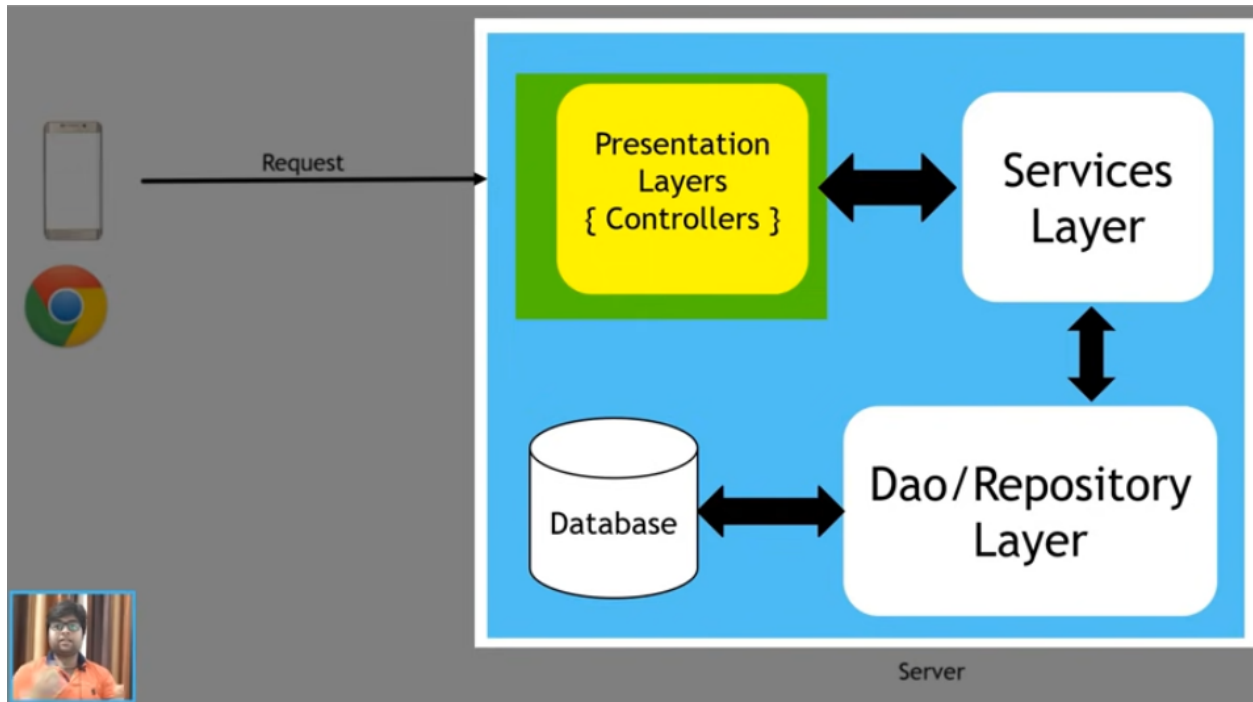


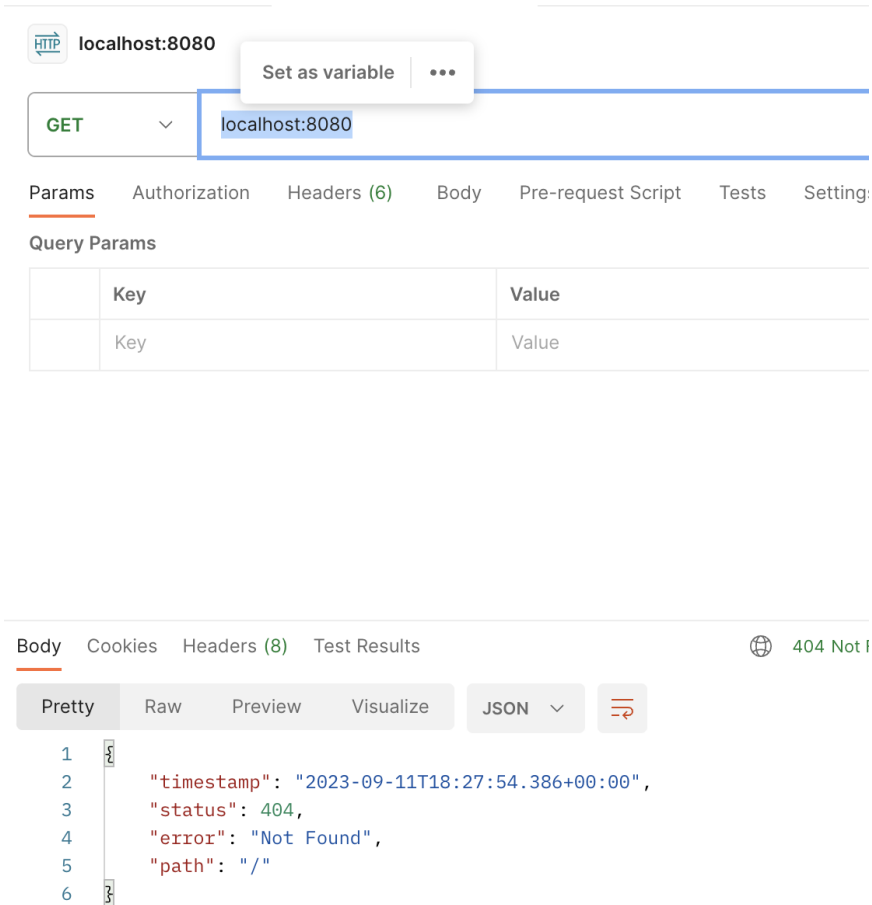
API

j2ee arch-



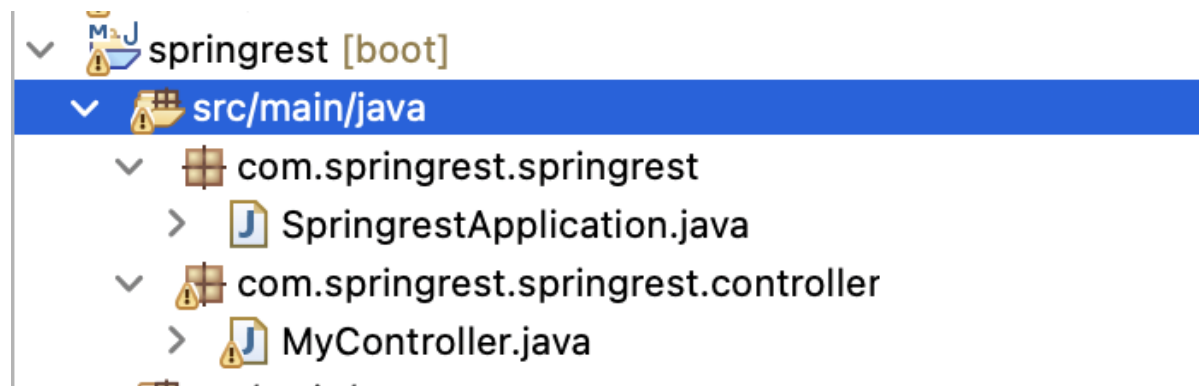
rest api using spboot

- create a spring boot project online using spring intitalizer online put the foll dependencies-
- Spring Web
- Spring Data JPA
- Spring my sql
- then open this project in ide of your choice (eclipse or intellij)
- when we run this project as spring boot project by default it runs on port 8080
- when we enter localhost:8080 on postman for get we get



since there is nothing that we wrote in our spring boot api

** now we make controllers our gate keepers in a new package




- write the following code-

```
SpringrestApplication.java  springrest/pom.xml  *MyController.java X
1 package com.springrest.springrest.controller;
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.web.bind.annotation.GetMapping;
5 import org.springframework.web.bind.annotation.RestController;
6
7 @RestController
8 public class MyController {
9     @GetMapping("/home")
10     public String getname() {
11         return "pravar";
12     }
13
14
15 }
16
```

o

when we run our app as a spring boot app and then type the following in postman

 localhost:8080/home

GET

▼

localhost:8080/home

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

5

Query Params

	Key	Value
	Key	Value

Body

Cookies

Headers (5)

Test Results


Pretty

Raw

Preview

Visualize

Text ▼



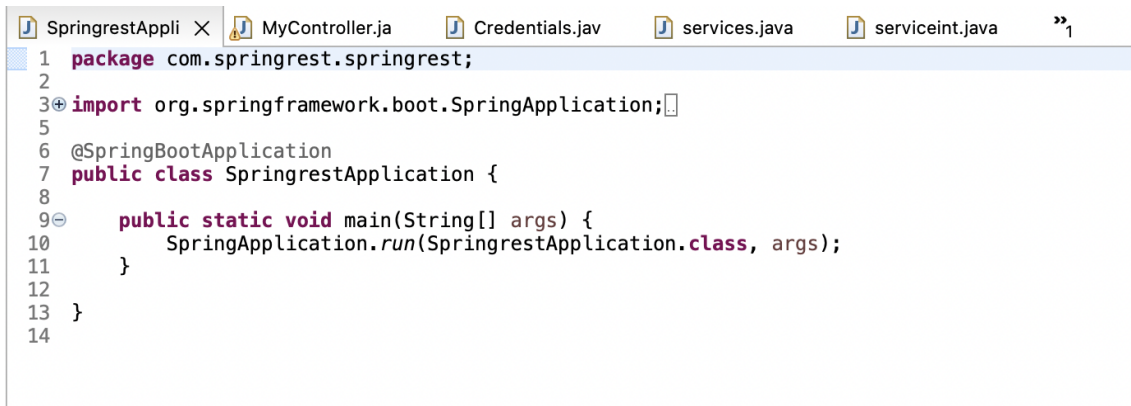
1 pravar

designing an proper rest api

- start from the controller class add the getmapping methods required
- create this class in a new controller package
- now create the entities required ie what you need here we wanted credentials of a person so we made a credentials class in the entities package.
- this class is a bean like class in springboot with all setters getters and constructors with a to string method

- now this controller needs to get the json list of credentials unlike the simple print statement above so it takes help of a service so we need to make a service package which has a parent interface whose methods are implemented by its child class in the same package (it is not necessary but it is convention followed in REST)
- for our eg we took a arraylist of credential class items and added 3 records to it
- then in our main controller class we make a autowired tagged object of the service interface
- then use it and call that service method under getmapping tag with the ultimate get credentials method

the following are the ss of the eg from which the above points are inspired-



```

1 package com.springrest.springrest;
2
3 import org.springframework.boot.SpringApplication;
4
5
6 @SpringBootApplication
7 public class SpringrestApplication {
8
9     public static void main(String[] args) {
10         SpringApplication.run(SpringrestApplication.class, args);
11     }
12
13 }
14

```

```

SpringrestAppli  MyController.java X Credentials.jav services.java serviceint.java ”1
1 package com.springrest.springrest.controller;
2
3 import java.util.List;
4
5 import org.springframework.beans.factory.annotation.Autowired;
6
7 import org.springframework.web.bind.annotation.GetMapping;
8 import org.springframework.web.bind.annotation.RestController;
9
10 import com.springrest.springrest.entities.Credentials;
11 import com.springrest.springrest.services.serviceint;
12 import com.springrest.springrest.services.services;
13
14 @RestController
15 public class MyController {
16     @GetMapping("/home")
17     public String getname() {
18         return "pravar";
19     }
20     @Autowired
21     public serviceint si;
22     @GetMapping("/credentials")
23     public List<Credentials> GetCredentials(){
24         return this.si.GetCredentials();
25     }
26 }
27
28 }
29

```

```

SpringrestAppli  MyController.java Credentials.jav X services.java serviceint.java ”1
1 package com.springrest.springrest.entities;
2
3 public class Credentials {
4     private long id;
5     private String name;
6     private String verbalpasscode;
7     public Credentials() {
8         super();
9         // TODO Auto-generated constructor stub
10    }
11    public Credentials(long id, String name, String verbalpasscode) {
12        super();
13        this.id = id;
14        this.name = name;
15        this.verbalpasscode = verbalpasscode;
16    }
17    public long getId() {
18        return id;
19    }
20    public void setId(long id) {
21        this.id = id;
22    }
23    public String getName() {
24        return name;
25    }
26    public void setName(String name) {
27        this.name = name;
28    }
29    public String getVerbalpasscode() {
30        return verbalpasscode;
31    }
32    public void setVerbalpasscode(String verbalpasscode) {
33        this.verbalpasscode = verbalpasscode;
34    }
35    @Override
36    public String toString() {
37        return "Credentials [id=" + id + ", name=" + name + ", verbalpasscode=" + verbalpasscode + "]"
38    }
39 }
40
41 }
42

```

```






















SpringrestAppli  MyController.java  Credentials.java  services.java X  serviceint.java  "1
1  package com.springrest.springrest.services;
2
3  import java.util.ArrayList;
4  import java.util.List;
5
6  import org.springframework.stereotype.Service;
7
8  import com.springrest.springrest.entities.Credentials;
9
10 @Service
11 public class services implements serviceint {
12     List<Credentials> l;
13
14     // Constructor to initialize the list of Credentials
15     public services() {
16         cred();
17     }
18
19     private void cred() {
20         l = new ArrayList<>();
21         l.add(new Credentials(1, "pravar", "something"));
22         l.add(new Credentials(2, "ram", "nothing"));
23         l.add(new Credentials(3, "rahul", "everything"));
24     }
25
26     @Override
27     public List<Credentials> GetCredentials() {
28         return l;
29     }
30 }
31

```

```



SpringrestAppli  MyController.java  Credentials.java  Previous Annotation (⌘⌘.) .java  serviceint.java X  "1
1  package com.springrest.springrest.services;
2
3  import java.util.List;
4  import com.springrest.springrest.entities.Credentials;
5
6  public interface serviceint {
7      public List<Credentials> GetCredentials();
8  }
9

```

- ▼  springrest [boot]
 - ▼  src/main/java
 - ▼  com.springrest.springrest
 - >  SpringrestApplication.java
 - ▼  com.springrest.springrest.controller
 - >  MyController.java
 - ▼  com.springrest.springrest.entities
 - >  Credentials.java
 - ▼  com.springrest.springrest.services
 - >  serviceint.java
 - >  services.java
 - >  src/main/resources
 - >  src/test/java
 - >  JRE System Library [JavaSE-17]
 - >  Maven Dependencies
 - >  src
 - >  target
 - >  HELP.md
 - >  mvnw
 - >  mvnw.cmd
 - >  pom.xml

HTTPlocalhost:8080/credentials

Save



GETlocalhost:8080/credentials

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

	key	value	description	domain	path	expires
<input checked="" type="checkbox"/>	Postman-Token	<calculated when request is sent>				
<input checked="" type="checkbox"/>	Host	<calculated when request is sent>				

Body

Cookies

Headers (5)

Test Results

200 OK129 ms322 B

Save as Example

Pretty

Raw

Preview

Visualize

JSON

```
1  {
2    {
3      "id": 1,
4      "name": "pravar",
5      "verbalpasscode": "something"
6    },
7    {
8      "id": 2,
9      "name": "ram",
10     "verbalpasscode": "nothing"
11   },
12   {
13     "id": 3,
14     "name": "rahul",
15     "verbalpasscode": "everything"
16   }
17 }
```