8. Spring boot : Bean Scopes | Singleton, Prototype, Request, Session Scopes with Examples in Java

Prerequisite to understand Scopes: Beans and its lifecycle.

Life Cycle Of Bean.

Application Start > **IOC Container Started [Configuration Loaded]** > **Construct Bean** > **Inject Dependency Into Constructed Bean** > **@PostConstruct** > **Use the Bean** > **@PreDestroy** > **Bean Destroyed**

Scopes

1. Singleton > ONE OBJECT FOR IOC
2. Prototype
3. Request
4. Session
5. Application > ONE Object for different IOC LIKE IOC1 IOC2 IOC3 AND SO ON.

1. **Singleton**

- Default scope
- Only 1 instance created per IOC
- Considering 1 application is running on ONE IOC
- Eagerly initialized by IOC [Means at the time of application startup, object get created]

There Is No Scope defined so, by default it is singleton.

```
package com.springProject.SpringCourseProject.controller;

import com.springProject.SpringCourseProject.component.User;
import jakarta.annotation.PostConstruct;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping(value = "/api/")
public class TestController1 {

    @Autowired
    User user;

    public TestController1(){
        System.out.println("TestController1 Instance Initialization");
    }


    //This will call only after construction of all @Autowired or all Dependency
Injection
    @PostConstruct
    public void init(){
        System.out.println("TestController1 object hascode: "+this.hashCode()+" User
object Hashcode: "+user.hashCode());
    }

    @GetMapping(path = "/fetchUser")
    public ResponseEntity<String> getUserDetails(){
        System.out.println("Fetch user api invoked");
        return ResponseEntity.status(HttpStatus.OK).body("");
    }
}
```

With Scope oriented.

It's already been singleton but for demo purpose you can also define it.

@Scope(value = ConfigurableBeanFactory.SCOPE_SINGLETON)

```
package com.springProject.SpringCourseProject.controller;

import com.springProject.SpringCourseProject.component.User;
import jakarta.annotation.PostConstruct;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.config.ConfigurableBeanFactory;
import org.springframework.context.annotation.Scope;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping(value = "/api/")
@Scope(value = ConfigurableBeanFactory.SCOPE_SINGLETON)
public class TestController2 {

    @Autowired
    User user;

    public TestController2(){
        System.out.println("TestController2 Instance Initialization");
    }

    @PostConstruct
    public void init(){
        System.out.println("TestController2 object hascode: "+this.hashCode()+" User
object Hashcode: "+user.hashCode());
    }

    @GetMapping(path = "/fetchUser2")
    public ResponseEntity<String> getUserDetails(){
        System.out.println("Fetch user api 2 invoked");
        return ResponseEntity.status(HttpStatus.OK).body("");
    }
}
```

output

```
 :: Spring Boot ::                (v3.3.2)


2024-08-19T09:34:28.456+05:30  INFO 16100 --- [SpringCourseProject] [           main]
c.s.S.SpringCourseProjectApplication     : Starting SpringCourseProjectApplication using Java
22.0.2 with PID 16100 (C:\volume e\java_projects\SpringCourseProject\target\classes started by
DELL in C:\volume e\java_projects\SpringCourseProject)
2024-08-19T09:34:28.461+05:30  INFO 16100 --- [SpringCourseProject] [           main]
c.s.S.SpringCourseProjectApplication     : No active profile set, falling back to 1 default profile:
"default"
2024-08-19T09:34:30.417+05:30  INFO 16100 --- [SpringCourseProject] [           main]
o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port 8080 (http)
2024-08-19T09:34:30.446+05:30  INFO 16100 --- [SpringCourseProject] [           main]
o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2024-08-19T09:34:30.447+05:30  INFO 16100 --- [SpringCourseProject] [           main]
o.apache.catalina.core.StandardEngine    : Starting Servlet engine: [Apache Tomcat/10.1.26]
```

```
2024-08-19T09:34:30.577+05:30  INFO 16100 --- [SpringCourseProject] [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2024-08-19T09:34:30.579+05:30  INFO 16100 --- [SpringCourseProject] [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 2023 ms
Initialization User
User object @PostConstruct User object hascode: 999699153
TestController1 Instance Initialization
TestController1 object hascode: 975372289 User object Hashcode: 999699153
TestController2 Instance Initialization
TestController2 object hascode: 715534618 User object Hashcode: 999699153
2024-08-19T09:34:31.328+05:30  INFO 16100 --- [SpringCourseProject] [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port 8080 (http) with context path '/'
2024-08-19T09:34:31.344+05:30  INFO 16100 --- [SpringCourseProject] [           main] c.s.S.SpringCourseProjectApplication     : Started SpringCourseProjectApplication in 3.917 seconds (process running for 4.665)
```

Prototype

- Each time new Object is created.
- Its Lazily Initialized, means when object is created only its required.

```
package com.springProject.SpringCourseProject.component;

import jakarta.annotation.PostConstruct;
import org.springframework.context.annotation.Scope;
import org.springframework.stereotype.Component;

@Component
@Scope("prototype")
public class User{
    public User(){
        System.out.println("Initialization User");
    }
    @PostConstruct
    public void init(){
        System.out.println("User object @PostConstruct User object hascode: 
"+this.hashCode());
    }
}
```

```
package com.springProject.SpringCourseProject.component;

import jakarta.annotation.PostConstruct;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Scope;
import org.springframework.stereotype.Component;

@Component
//By default singleton
public class Student {
    @Autowired
    User user;

    public Student(){
        System.out.println("Initialization Student");
    }
    @PostConstruct
    public void init(){
        System.out.println("Student object @PostConstruct Student object hascode:
"+this.hashCode()+ " User Object Hascode: "+user.hashCode());
    }
}
```

```
package com.springProject.SpringCourseProject.controller;

import com.springProject.SpringCourseProject.component.Student;
import com.springProject.SpringCourseProject.component.User;
import jakarta.annotation.PostConstruct;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.config.ConfigurableBeanFactory;
import org.springframework.context.annotation.Scope;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping(value = "/api/")
@Scope(ConfigurableBeanFactory.SCOPE_PROTOTYPE)
public class TestController1 {

    @Autowired
    User user;

    @Autowired
    Student student;
    public TestController1(){
        System.out.println("TestController1 Instance Initialization");
    }

    @PostConstruct
    public void init(){
        System.out.println("TestController1 object hascode: "+this.hashCode()+" User object Hashcode: "+user.hashCode()+" Student object Hashcode: "+student.hashCode());
    }

    @GetMapping(path = "/fetchUser")
    public ResponseEntity<String> getUserDetails(){
        System.out.println("Fetch user api invoked");
        return ResponseEntity.status(HttpStatus.OK).body("");
    }
}
```

```
 :: Spring Boot ::                (v3.3.2)

2024-08-19T17:45:07.073+05:30  INFO 3540 --- [SpringCourseProject] [           main] c.s.S.SpringCourseProjectApplication     : Starting SpringCourseProjectApplication using Java 22.0.2 with PID 3540 (C:\volume e\java_projects\SpringCourseProject\target\classes started by DELL in C:\volume e\java_projects\SpringCourseProject)
2024-08-19T17:45:07.077+05:30  INFO 3540 --- [SpringCourseProject] [           main] c.s.S.SpringCourseProjectApplication     : No active profile set, falling back to 1 default profile: "default"
2024-08-19T17:45:09.058+05:30  INFO 3540 --- [SpringCourseProject] [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port 8080 (http)
2024-08-19T17:45:09.086+05:30  INFO 3540 --- [SpringCourseProject] [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2024-08-19T17:45:09.086+05:30  INFO 3540 --- [SpringCourseProject] [           main] o.apache.catalina.core.StandardEngine    : Starting Servlet engine: [Apache Tomcat/10.1.26]
2024-08-19T17:45:09.224+05:30  INFO 3540 --- [SpringCourseProject] [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2024-08-19T17:45:09.226+05:30  INFO 3540 --- [SpringCourseProject] [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 2051 ms
```

Initialization Student

Initialization User

```
User object @PostConstruct User object hascode: 1310167355
Student object @PostConstruct Student object hascode: 1063423209 User Object Hascode: 1310167355
2024-08-19T17:45:09.963+05:30  INFO 3540 --- [SpringCourseProject] [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port 8080 (http) with context path '/'
2024-08-19T17:45:09.978+05:30  INFO 3540 --- [SpringCourseProject] [           main] c.s.S.SpringCourseProjectApplication     : Started SpringCourseProjectApplication in 3.911 seconds (process running for 4.68)
2024-08-19T17:51:58.965+05:30  INFO 3540 --- [SpringCourseProject] [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring DispatcherServlet 'dispatcherServlet'
2024-08-19T17:51:58.966+05:30  INFO 3540 --- [SpringCourseProject] [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet        : Initializing Servlet 'dispatcherServlet'
2024-08-19T17:51:58.968+05:30  INFO 3540 --- [SpringCourseProject] [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet        : Completed initialization in 2 ms
TestController1 Instance Initialization
Initialization User
User object @PostConstruct User object hascode: 462113271
TestController1 object hascode: 1383760613 User object Hashcode: 462113271 Student object Hashcode: 1063423209
Fetch user api invoked
```

fetchuser api inviked for one more time

```
TestController1 Instance Initialization
Initialization User
User object @PostConstruct User object hascode: 925457676
TestController1 object hascode: 971945369 User object Hashcode: 925457676 Student object Hashcode: 1063423209
Fetch user api invoked
```

Request

- New Object is created for each **HTTP** request.
- Lazily initialized.

```
package com.springProject.SpringCourseProject.component;

import jakarta.annotation.PostConstruct;
import org.springframework.context.annotation.Scope;
import org.springframework.stereotype.Component;

@Component
@Scope("request")
public class User{
    public User(){
        System.out.println("Initialization User");
    }
    @PostConstruct
    public void init(){
        System.out.println("User object @PostConstruct User object hascode:
"+this.hashCode());
    }
}
```

```
package com.springProject.SpringCourseProject.component;

import jakarta.annotation.PostConstruct;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Scope;
import org.springframework.stereotype.Component;

@Component
@Scope("prototype")
public class Student {
    @Autowired
    User user;

    public Student(){
        System.out.println("Initialization Student");
    }
    @PostConstruct
    public void init(){
        System.out.println("Student object @PostConstruct Student object hascode:
"+this.hashCode()+ " User Object Hascode: "+user.hashCode());
    }
}
```

```
package com.springProject.SpringCourseProject.controller;

import com.springProject.SpringCourseProject.component.Student;
import com.springProject.SpringCourseProject.component.User;
import jakarta.annotation.PostConstruct;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.config.ConfigurableBeanFactory;
import org.springframework.context.annotation.Scope;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping(value = "/api/")
@Scope("request")
public class TestController1 {

    @Autowired
    User user;

    @Autowired
    Student student;

    public TestController1(){
        System.out.println("TestController1 Instance Initialization");
    }

    @PostConstruct
    public void init(){
        System.out.println("TestController1 object hascode: "+this.hashCode()+" User
object Hashcode: "+user.hashCode()+" Student object Hashcode: "+student.hashCode());
    }

    @GetMapping(path = "/fetchUser")
    public ResponseEntity<String> getUserDetails(){
        System.out.println("Fetch user api invoked");
        return ResponseEntity.status(HttpStatus.OK).body("");
    }
}
```

```
 :: Spring Boot ::                (v3.3.2)

2024-08-19T18:02:20.597+05:30  INFO 12236 --- [SpringCourseProject] [           main]
c.s.S.SpringCourseProjectApplication     : Starting SpringCourseProjectApplication using Java
22.0.2 with PID 12236 (C:\volume e\java_projects\SpringCourseProject\target\classes started by
DELL in C:\volume e\java_projects\SpringCourseProject)
2024-08-19T18:02:20.602+05:30  INFO 12236 --- [SpringCourseProject] [           main]
c.s.S.SpringCourseProjectApplication     : No active profile set, falling back to 1 default profile:
"default"
```

```
2024-08-19T18:02:22.569+05:30  INFO 12236 --- [SpringCourseProject] [           main]
o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port 8080 (http)
2024-08-19T18:02:22.596+05:30  INFO 12236 --- [SpringCourseProject] [           main]
o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2024-08-19T18:02:22.597+05:30  INFO 12236 --- [SpringCourseProject] [           main]
o.apache.catalina.core.StandardEngine    : Starting Servlet engine: [Apache Tomcat/10.1.26]
2024-08-19T18:02:22.748+05:30  INFO 12236 --- [SpringCourseProject] [           main] o.a.c.c.C.
[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2024-08-19T18:02:22.750+05:30  INFO 12236 --- [SpringCourseProject] [           main]
w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed
in 2056 ms
2024-08-19T18:02:23.519+05:30  INFO 12236 --- [SpringCourseProject] [           main]
o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port 8080 (http) with context
path '/'
2024-08-19T18:02:23.534+05:30  INFO 12236 --- [SpringCourseProject] [           main]
c.s.S.SpringCourseProjectApplication     : Started SpringCourseProjectApplication in 3.924 seconds
(process running for 4.686)
2024-08-19T18:04:06.487+05:30  INFO 12236 --- [SpringCourseProject] [nio-8080-exec-2] o.a.c.c.C.
[Tomcat].[localhost].[/]       : Initializing Spring DispatcherServlet 'dispatcherServlet'
2024-08-19T18:04:06.488+05:30  INFO 12236 --- [SpringCourseProject] [nio-8080-exec-2]
o.s.web.servlet.DispatcherServlet        : Initializing Servlet 'dispatcherServlet'
2024-08-19T18:04:06.491+05:30  INFO 12236 --- [SpringCourseProject] [nio-8080-exec-2]
o.s.web.servlet.DispatcherServlet        : Completed initialization in 3 ms
```

after hitting fetch User API

TestController1 Instance Initialization

Initialization User

User object @PostConstruct User object hascode: 613091292

Initialization Student

Student object @PostConstruct Student object hascode: 160541368 User Object Hascode: 613091292

TestController1 object hascode: 1450694144 User object Hashcode: 613091292 Student object Hashcode: 160541368

Fetch user api invoked

>**after hitting fetch User API for 2nd time**

TestController1 Instance Initialization

Initialization User

User object @PostConstruct User object hascode: 1763395825

Initialization Student

Student object @PostConstruct Student object hascode: 1742515106 User Object Hascode: 1763395825

TestController1 object hascode: 1675830145 User object Hashcode: 1763395825 Student object Hashcode: 1742515106

Fetch user api invoked

Consider below code what will happen

User marked scope as request

```
package com.springProject.SpringCourseProject.component;

import jakarta.annotation.PostConstruct;
import org.springframework.context.annotation.Scope;
import org.springframework.stereotype.Component;

@Component
@Scope("request")
public class User{
    public User(){
        System.out.println("Initialization User");
    }
    @PostConstruct
    public void init(){
        System.out.println("User object @PostConstruct User object hascode:
"+this.hashCode());
    }
}
```

Testcontroller1 Marked as as singleton

```
package com.springProject.SpringCourseProject.controller;

import com.springProject.SpringCourseProject.component.Student;
import com.springProject.SpringCourseProject.component.User;
import jakarta.annotation.PostConstruct;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.config.ConfigurableBeanFactory;
import org.springframework.context.annotation.Scope;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
```

```java
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping(value = "/api/")
@Scope("singleton")
public class TestController1 {

    @Autowired
    User user;

//    @Autowired
//    Student student;

    public TestController1(){
        System.out.println("TestController1 Instance Initialization");
    }

    @PostConstruct
    public void init(){
        System.out.println("TestController1 object hascode: "+this.hashCode()
                +" User object Hashcode: "+user.hashCode()
//                +" Student object Hashcode: "+student.hashCode()
        );
    }

    @GetMapping(path = "/fetchUser")
    public ResponseEntity<String> getUserDetails(){
        System.out.println("Fetch user api invoked");
        return ResponseEntity.status(HttpStatus.OK).body("");
    }
}
```

```
:: Spring Boot ::                (v3.3.2)

2024-08-19T18:17:56.858+05:30  INFO 7332 --- [SpringCourseProject] [           main] c.s.S.SpringCourseProjectApplication     : Starting SpringCourseProjectApplication using Java 22.0.2 with PID 7332 (C:\volume e\java_projects\SpringCourseProject\target\classes started by DELL in C:\volume e\java_projects\SpringCourseProject)
2024-08-19T18:17:56.863+05:30  INFO 7332 --- [SpringCourseProject] [           main] c.s.S.SpringCourseProjectApplication     : No active profile set, falling back to 1 default profile: "default"
```

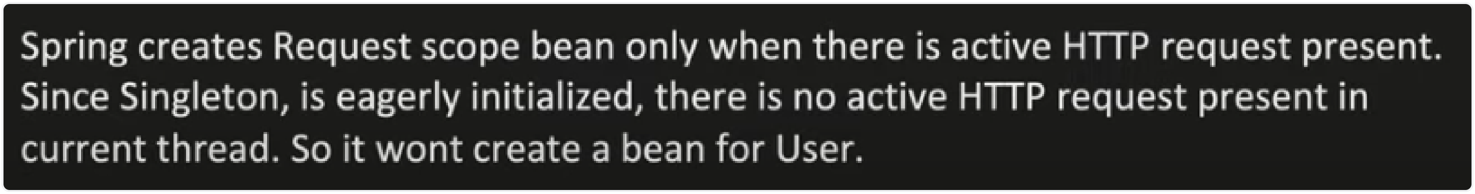
```
2024-08-19T18:17:58.749+05:30  INFO 7332 --- [SpringCourseProject] [           main]
o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port 8080 (http)
2024-08-19T18:17:58.777+05:30  INFO 7332 --- [SpringCourseProject] [           main]
o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2024-08-19T18:17:58.778+05:30  INFO 7332 --- [SpringCourseProject] [           main]
o.apache.catalina.core.StandardEngine    : Starting Servlet engine: [Apache Tomcat/10.1.26]
2024-08-19T18:17:58.911+05:30  INFO 7332 --- [SpringCourseProject] [           main] o.a.c.c.C.
[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2024-08-19T18:17:58.913+05:30  INFO 7332 --- [SpringCourseProject] [           main]
w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed
in 1953 ms
TestController1 Instance Initialization
2024-08-19T18:17:59.034+05:30  WARN 7332 --- [SpringCourseProject] [           main]
ConfigServletWebServerApplicationContext : Exception encountered during context initialization -
cancelling refresh attempt: org.springframework.beans.factory.UnsatisfiedDependencyException:
Error creating bean with name 'testController1': Unsatisfied dependency expressed through field
'user': Error creating bean with name 'user': Scope 'request' is not active for the current thread;
consider defining a scoped proxy for this bean if you intend to refer to it from a singleton
2024-08-19T18:17:59.042+05:30  INFO 7332 --- [SpringCourseProject] [           main]
o.apache.catalina.core.StandardService   : Stopping service [Tomcat]
2024-08-19T18:17:59.064+05:30  INFO 7332 --- [SpringCourseProject] [           main]
.s.b.a.l.ConditionEvaluationReportLogger :

Error starting ApplicationContext. To display the condition evaluation report re-run your
application with 'debug' enabled.
2024-08-19T18:17:59.126+05:30 ERROR 7332 --- [SpringCourseProject] [           main]
o.s.boot.SpringApplication               : Application run failed
```

Solution

- Testcontroller1 is singleton so, it is eagerly initialized at the time of application startup.
- But in this controller it depends on user which have scope of request at each http request object will going to created so that's why application failed while creating bean for User Object.

```
@Scope(value = "request",proxyMode = ScopedProxyMode.TARGET_CLASS)
```

- At the time of application startup a dummy object will created for User Component and injected into Testcontroller1 User Field.



Spring creates Request scope bean only when there is active HTTP request present. Since Singleton, is eagerly initialized, there is no active HTTP request present in current thread. So it wont create a bean for User.

```java
package com.springProject.SpringCourseProject.component;

import jakarta.annotation.PostConstruct;
import org.springframework.context.annotation.Scope;
import org.springframework.context.annotation.ScopedProxyMode;
import org.springframework.stereotype.Component;

@Component
@Scope(value = "request",proxyMode = ScopedProxyMode.TARGET_CLASS)
public class User{
    public User(){
        System.out.println("Initialization User");
    }
    @PostConstruct
    public void init(){
        System.out.println("User object @PostConstruct User object hascode:
"+this.hashCode());
    }
}
```

```java
package com.springProject.SpringCourseProject.controller;

import com.springProject.SpringCourseProject.component.Student;
import com.springProject.SpringCourseProject.component.User;
import jakarta.annotation.PostConstruct;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.config.ConfigurableBeanFactory;
import org.springframework.context.annotation.Scope;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping(value = "/api/")
@Scope("singleton")
```

```
public class TestController1 {

    @Autowired
    User user;

//    @Autowired
//    Student student;

    public TestController1(){
        System.out.println("TestController1 Instance Initialization");
    }

    @PostConstruct
    public void init(){
        System.out.println("TestController1 object hascode: "+this.hashCode()
                +" User object Hashcode: "+user.hashCode()
//                +" Student object Hashcode: "+student.hashCode()
        );
    }

    @GetMapping(path = "/fetchUser")
    public ResponseEntity<String> getUserDetails(){
        System.out.println("Fetch user api invoked");
        return ResponseEntity.status(HttpStatus.OK).body(user.getUserDetails());
    }
}
```

```
TestController1 Instance Initialization
TestController1 object hascode: 2043702205 User object Hashcode: 1456762856

Fetch user api invoked
Initialization User
User object @PostConstruct User object hascode: 1099746262
```

Session:

- New Object is created for each **HTTP** session.
- Lazily initialized
- When user accesses any endpoint, session is created.
- Remains active ,till it does not expires.

TestController1 Scope marked as session

```
package com.springProject.SpringCourseProject.controller;

import com.springProject.SpringCourseProject.component.Student;
import com.springProject.SpringCourseProject.component.User;
import jakarta.annotation.PostConstruct;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.config.ConfigurableBeanFactory;
import org.springframework.context.annotation.Scope;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping(value = "/api/")
@Scope(value="session")
public class TestController1 {

    @Autowired
    User user;

//    @Autowired
//    Student student;

    public TestController1(){
        System.out.println("TestController1 Instance Initialization");
    }

    @PostConstruct
    public void init(){
        System.out.println("TestController1 object hascode: "+this.hashCode()
                +" User object Hashcode: "+user.hashCode()
//                +" Student object Hashcode: "+student.hashCode()
        );
    }

    @GetMapping(path = "/fetchUser")
    public ResponseEntity<String> getUserDetails(){
        System.out.println("Fetch user api invoked");
        return ResponseEntity.status(HttpStatus.OK).body(user.getUserDetails());
    }
}
```

User Scope marked as singleton

```
package com.springProject.SpringCourseProject.component;

import jakarta.annotation.PostConstruct;
import org.springframework.context.annotation.Scope;
import org.springframework.context.annotation.ScopedProxyMode;
import org.springframework.stereotype.Component;

@Component
//@Scope(value = "request",proxyMode = ScopedProxyMode.TARGET_CLASS)
public class User{
    public User(){
        System.out.println("Initialization User");
    }
    @PostConstruct
    public void init(){
        System.out.println("User object @PostConstruct User object hascode: 
"+this.hashCode());
    }

    public String getUserDetails(){
        return "Hey! This Is Saurav";
    }
}
```

```
 :: Spring Boot ::                (v3.3.2)

2024-08-19T18:50:47.661+05:30  INFO 15064 --- [SpringCourseProject] [           main] c.s.S.SpringCourseProjectApplication     : Starting SpringCourseProjectApplication using Java 22.0.2 with PID 15064 (C:\volume e\java_projects\SpringCourseProject\target\classes started by DELL in C:\volume e\java_projects\SpringCourseProject)
2024-08-19T18:50:47.666+05:30  INFO 15064 --- [SpringCourseProject] [           main] c.s.S.SpringCourseProjectApplication     : No active profile set, falling back to 1 default profile: "default"
2024-08-19T18:50:49.676+05:30  INFO 15064 --- [SpringCourseProject] [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port 8080 (http)
2024-08-19T18:50:49.704+05:30  INFO 15064 --- [SpringCourseProject] [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2024-08-19T18:50:49.705+05:30  INFO 15064 --- [SpringCourseProject] [           main] o.apache.catalina.core.StandardEngine    : Starting Servlet engine: [Apache Tomcat/10.1.26]
2024-08-19T18:50:49.844+05:30  INFO 15064 --- [SpringCourseProject] [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2024-08-19T18:50:49.847+05:30  INFO 15064 --- [SpringCourseProject] [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 2090 ms
```

Initialization User

User object @PostConstruct User object hascode: 1310167355

```
2024-08-19T18:50:50.563+05:30  INFO 15064 --- [SpringCourseProject] [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port 8080 (http) with context path '/'
2024-08-19T18:50:50.578+05:30  INFO 15064 --- [SpringCourseProject] [           main] c.s.S.SpringCourseProjectApplication     : Started SpringCourseProjectApplication in 3.897 seconds (process running for 4.668)
```

1.after hitting the API

http://localhost:8080/api/fetchUser

TestController1 Instance Initialization

TestController1 object hascode: 753853159 User object Hashcode: 1310167355

Fetch user api invoked

2.after hitting the API

http://localhost:8080/api/fetchUser

Fetch user api invoked

- Here what happens http session is still alive it's just new HTTP Request so no object creation **till active HTTP SESSION**

http://localhost:8080/api/logout

End the session

3.after hitting the API

http://localhost:8080/api/fetchUser

TestController1 Instance Initialization

TestController1 object hascode: 33853603 User object Hashcode: 1310167355

Fetch user api invoke