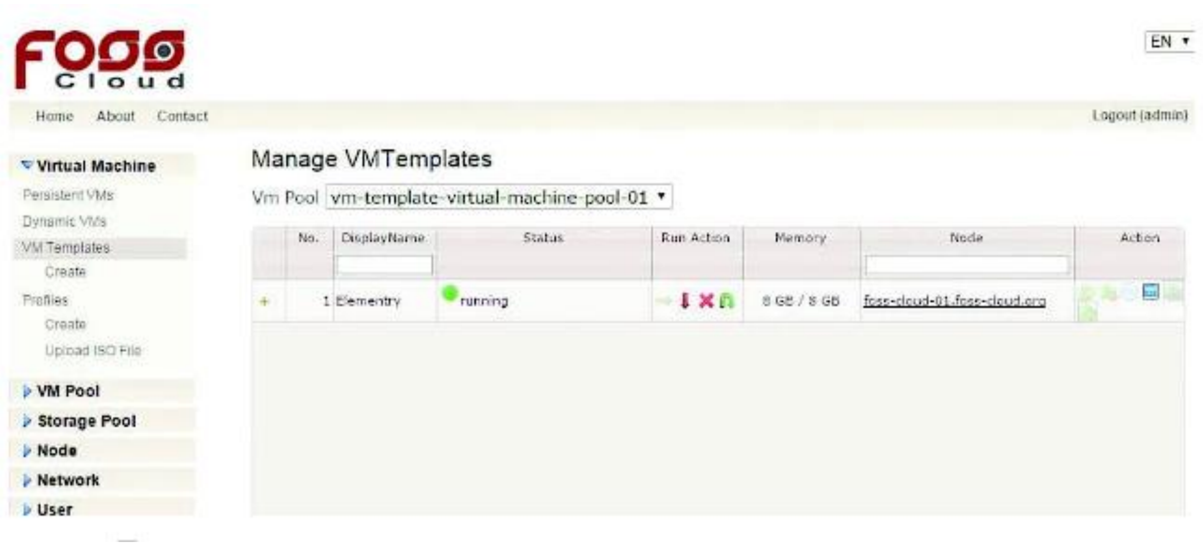
Volume Capacity *: 64 GB (10 GB – 2048 GB)

CPU *: 4

Clock Offset *: utc

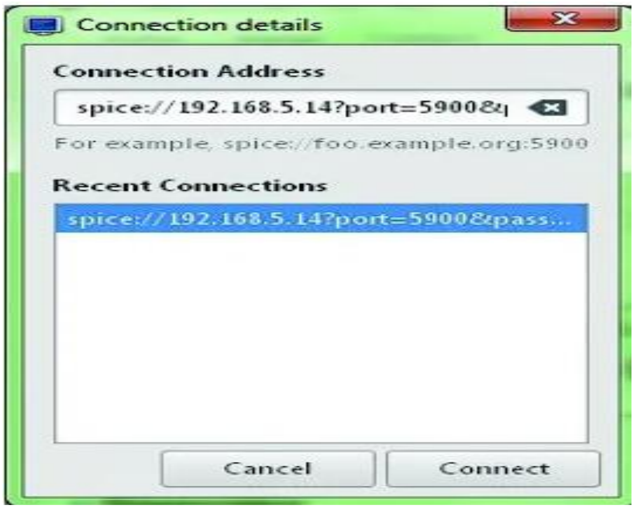
Number of displays *

NOW, CLICK ON VMTEMPLATES AND YOU WILL SEE A TEMPLATE WHICH YOU HAVE CREATED IN STEP 4. TO START YOUR MACHINE GO TO RUN ACTION TAB AND CLICK ON THE GREEN ARROW. UNDER STATUS TAB, IT SHOWS THE RUNNING TEXT WITH THE GREEN CIRCLE WHICH SHOWS THAT YOUR MACHINE IS RUNNING WITHOUT ANY ERRORS. TO VIEW YOUR VIRTUAL MACHINE CLICK ON A BLUE SQUARE BOX UNDER ACTION TAB.

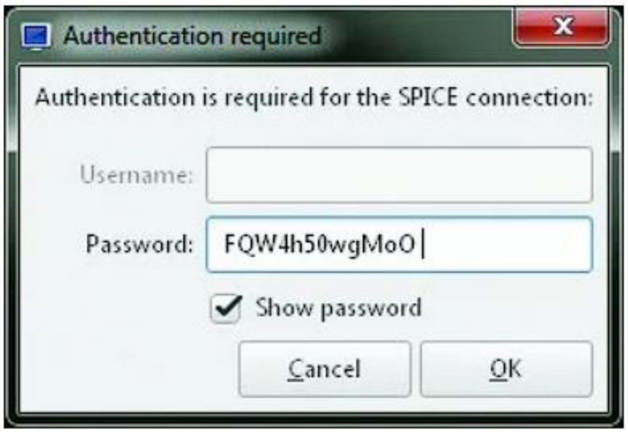


TO VIEW YOUR VIRTUAL MACHINE YOU HAVE TO DOWNLOAD SPICE CLIENT TOOL. THE DOWNLOAD LINK CAN BE FOUND UNDER THE LINKS OPTION. AFTER DOWNLOADING THE APPLICATION, CLICK ON THE BLUE SQUARE TO VIEW MACHINE. WHEN YOU CLICK ON IT, THE BROWSER WILL POP-UP FOR LAUNCHING THE APPLICATION.

IF THE APPLICATION DOES NOT LAUNCH AUTOMATICALLY, LAUNCH IT MANUALLY BY ENTERING THE LINK AND PASSWORD IN THE REMOTE VIEWER TOOL WHICH YOU WILL SEE IN THE POP-UP MESSAGE.


Connection details
Connection Address
spice://192.168.5.14?port=5900&
For example, spice://foo.example.org:5900
Recent Connections
spice://192.168.5.14?port=5900&pass...
Cancel
Connect

ONCE IT CONNECTS, ENTER THE PASSWORD WHICH WAS GIVEN IN THE LINK AND CLICK OK.



Authentication required

Authentication is required for the SPICE connection:

Username:

Password: FQW4h50wgMoO

[✓] Show password

FINALLY YOU WILL ABLE TO VIEW AND CONTROL YOUR VIRTUAL MACHINE.