Rest Api

**Created by**

**Saurav Kumar Jha(**[**321sauravkumar@gmail.com**](mailto:321sauravkumar@gmail.com)**)**

**Note:** you can’t sell or re-distribute without the consent of the author

**REST**

transfer(send/receive)

state

representational

🡪what represent state in object oriented language

object represent the state and fields holds the state in object

🡪we have learned this in transaction like new state,stable etc etc,

methods don’t hold the state

suppose we have railway monolithic app we are sending the request and giving the html as response

indian railway

user

frontend

backend

request

response

processing happens here

DB

this is fine but instead the book ticket directly from railway app we can also book the railway ticket using makeMyTrip and paytm and other application in that case

request1 Indian railway server

MakeMyTripp

user1

java backend

dao layer methods

service layer methods

request2 traindetails

response response

ticketdetails

DB

bookTicket

Paytm

endpoints

user2

makeMyTrip has only view not the model(data) so when user will request to the makeMyTrip let’s say request1 then makeMyTrip will internally request to railway server request2 for the data then it will get the response from railway server then it will give back the response to the user same for paytm

now problem is that you are saying that railway server will give the object to the paytm as the response and railway backend is written in java and now paytm let’s say javascript application or php application etc and paytm application will no bet having jvm and railway server is sending java object means bytecode and paytm

application doesn’t have jvm then how it will use the object so railway server will not send the object ,it will send the representation of object and obviously this representation should be language independent, platform independent etc like xml or json

and in spring there is a default support for json

now railway server has java object so it will not send the java object it will send its representation that is json so now if paytm side is either php application or ruby application etc it will process the json and same for request data just like response data

java server

railway server

response

request

representation

{“id”:1,”name”:”gatimaan”}

request request(json)

user

response response(json) java

php/ruby/python/java etc

request request(json)

response response(json)

user frontend(Angular) java server

now in REST we work on resources , we do CRUD operation on resources , resource can be Product, Author, Book etc

C🡪create (POST request method)

R🡪read/fetch (GET request method)

U🡪change (PUT request method but we can also use POST but conventionally wrong)

D🡪delete (DELETE request method)

REST is based on the http protocol

HTTP response

HTTP request

Header

Body(data)

Header

Body(data)

content-type

status code etc

in GET request we will send the data as the part of url

for ex🡪 <http://localhost:8585/trains/id=2>

here id is path variable(@PathVariable)

or

<http://localhost:8585/trains?id=2>

here id is request parameter(@RequestParam)

but in REST path variable is most popular

in POST method we will send the data as the part of the body not url it means you will send the lot of data And it makes sense, also because, then you are creating a resource, you have to send a lot of data that data is not possible, you cannot send in url

For updating(PUT method) again your have to send a lot of data, data you can send as the part of body , same with DELETE

Q. what is @RestController

🡪 @RestController annotation is a special controller used in RESTful Web services, and it's the combination of @Controller and @ResponseBody annotation automatically so we do not have to add @ResponseBody to our mapping methods.. It is a specialized version of @Component annotation

🡪 RestController is used for making restful web services with the help of the @RestController annotation. This annotation is used at the class level and allows the class to handle the requests made by the client

Q. difference between @Controller and @RestController

| @Controller | @RestController |
| --- | --- |
| @Controller is used to mark classes as Spring MVC Controller. | @RestController annotation is a special controller used in RESTful Web services, and it’s the combination of @Controller and @ResponseBody annotation. |
| It is a specialized version of @Component annotation. | It is a specialized version of @Controller annotation. |
| In @Controller, we can return a view in Spring Web MVC. | In @RestController, we can not return a view. |
| @Controller annotation indicates that the class is a “controller” like a web controller. | @RestController annotation indicates that class is a controller where @RequestMapping methods assume @ResponseBody semantics by default. |
| In @Controller, we need to use @ResponseBody on every handler method. | In @RestController, we don’t need to use @ResponseBody on every handler method. |
| It was added to Spring 2.5 version. | It was added to Spring 4.0 version. |

🡪Spring converts the object to json format automatically, If spring was not doing that, then it becomes your responsibility to convert but spring is converting.

🡪what is @RequestBody

@RequestBody annotation maps the HttpRequest body to a transfer or domain object, enabling automatic deserialization of the inbound HttpRequest body onto a Java object. Spring automatically deserializes the JSON into a Java type, assuming an appropriate one is specified

using @RequestBody we are mentioning that json data which is there in the body of request convert to object,spring will intercept and convert from json to object if spring is not doing that you will have to do by yourself