=>

{REST Architecture Principles }

============================

1. Unique Address
2. Uniform Constraint interfaces } -> bind methods to HTTP method:
3. Media Representation -> 100% — Accesibity / Adopbility
4. Communication Stateless }
5. Hateos }

JSON/XML

UDDI

Interopraability

Webservices

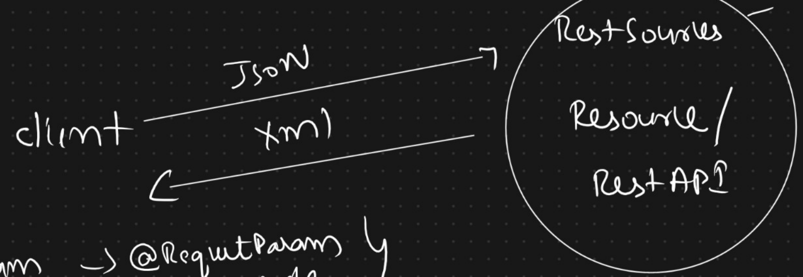
WSDL

Consumer — xml ------> Producer }

=> Restfulservices :-> 100% intropraility -> xml } -> JSON

Restfulservices

Source



Bounces / API / -> binded HTTP method

=> { QueryParam -> @RequestParam }
PathParam -> @Pathvariable

---------- {404} ----------

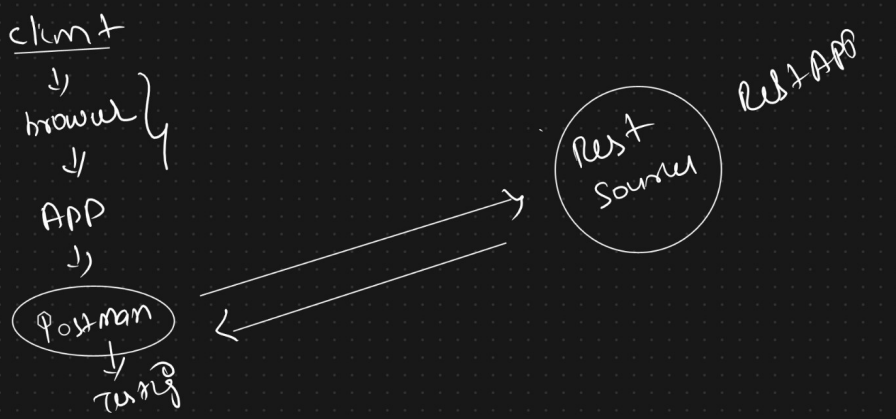
=> spring-mvc -> stable

=> { @RestController => response -> raw Data

⇓

=> required methods within @RestController annoted class should be binded to HTTP Request method -> unique url-pattern

@GetMapping
@PostMapping
@PutMapping
≡



@Rest Controller :-(Spring 4.0) @Controller + @ResponseBody
Rest source:-Provides business service to other applications
Rest Client:-Who seek other application services(Device,Application,Human accessing from browser
(Any one capable of sending HTTP Request)

UI to Controller
1.Queryparam (http://localhost:8085/MyApp/message2?name=janardhana&cource=java)
2.Pathparam (http://localhost:8086/MyApp/message2/janardhana/microservices)

Whenever resource (Rest)Send data to client then we must bind method to GET Request method
example:client want to access some data (Product details/Employee details)
Http Get Method is safe idempotent

Query Parameter

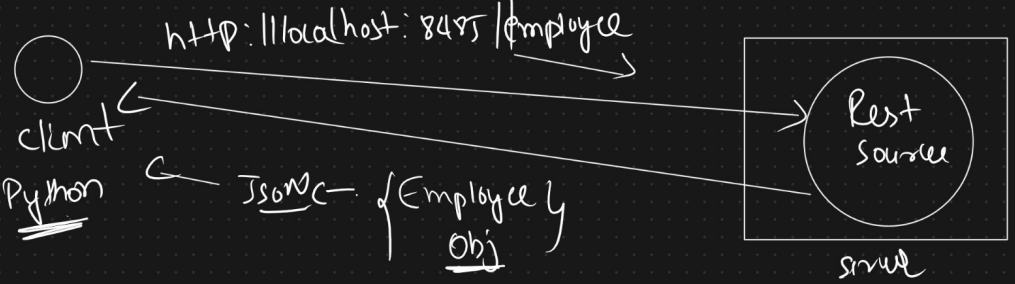
- send Data → client to server through URL
- data will in key-value pair
- present at the end of URL
- it starts with ? & if more data then separated with &
- @RequestParam is used to read Query Param

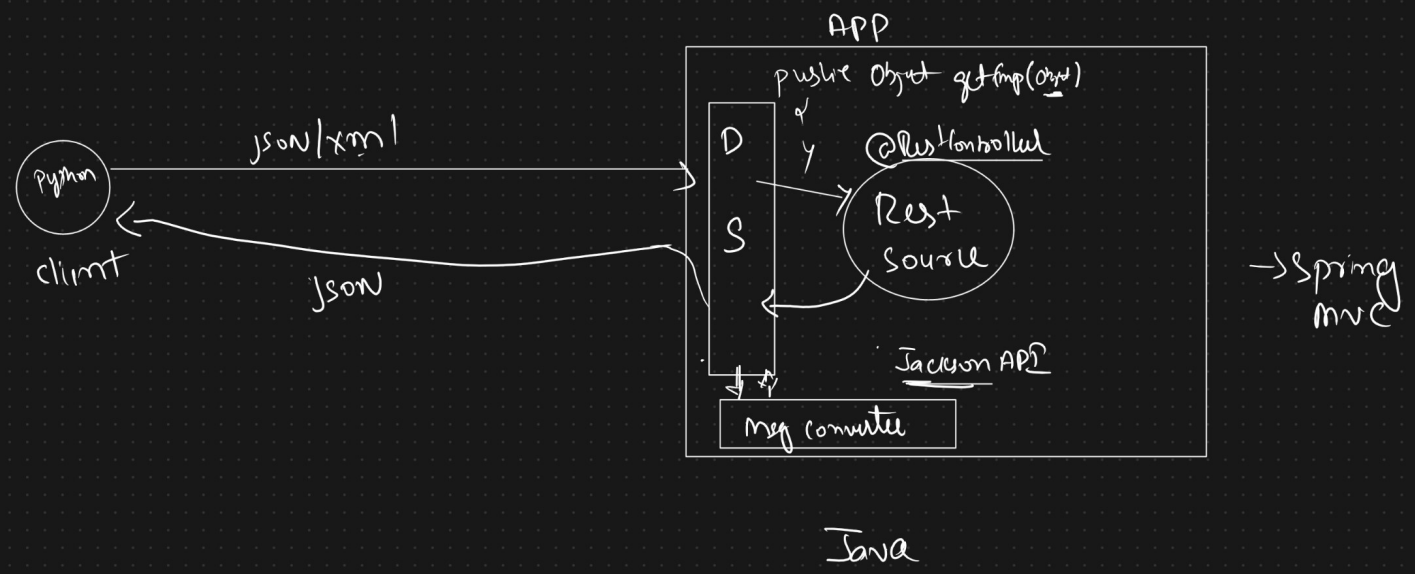
Path Parameter

- send Data → client to server - URL
- directly values in URL
- anywhere in URL

it starts with / more data then we use / as separator

@PathVariable is used to read data.





JSON --> Java Script Object Notation
==> It is very lightweight compare to XML
==>Its one the most popular or defacto standard to exchange data over network

syntax ==>

```
{
   "id": 102,
   "name": "Rohan",
   "city": "Bengaluru"
}
```

JacksonAPI :-) JSON processor for java

↳ convert java object to JSON & jsonData to Java object

(JSON Parser)

JSON → Universal

Jackson → Java specific

ObjectMapper class provided by Jackson api is used to convert Java obj to JSON format

```
    Student st=new Student();
          st.setId(101);
          st.setName("Rohit");
          st.setCity("Bengaluru");

          ObjectMapper mapper = new ObjectMapper();
         String json=  mapper.writerWithDefaultPrettyPrinter().writeValueAsString(st);
//          String str = mapper.writeValueAsString(st);
          System.out.println(json);
```