

@component
@service
@repository

Spring give object to us

Service :- business logic

Repository:- dao data base interaction and other logic.

Service instead Repository for readability
Help to handle persistent related exception

Common DataAccess Exception

Bean life cycle
@postconstruct
@predestroy
E.g database connections create in pre construction

ContextLoaderListner
Start and stop container

Restclient using in application? How?

Rest template

Openfin

```
public ProductDTO getProductDataByIdNew(int id) {  
    String url = "http://localhost:8080/products/{id}";  
    ProductDTO productDTO = restTemplate.getForObject(url, ProductDTO.class,id);  
    return productDTO;  
}
```

Getforentity or getforobject

```
public ProductDTO getProductDataByIdNew(int id) {  
    String url = "http://localhost:8080/products/{id}";  
    ProductDTO productDTO = restTemplate.getForObject(url, ProductDTO.class,id);  
    return productDTO;  
}
```

GetforObject Return Object

GetforEntity Return response header + object body etc

```
public ResponseEntity<ProductDTO> getProductDataById(int id) {  
    String url = "http://localhost:8080/products/{id}";  
    ResponseEntity<ProductDTO> response = restTemplate.getForEntity(url, ProductDTO.class,id);  
    System.out.println(response.getStatusCode());  
    System.out.println(response.getBody());  
    System.out.println(response.getHeaders());  
  
    return response;  
}
```

Exchange method as well

Response entity advantages

Extra info can send status code etc

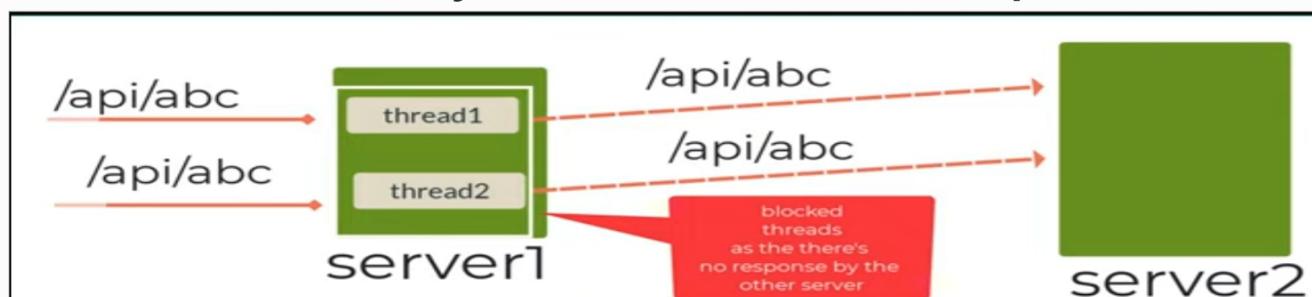
```
@GetMapping("/customers")
public ResponseEntity<List<Customer>> getAllCustomers(){
    List<Customer> customers = customerRepo.findAll();
    return ResponseEntity.status(HttpStatus.ACCEPTED).body(customers);
}
```

Delete method what kind of response send?

Delete api status code :
204 - No content
200 - ok



Why do you think timeout should be sent?
If one thread is block
Another user try to use it it create problem.



With timeout thread release.

Thread pool where all thread resides

Versioning in webservice

Extra parameter version

Uri change

```
@GetMapping("/course/v1")
public CourseReponseV1 getCourseDetailsV1() {
    return new CourseReponseV1("Spring With Spring boot");
}

@GetMapping("/course/v2")
public CourseReponseV2 getCourseDetailsV2() {
    CourseReponseV2 courseReponseV2 = new CourseReponseV2("Spring With Spring boot");
    CourseDetail courseDetail = new CourseDetail("Abhilash", "30 days");
    courseReponseV2.setCourseDetail(courseDetail);
    return courseReponseV2;
}
```

Authentication and security how to handle
Security configuration class

Basic authentication works

Request- Authentication Filter - Authentication
Manager Filter - security context folder

Check and allow sign in

Header

Jwt token pass

Other server validate like I only generated

Session base approach basic Authentication

Which approach better why?

Token base

- It should be stateless. Server not storing state of client
- Many systems might hit the servers.

Bearer meaning

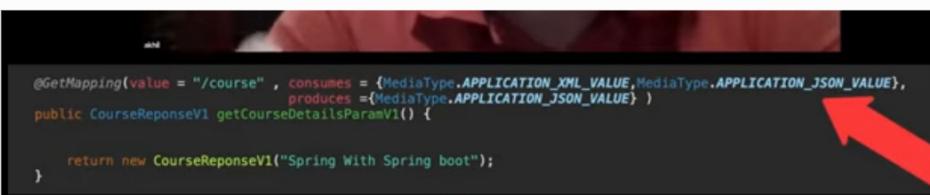
No further validation required.

Jwt use for Authentication or authorization?

- Authorization
- it will be done after Authentication

What is content negotiation?

- what kind of content ur api accept
- what kind of content ur api produces
- via media type



```
@GetMapping(value = "/course" , consumes = {MediaType.APPLICATION_XML_VALUE,MediaType.APPLICATION_JSON_VALUE} , produces ={MediaType.APPLICATION_JSON_VALUE} )
public CourseReponseV1 getCourseDetailsParamV1() {
    return new CourseReponseV1("Spring With Spring boot");
}
```

What is cross cutting concerning?

For all cross cutting concerns
Spring AOP
is the solution

Logging
Security
Monitor

Api level microservices architecture requires

Custom validators in Spring
@interface

Internal working of hashmap
List vs set
ArrayList with size faster
Failsafe vs Failfast collections
Which type of collection consume more memory

fail fast or fail saf consume more memory?
Fail safe it will iterate on copy

```
For()  
sop(s);  
Al.remove();  
)
```

Concurrent modification exception

Gof singleton vs Spring Singleton
1 per container
1 per runtime

Spring security and login flow

- Security dependency
- login page populated
- username password Authentication Filter
- Attempt authentication method
- UsernamePasswordAuthenticationFilter call

```
public UsernamePasswordAuthenticationFilter() {  
    super(DEFAULT_ANT_PATH_REQUEST_MATCHER);  
}  
  
public UsernamePasswordAuthenticationFilter(AuthenticationManager authenticationManager) {  
    super(DEFAULT_ANT_PATH_REQUEST_MATCHER, authenticationManager);  
}  
  
@Override  
public Authentication attemptAuthentication(HttpServletRequest request, HttpServletResponse response)  
    throws AuthenticationException {  
    if (this.postOnly && !request.getMethod().equals("POST")) {  
        throw new AuthenticationServiceException("Authentication method not supported: " + request.getMethod());  
    }  
    String username = obtainUsername(request);  
    username = (username != null) ? username : "";  
    username = username.trim();  
    String password = obtainPassword(request);  
    password = (password != null) ? password : "";  
    UsernamePasswordAuthenticationToken authRequest = new UsernamePasswordAuthenticationToken(username, password);  
    // Allow subclasses to set the "details" property  
    setDetails(request, authRequest);  
    return this.getAuthenticationManager().authenticate(authRequest);  
}
```

Authentication manager

Base on Authentication provider it work.

