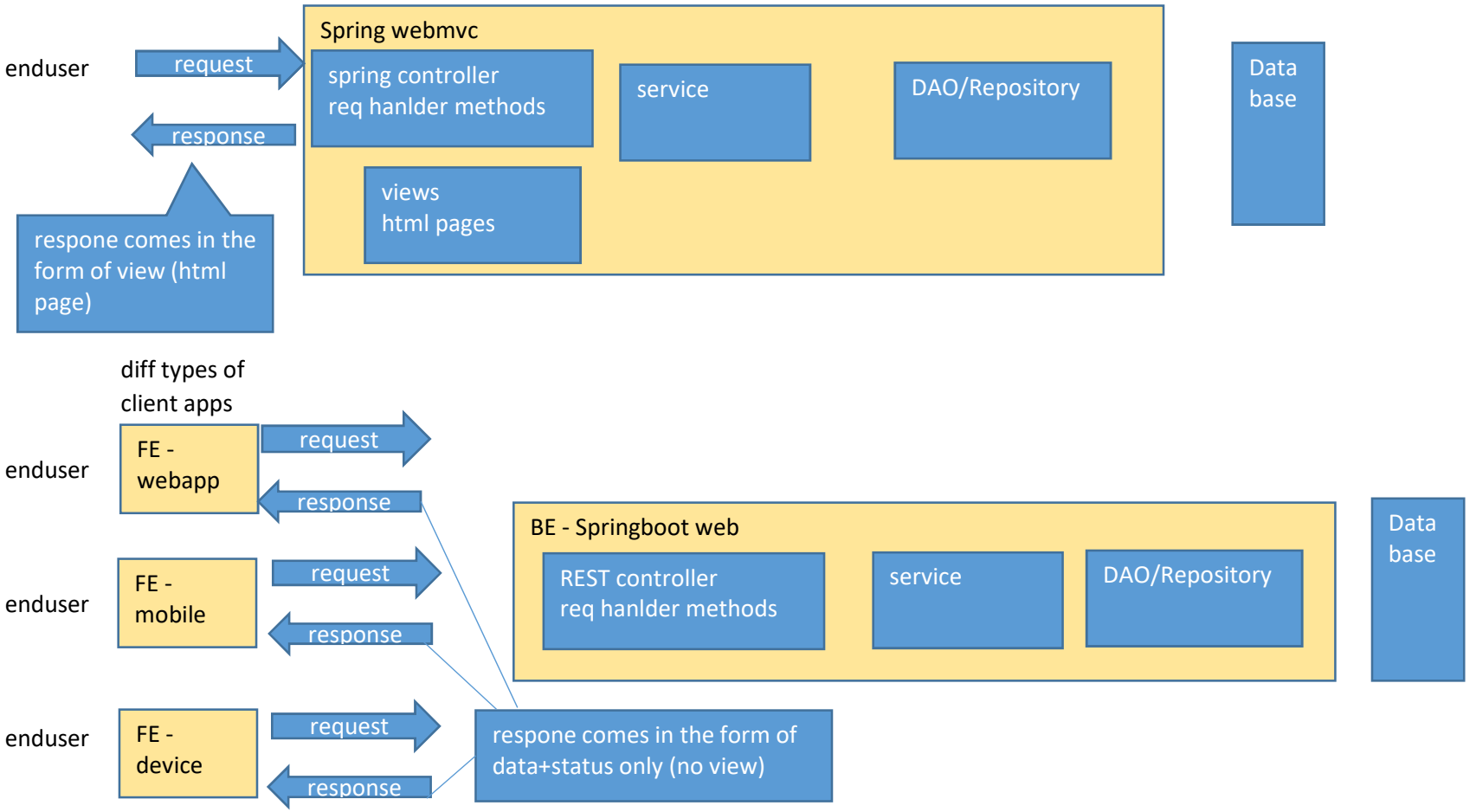
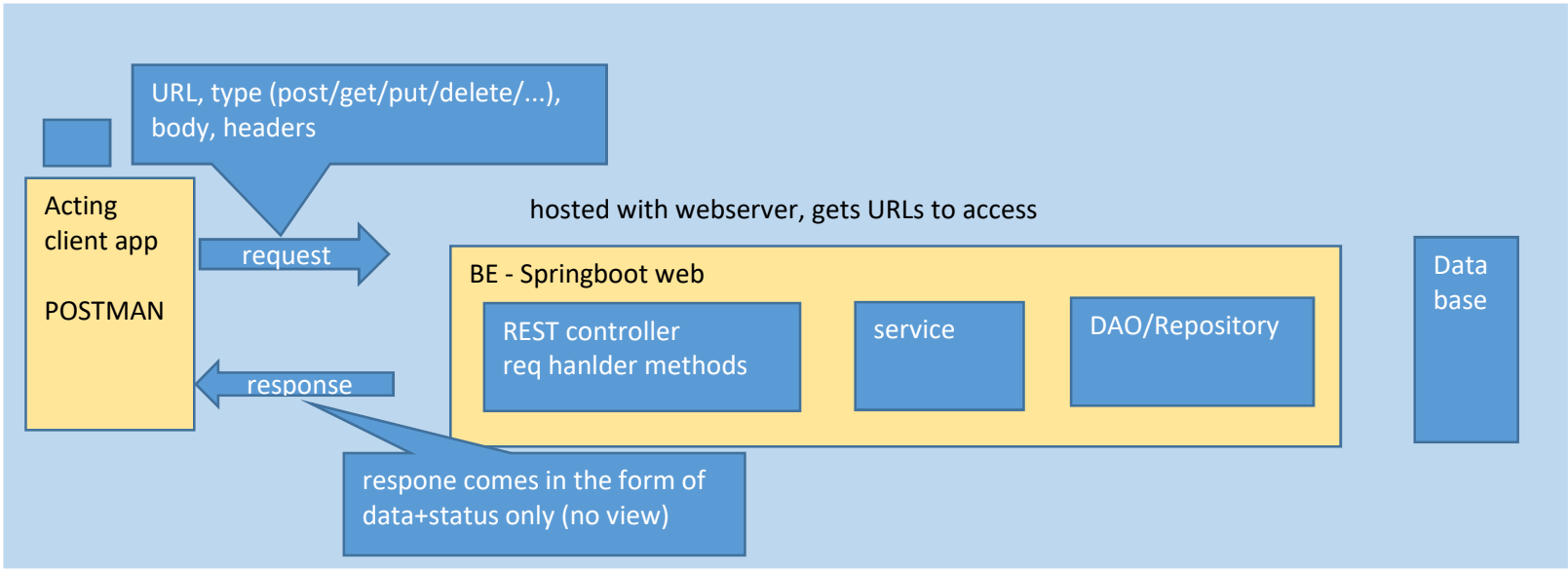
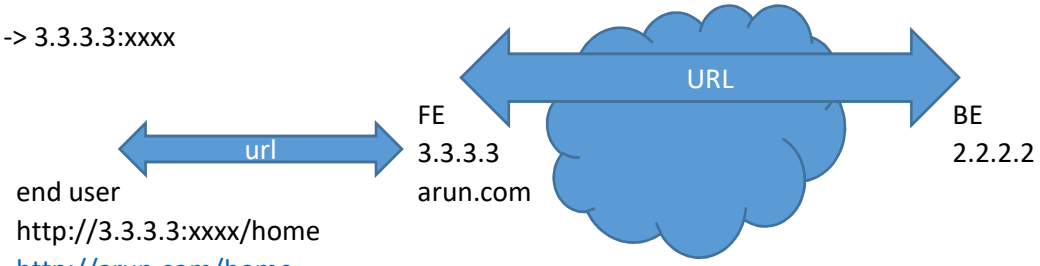
Spring MVC

End User <-> UI <-> Controller <-> Service <-> DAO <-> DB





dns
arun.com -> 3.3.3.3:xxxx



end user
http://3.3.3.3:xxxx/home
http://arun.com/home

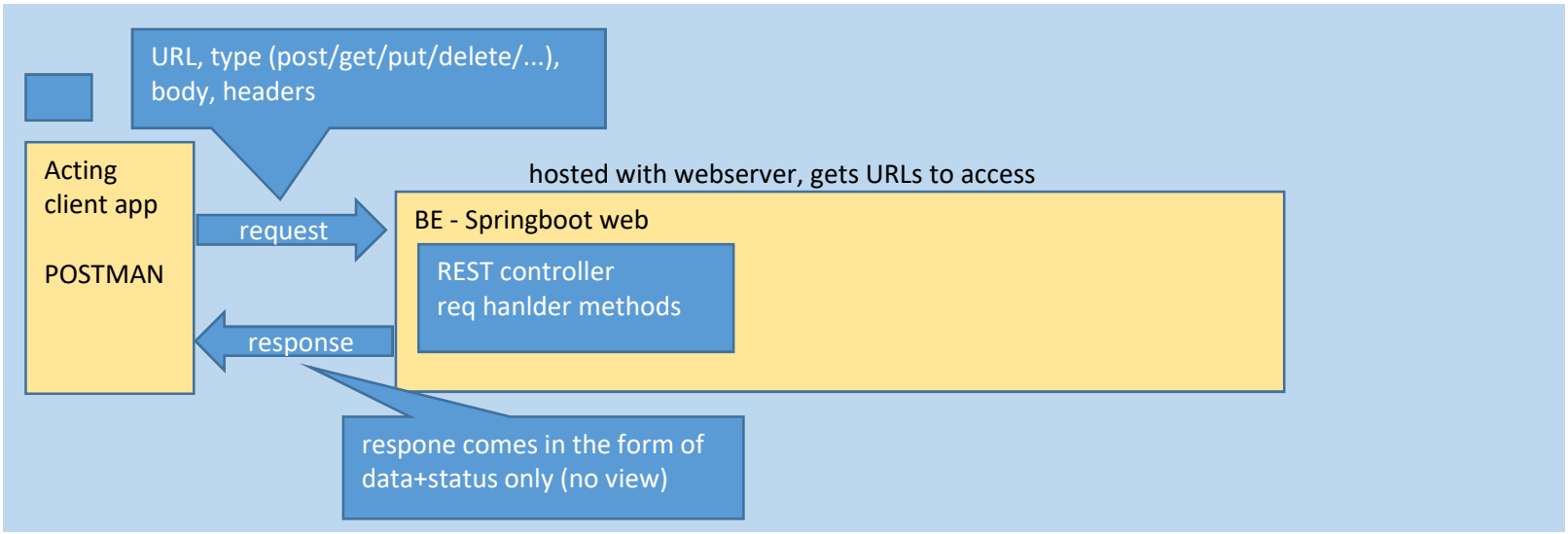
REST Controller

Holds request hanlder methods

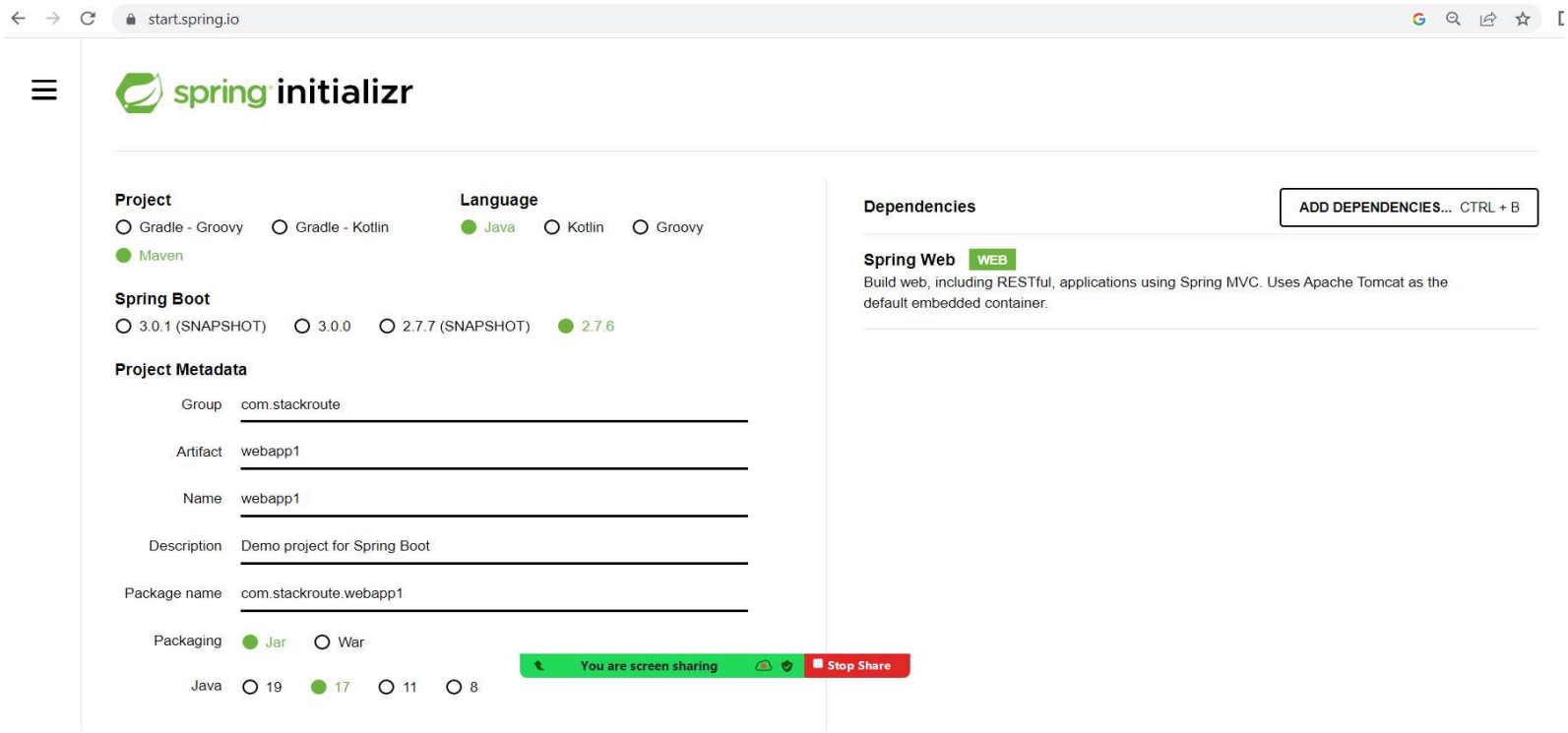
Example 1

Creating simple springboot web application

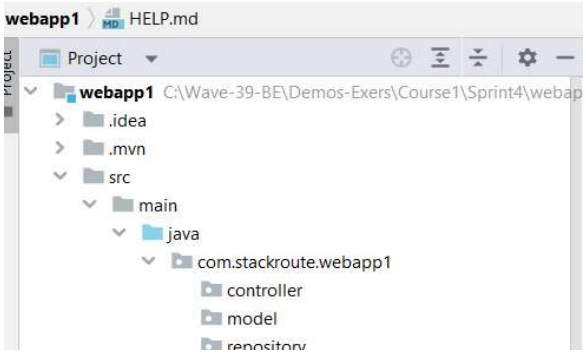
Implementing request handling with REST controller

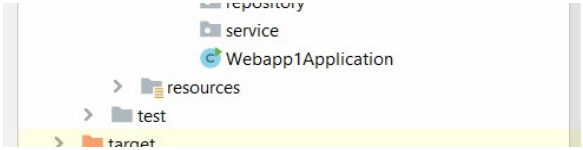


Step 1 Create new springboot application using spring initializer
add dependency 'web'

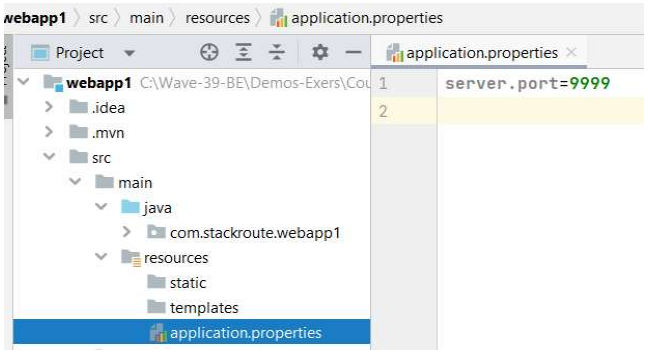


download, extract, copyt to workspace, load in intellij
check settings if required
create required packages

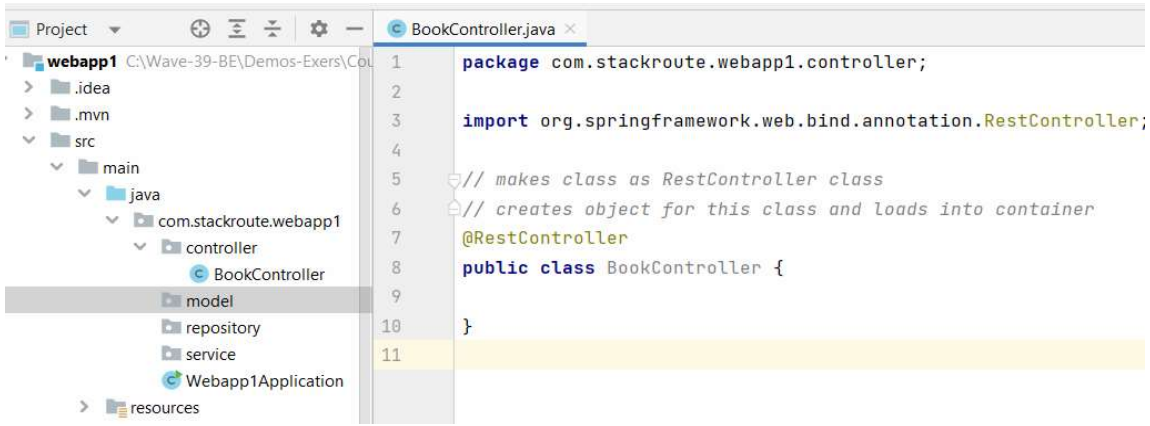




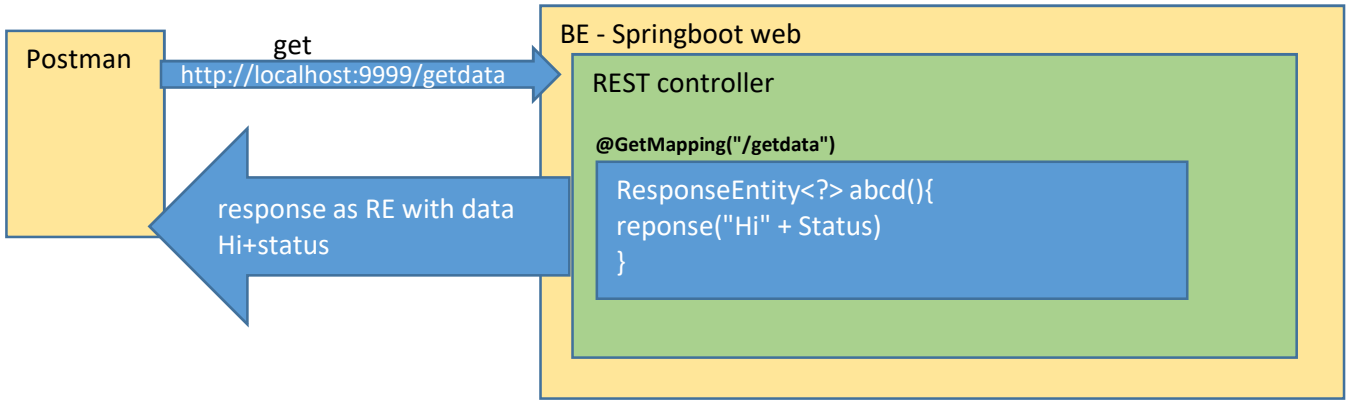
change port number in application.properties



Step 2 Create controller class



BASE_URL
http://localhost:9999/



Request mapping
Associating request with request handler

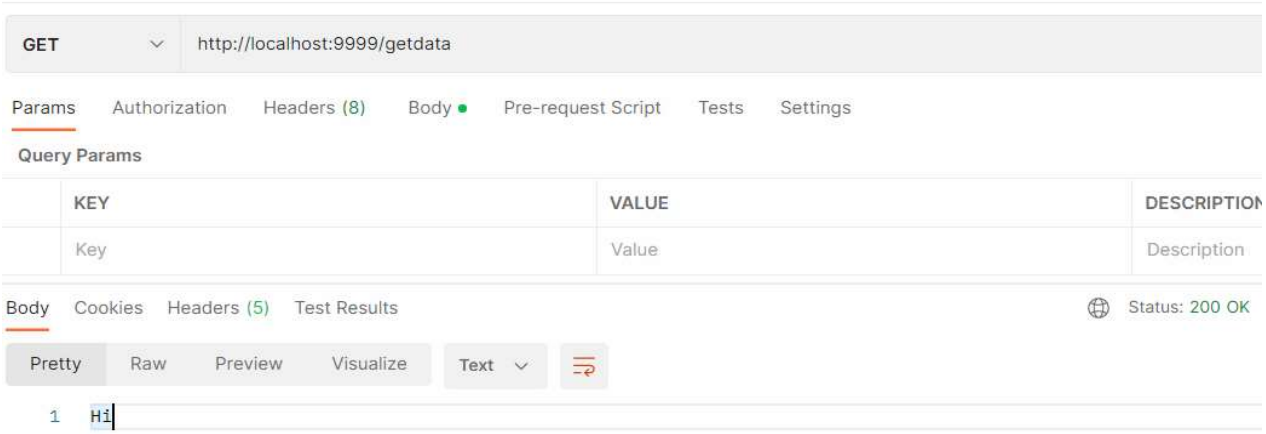
get type /getdata abcd()

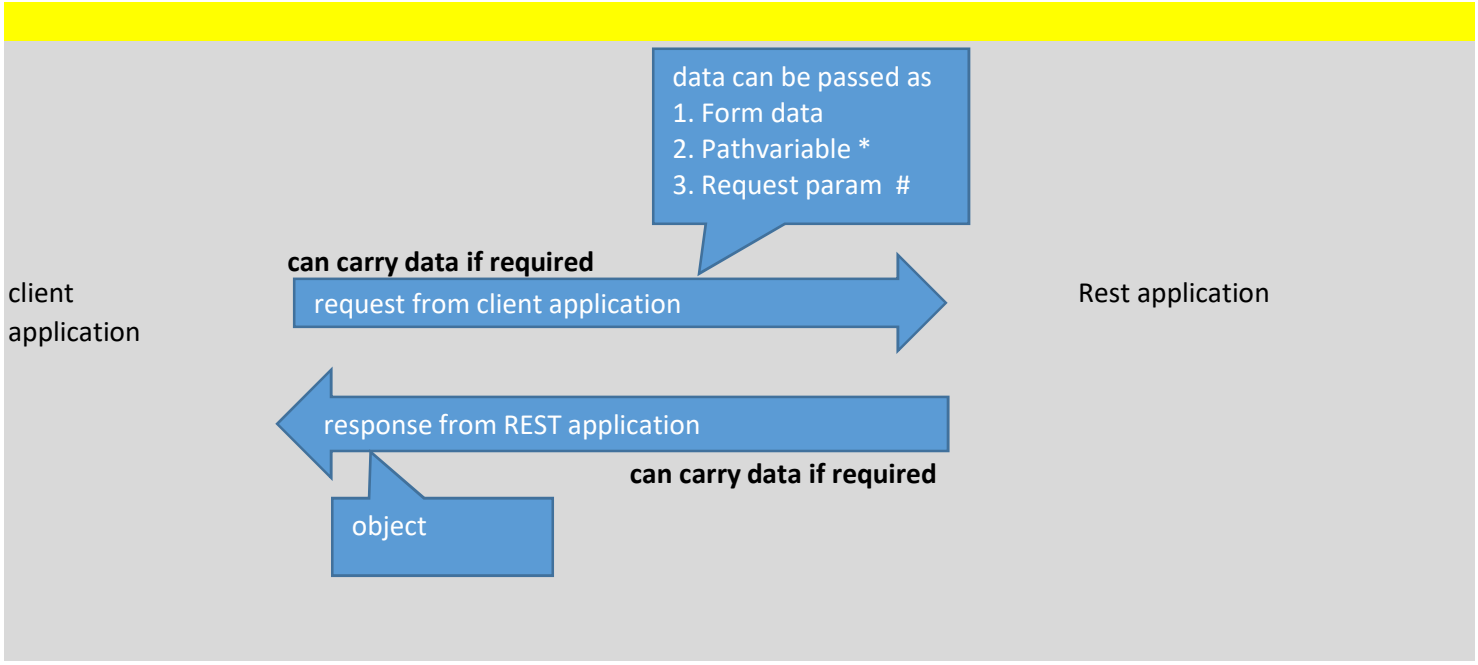
abcd()
should return response with data+status

Step 3 Define request handler method in controller

```
8   // makes class as RestController class
9   // creates object for this class and loads into container
10  @RestController
11  public class BookController {
12
13      @GetMapping("/getdata")
14      public ResponseEntity<?> abcd(){
15          return new ResponseEntity<>( body: "Hi", HttpStatus.OK); // data+status
16      }
17
18  }
```

run application
check in postman by raising request



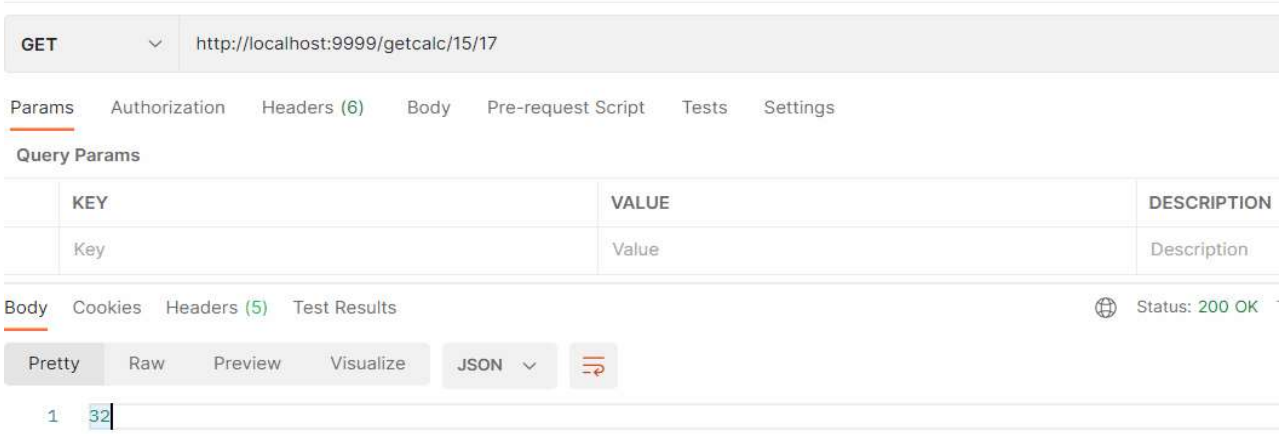


Pathvariable
Data can be passed as part of URL
using '/' as seperator

BASE_URL	http://localhost:9999
BASE_URL/request	http://localhost:9999/calculate
BASE_URL/request/PV1/PV2	http://localhost:9999/calculate/20/3

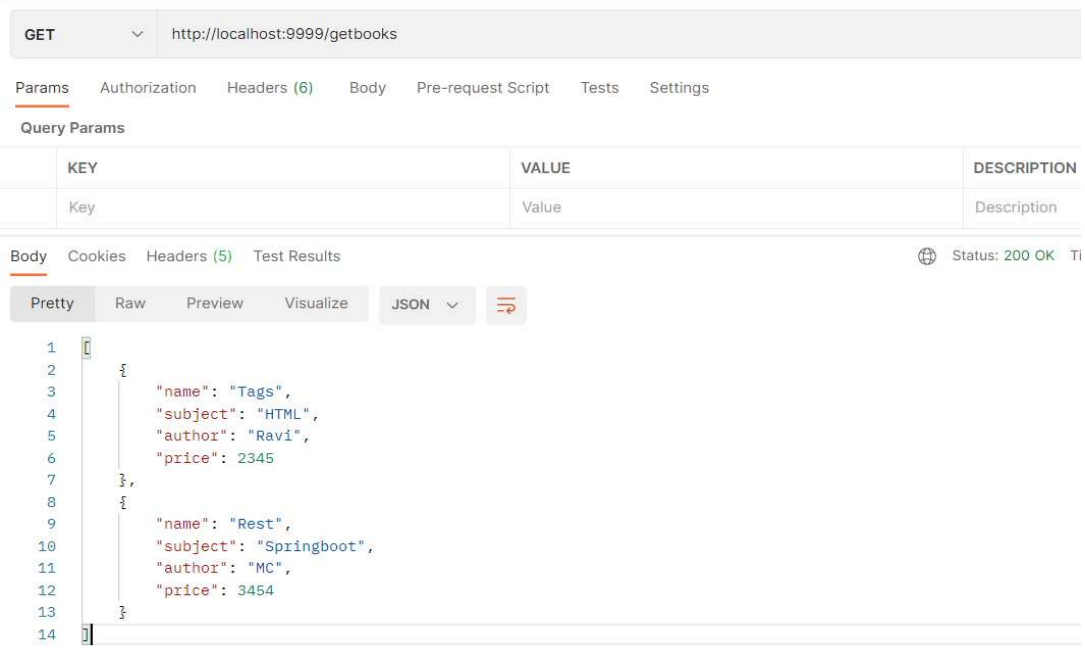
defining request handler method in controller to receive data which comes along with request

```
23      /* GET
24      http://localhost:9999/getcalc/20/3
25       */
26      @GetMapping("/getcalc/{value1}/{value2}")
27      public ResponseEntity<?> calculate(@PathVariable int value1, @PathVariable int value2){
28          int result=value1+value2;
29          return new ResponseEntity<>(result,HttpStatus.OK);
30      }
```



getting book objects as response

```java
/* GET
http://localhost:9999/getbooks
 */
@GetMapping("/getbooks")
public ResponseEntity<?> getBooks(){
    Book books[ ] = {
        new Book( name: "Tags", subject: "HTML", author: "Ravi", price: 2345),
        new Book( name: "Rest", subject: "Springboot", author: "MC", price: 3454)
    };
    return new ResponseEntity<>(books,HttpStatus.OK);
}
```



sending book data as form data

POST http://localhost:9999/addbook Request handler method has to receive as java object

carry book object in JSON format

```java
/*
   POST
http://localhost:9999/addBook
 */
@PostMapping("/addBook")
public ResponseEntity<?> addBook(@RequestBody Book book){
    System.out.println(book);
    return new ResponseEntity<>( body: "Data received",HttpStatus.OK);
}
```