# Flow of Execution in Spring Rest/Spring Boot Rest App

Spring Rest /Spring Boot Rest = Spring MVC++ /spring boot MVC++

=> Technically speaking , there is no spring Rest/ spring boot Rest module .. the Restfull web Services development api of spring MVC or spring boot MVC module is called as Spring Rest or Spring Boot Rest module
=> Using Spring Rest/Spring boot Rest we can develop only RestFull webServices.. not the SOAP based webServices

=> In Spring Rest /Spring Boot Rest Applications the Provider /Producer/Server App will be developed as @RestController class by linking with ready Made FrontController servlet called DispatcherServlet This Producer Applications is called Rest API/ API /Service API development.. and Every Service API and its methods are indentified with URL + request paths + request modes (These are also called as End Point Details)

To invoke each method/endpoint of @RestController we must know that method related base URL+ request Path+ request Mode +params (End Point Details)

@RestController = @Controller + @ResponseBody

makes the endpoint /b.methods of the @RestController class sending generated output as the response to the consumer app directly with out using ViewResolver and the View comps support (Through DispatcherServlet)

=> The Producer/Provider App that is developed as @RestController can interact with service,DAO classes if neccessary while processing requests and generating the results..

=> The methods in @RestContorller class are called as Operations/handler methods / RestOperations/... and they are indentifed with thier request path+ request mode + .... params and etc..

The b.methods of @RestController are claled Rest Operations or Rest Methods or etc... or endpoints or endoperations and etc..

=> The Client for webapplication is browser which gives requests and gets the responses. ( browser to website interaction)
=> The Client for Restful App's Provider/Producer App is another App which can be Destktop App, Mobile App , web application , Embedded System App, IOT app developed in different languages or technologies or frameworks.

Application to Application interaction or Consumer to Producer interaction

One API/RestService = One @RestController
One endpoint = one b. method of @RestController

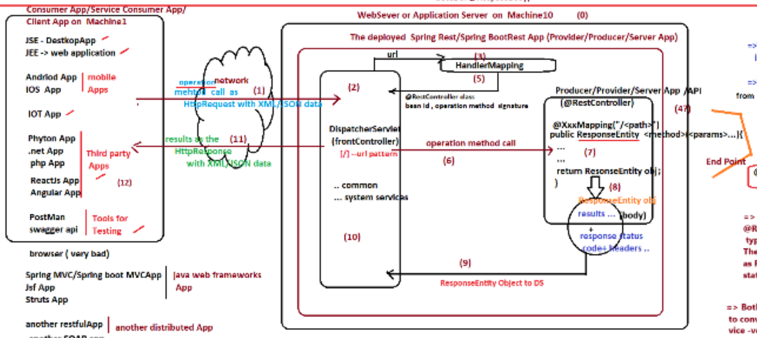
## Spring Rest /Spring Boot Rest App's flow of execution

@RestController = @Controller +@ResponseBody

note:: The methods MVC App's @Controller classes are called handler methods
The method Rest App's @RestController classes are called b.methods or Rest Operations or Rest Methods or Endpoints

=> It is based on spring MVC /spring boot MVC flow of execution.. where ViewResolver and View comps will not there..becoz the @RestController applied on Provider or Producer or Server App automatically applies @ResponseBody on all the service operations/handler /operations methods which makes these methods to send the generated results directly to Client/Cosumer app through FrontController Servlet comp (DispatcherServlet)

(This controller does not use ViewResolver,View comp towards sending output to the client... it directly sends to client app becoz of @ResponseBody)



Consumer App/Service Consumer App/ Client App on Machine1
JSE - DestkopApp
JEE -> web application
Andriod App
IOS App
mobile Apps
IOT App
Phyton App
.net App
php App
ReactJs App
Angular App
Third party Apps
(12)
PostMan
swagger api
Tools for Testing
network
operation mehtod call as
(1)
HttpRequest with XML/JSON data
results as the HttpResponse with XML/JSON data
(11)
WebSever or Application Server on Machine10
(0)
The deployed Spring Rest/Spring BootRest App (Provider/Producer/Server App)
url
(3)
HandlerMapping
(5)
(2)
@RestController class bean id , operation method signature
DispatcherServlet (frontController)
[/] --url pattern
.. common ... system services
(10)
operation method call
(6)
Producer/Provider/Server App /API
(@RestController)
(4?)
@XxxMapping("/<path>")
public ResponseEntity <method>(<params>...){
...
...
return ResonseEntity obj;
}
(7)
(8)
ResponseEntity obj
results ... (body)
+
response status code+ headers ..
(9)
ResponseEntity Object to DS
End Point

=>The Process of keeping App (web application or Distributed App) in Server is called deployment

=>The Process of removing App (web application or Distributed App) from Server is called undeployment

=> ResponseEntity obj holds results/outputs + response status code, response headers ,...

@RestController = @Controller + @ResponseBody

=> The Endpoint /rest operation methods of @RestController , taking ResponseEntity<T> as the return type is recommended option ,but not mandatory option. The other return types just gives result content to DS where as ResponseEntity <T>obj can give results + response statuscode,headers to DS

=> Both Provider and Consumer Apps use XML APIs/JSON APIs to convert global formats (XML/JSON) to Java Objects and vice -versa

browser ( very bad)

Spring MVC/Spring boot MVCApp | java web frameworks App
Jsf App
Struts App

another restfulApp | another distributed App
another SOAP app
...
...
...

W.r.t the Diagram

(0) Programmer deploys Provider app /Restful service App in web server , Since load on startup enabled on DispatcherServlet , the Servletcontainer performs pre-instantiation of DS , as part of DS initialization one IOC container is created and that IOC container performs pre-instantiation of singleton scope spring beans and also completes injections among the spring beans

(1) Consumer App gathers the Proivider App's Endpoints (API endpoints like url + request paths + reuqest modes and etc..) and invokes the Provider App's handler or operation mehtod which intenally generates http request with JSON/XML data .

(2) The FrontController Servlet called DispatcherServlet traps and takes the requests and applies the common system services like auding , loggging and etc..

(3) DispatcherServlet(DS) hand overs the request to Handler Mapping component

(4?) The HandlerMapping comp searches in all the @RestContorller classes for matching request path handler method or operation method and gets that operation method signature and @RestController class bean id

(5) HandlerMapping comp gives @RestController class bean id and operation method signature details to DispatcherServlet comp

(6) DS calls the @RestController class handler method/operation method by arranging the neccessary required arguments and

(7) The handler method or operation method of @RestController class executes either to process request directly or to process request by taking the support of service,DAO classes..

(8)&(9) Handler or operation method prepares and returns ResponseEntity (Most of times) to DS having generated results + response status code, headers and etc..

(10) DS recives the ResponseEntity object and converts the results into XML/JSON content (if the output is other than string)

(11) DS sends the results to Consumer App in the form of HttpRespons with XML/JSON content.

(12) Consumer App takes the results from http response and process them

What is the difference between and Spring MVC/spring boot MVC flow and spring Rest/Spring Boot Rest flow?

Ans) Spring MVC/Spring Boot MVC flow contains the ViewReoslver , View comps to format the generated results using presentation logic becoz the Client is always browser and it wants to display formatted results to enduser

Spring Rest/Spring BootRest flow does not contain the ViewReoslver , View comps becoz of @ResponseBody is applied on the handler/operation method,the response having results will go to Consumer App directly through DS .. More over here the client is not browser.. here client is another serious App which is not expecting html style formatted results ... rather it is expecting global format results(JSON/XML) to use as inputs in its business operation . The @ResponseBody that is applied on Endpoint method indirectly will make the endpoint method of @Restcontroller to send generated /gather output directly to consumer app with out involving View Resolver and View comp

Flow of Execution in Spring Rest/Spring Boot Rest App

Spring Rest/Spring Boot Rest = Spring MVC++/spring boot MVC++

=> Technically speaking, there is no spring Rest/ spring boot Rest module.. the Restfull web Services development api of spring MVC or spring boot MVC module is called as Spring Rest or Spring Boot Rest module

=> Using Spring Rest/Spring boot Rest we can develop only RestFull webServices.. not the SOAP based webServices

=> In Spring Rest /Spring Boot Rest Applications the Provider /Producer/Server App will be developed as @RestController class by linking with ready Made FrontController servlet called DispatcherServlet This Producer Applications is called Rest API/ API /Service API development.. and Every Service API and its methods are indentified with URL + request paths + request modes (These are also called as End Point Details)

=> The Producer/Provider App that is developed as @RestController can interact with service,DAO classes if neccessary while processing requests and generating the results..

=> The methods in @RestContorller class are called as Operations/handler methods / RestOperations/... and they are indentifed with thier request path + request mode + .... params and etc..

To invoke each method/endpoint of @RestController

base

we must know that method related URL+ request Path+ request Mode +params [End Point Details)

@RestController = @Controller + @ResponseBody

makes the endpoint /b.methods of the @RestController class sending generated output as the response to the consumer app directly with out using

The b.methods of @RestController are claled Rest Operations or Rest Methods or etc... or endpoints or endoperations and etc..

Application to Application

=> The Client for webapplication is browser which gives requests and gets the responses. (browser to website interaction) => The Client for Restful App's Provider/Producer App is another App which can be Destktop App, Mobile App, web application, Embedded System App, IOT app developed in different languages or technologies or frameworks.

Spring Rest/Spring Boot Rest App's flow of execution

on

of

ViewResolver and the View comps support (Through DispatcherServlet)

One API/RestService = One @RestController One endpoint = one b. method of @RestController

interaction or Consumer to Producer interaction

@RestController = @Controller +@ResponseBody

note:: The methods The method

=> It is based spring MVC /spring boot MVC flow execution.. where ViewResolver and View comps will not there..becoz the @RestController applied on Provider or Producer or Server App automatically applies @ResponseBody on all the service operations/handler/operations methods which makes these methods to send the generated results directly to tifeftumeshuifwResolver, View comp towards sending output to the client.. it directly sends to client app FrontController Servlet comp (DispatcherServlet) becoz of @ResposeBody)

Consumer App/Service Consumer App/

Client App on Machine1

JSE - DestkopApp

JEE -> web application

MVC App's @Controller classes are called handler methods Rest App's @RestController classes are called b.methods or Rest Operations or Rest Methods or Endpoints

WebSever or Application Server on Machine10

(0)

url

The deployed Spring Rest/Spring Boot Rest App (Provider/Producer/Server App)

Andriod App

IOS App

mobile Apps

operationnetwork

(3) HandlerMapping (5)

mehto call as HttpRequest with XM

(1)

(2)

@RestController class

SON data

bean id, operation method signature

IOT App

Phyton App

.net App

php App

Third party Apps

results as the HttpResponse

(11)

DispatcherServlet (frontController)

[/] --url pattern

with XMLXISON data

operation method call (6)

ReactJs App Angular App

PostMan

(12)

common system services

Tools for

swagger api

Testing

browser (very bad)

Spring MVC/Spring boot MVCApp Jsf App

java web frameworks

App

(10)

(9) ResponseEntity Object to DS

Producer/Provider/Server App /API (@RestController) (4?)

=>The Process of keeping App (web application or Distributed App) in Server is called deployment

=>The Process of removing App (web application or Distributed App) from Server is called undeployment

@XxxMapping("/<path>")

public ResponseEntity <method><params>...){\

(7)

End Point

return ResonseEntity obj;

♫ (8)

ResponseEntity obj

=> ResponseEntity obj holds

results/outputs + response status code, response headers,...

@RestController = @Controller + @ResponseBody

results...body)

+

response status

code+ headers..

=> The Endpoint /rest operation methods of @RestController, taking ResponseEntity<T> as the return type is recommended option,but not mandatory option. The other return types just gives result content to DS where as ResponseEntity <T>obj can give results + response statuscode,headers to DS

Struts App

another restfulApp | another distributed App

another SOAP app

W.r.t the Diagram

(0) Programmer deploys Provider app /Restful service App in web server, Since load on startup enabled on DispatcherServlet, the Servletcontainer performs pre-instantiation of DS, as part of DS initialization one IOC container is created and that IOC container performs pre-instantiation of singleton scope spring beans and also completes injections among the spring beans

(1) Consumer App gathers the Proivider App's Endpoints (API endpoints like url + request paths + reuqest modes and etc..) and invokes

**the Provider App's handler or operation mehtod which intenally generates http request with JSON/XML data.**

**(2) The FrontController Servlet called DispatcherServlet traps and takes the requests and applies the common system services like auding, loggging and etc..**

**(3) DispatcherServlet(DS) hand overs the request to Handler Mapping component**

**(4?) The HandlerMapping comp searches in all the @RestContorller classes for matching request path**

**handler method or operation method and gets that operation method signature and @RestController class bean id**

**(5) HandlerMapping comp gives @RestController class bean id and operation method signature details to DispatcherServlet comp**

**and**

**(6) DS calls the @RestController class handler method/operation method by arranging the neccessary required arguments directly**

**(7) The handler method or operation method of @RestController class executes either to process request directly or to process request by taking the support fervice, DAO classes..**

**(8)&(9) Handler or operation method prepares and returns ResponseEntity (Most of times) to DS having generated results + response status code, headers and etc..**

**(10) DS recives the ResponseEntity object and converts the results into XML/JSON content (if the output is other than string)**

**(11) DS sends the results to Consumer App in the form of HttpRespons with XML/JSON content.**

**(12) Consumer App takes the results from http response and process them**

**What is the difference between and Spring MVC/spring boot MVC flow and spring Rest/Spring Boot Rest flow? Ans) Spring MVC/Spring Boot MVC flow contains the ViewReoslver, View comps to format the generated results using presentation logic becoz the Client is always browser and it wants to display formatted results to enduser**

**Spring Rest/Spring Boot Rest flow does not contain the ViewReoslver, View comps becoz of @ResponseBody is applied on the handler/operation method,the response having results will go to Consumer App directly through DS.. More over here the client is not browser.. here client is another serious App which is not expecting html style formatted results .. rather it is expecting global format results(JSON/XML) to use as inputs in its business operation . The @ResponseBody that is applied on Endpoint method indirectly will make the endpoint method of @Restcontroller to send generated /gather output directly to consumer app with out involving View Resolver and View comp**

**=> Both Provider and Consumer Apps use XML APIs/JSON APIs to convert global formats (XML/JSON) to Java Objects and vice -versa**