

//Spring Boot Notes
By:- " Gauravsing Rajput "

@Component :- this is the managed bean after putting this controller spring will create the object for u in the container

Eg:-

```
@Component    // here it will automatically create the object
public class Dev {
    public void build()
    {
        System.out.println("Hello World");
    }
}
```

Learning Step Two :-
we learn below annotations

@Autowired
@Qualifier

@Component// this is the managed bean after putting this controller spring will create the object for u in the container

```
public class Dev {
    //Laptop laptop;//by default it will get null value
    //two ways to solve this problem one is creating a new object
    //eg
    // first is => // Laptop laptop=new Laptop();
    //and another is @Autoeired annotation
```

@Qualifier("laptop");// here laptop instance is provided so this instance will called

```
@Autowired// do that in top its a field injection
private Computer computer;
```

```
//or you can use constructor method
/*
    public Dev(Laptop laptop){
        this.laptop=laptop;
    }
* */
```

```
//or
//you can use setter injuction
/* @Autowired //here autowired is required in setteer injection
* public void setLaptop(Laptop laptop)
* {
* this.laptop=laptop
* }
```

```

* */

public void build()
{
    computer.compile();
    System.out.println("Hello World dev");
}
}

```

learning step three:-

@RestController
 @RequestMapping
 @Controller
 @ResponseBody
 this annotation we learn below
 Eg:

```

//if we remove rest it will return the page
@RestController//it transfer the data only not a layout
//@Controller//it handels a request
public class HomeController {
    @RequestMapping("/")
    // @ResponseBody//this means i am not looking for the page i am looking for the
    body
    public String greet()
    {
        return "Hello World";
    }

    @RequestMapping("/about")
    public String about()
    {
        return "we dont teach we educate";
    }
}

```

learning step four : Get

@Service

```

@Service // this annotation use to declear class as a service
public class ProductService {
    List<Product> products= Arrays.asList(new Product(101,"iphone ",7000),
        new Product(102,"idk",9000),
        new Product(103,"udk",20000));
}

```

// this is the service method to get the products

```

    public List<Product> getproduct()
    {
        return products;
    }
}
//for getting the above data we use below method

```

```

@RestController
public class ProductController {

    // this is the api to get the product
    @Autowired
    ProductService service;

    @RequestMapping("/products")
    public List<Product> getProduct()
    {
        return service.getproduct();
    }
}

```

learning step five :- Get
 // in this we mad a api of getting product by id

```

// this is the service method

    public Product getProdyctByid(int prodId) {
        return products.stream()
            .filter(p->p.getId()==prodId)
            .findFirst().orElse(new Product(100,"no item",0));
    }

```

//and its controller is

```

    @GetMapping("/products/{prodId}")
    public Product getProductByid(@PathVariable int prodId)//path variable is to
    match tha prodId in requestmappting
    {
        return service.getProdyctByid(prodId);
    }

```

learning step six : - Post
 //in this step we are making api for adding product

//below a serive method

```

    public void addProduct(Product prod)
    {
        products.add(prod);
    }

```

```
}
```

```
// and its controller is
```

```
@PostMapping("/products/add")  
public void addProduct(@RequestBody Product product)  
{  
    System.out.println(product);  
    service.addProduct(product);  
}
```
