WEEK – 4

Create a Spring Web Project using Maven


```java
1 package com.cognizant.demo;
2
3 import org.springframework.boot.SpringApplication;
4
5
6 @SpringBootApplication
7 public class SpringLearnApplication {
8
9     public static void main(String[] args) {
10         SpringApplication.run(SpringLearnApplication.class, args);
11     }
12
13 }
14
```



```xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
4     <modelVersion>4.0.0</modelVersion>
5     <parent>
6         <groupId>org.springframework.boot</groupId>
7         <artifactId>spring-boot-starter-parent</artifactId>
8         <version>3.5.3</version>
9         <relativePath/> <!-- lookup parent from repository -->
10     </parent>
11     <groupId>com.cognizant</groupId>
12     <artifactId>demo</artifactId>
13     <version>0.0.1-SNAPSHOT</version>
14     <name>spring-learn</name>
15     <description>Demo project for Spring Boot</description>
16     <url/>
17     <licenses>
18         <license/>
19     </licenses>
20     <developers>
21         <developer/>
22     </developers>
23     <scm>
24         <connection/>
25         <developerConnection/>
26         <tag/>
27         <url/>
28     </scm>
29     <properties>
30         <java.version>21</java.version>
31     </properties>
32     <dependencies>
33         <dependency>
34             <groupId>org.springframework.boot</groupId>
35             <artifactId>spring-boot-starter-web</artifactId>
36         </dependency>
37
38         <dependency>
39             <groupId>org.springframework.boot</groupId>
40             <artifactId>spring-boot-devtools</artifactId>
41             <scope>runtime</scope>
42             <optional>true</optional>
43         </dependency>
44         <dependency>
```

OUTPUT


```
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/

 :: Spring Boot ::                (v3.5.3)

2025-07-13T18:03:48.835+05:30  INFO 6496 --- [spring-learn] [   restartedMain] c.cognizant.demo.SpringLearnApplication   : Starting SpringLearnApp
2025-07-13T18:03:48.858+05:30  INFO 6496 --- [spring-learn] [   restartedMain] c.cognizant.demo.SpringLearnApplication   : No active profile set,
2025-07-13T18:03:48.981+05:30  INFO 6496 --- [spring-learn] [   restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : Devtools property defa
2025-07-13T18:03:48.982+05:30  INFO 6496 --- [spring-learn] [   restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : For additional web rel
2025-07-13T18:03:51.256+05:30  INFO 6496 --- [spring-learn] [   restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer   : Tomcat initialized wit
2025-07-13T18:03:51.292+05:30  INFO 6496 --- [spring-learn] [   restartedMain] o.apache.catalina.core.StandardService    : Starting service [Tomc
2025-07-13T18:03:51.293+05:30  INFO 6496 --- [spring-learn] [   restartedMain] o.apache.catalina.core.StandardEngine     : Starting Servlet engin
2025-07-13T18:03:51.393+05:30  INFO 6496 --- [spring-learn] [   restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/]        : Initializing Spring emb
2025-07-13T18:03:51.395+05:30  INFO 6496 --- [spring-learn] [   restartedMain] w.s.c.ServletWebServerApplicationContext  : Root WebApplicationCon
2025-07-13T18:03:52.225+05:30  INFO 6496 --- [spring-learn] [   restartedMain] o.s.b.d.a.OptionalLiveReloadServer        : LiveReload server is r
2025-07-13T18:03:52.295+05:30  INFO 6496 --- [spring-learn] [   restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer   : Tomcat started on port
2025-07-13T18:03:52.315+05:30  INFO 6496 --- [spring-learn] [   restartedMain] c.cognizant.demo.SpringLearnApplication   : Started SpringLearnApp
```
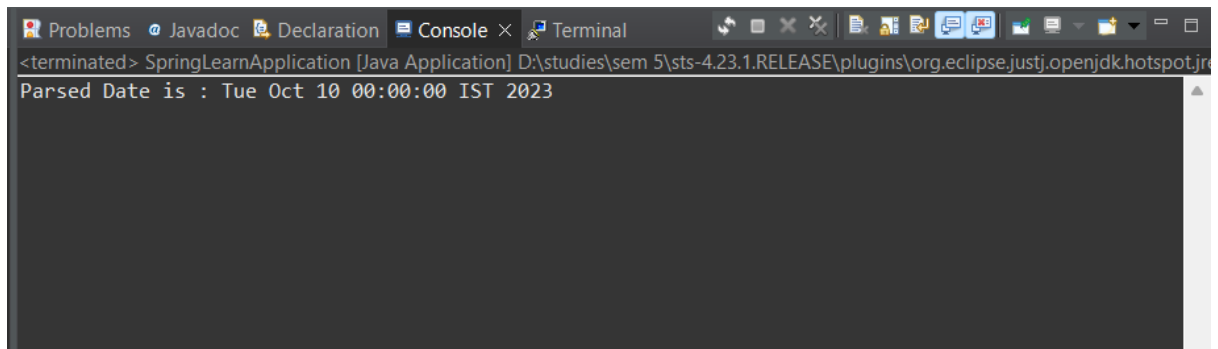
Spring Core – Load SimpleDateFormat from Spring Configuration XML

```xml
 1 <?xml version="1.0" encoding="UTF-8"?>
 2 <beans xmlns="http://www.springframework.org/schema/beans"
 3        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 4        xsi:schemaLocation="
 5         http://www.springframework.org/schema/beans
 6         https://www.springframework.org/schema/beans/spring-beans.xsd">
 7
 8     <bean id="dateFormat" class="java.text.SimpleDateFormat">
 9         <constructor-arg value="dd/MM/yyyy" />
10     </bean>
11
12 </beans>
13
```

```java
 1 package com.cognizant.demo;
 2
 3 import java.text.SimpleDateFormat;
 4 import java.util.Date;
 5
 6 import org.springframework.context.ApplicationContext;
 7 import org.springframework.context.support.ClassPathXmlApplicationContext;
 8
 9 public class SpringLearnApplication {
10
11     public static void main(String[] args) {
12         displayDate(); // calling the method
13     }
14
15     public static void displayDate() {
16         ApplicationContext context = new ClassPathXmlApplicationContext("date-format.xml");
17         SimpleDateFormat format = context.getBean("dateFormat", SimpleDateFormat.class);
18
19         try {
20             Date date = format.parse("10/10/2023");
21             System.out.println("Parsed Date is : " + date);
22         } catch (Exception e) {
23             e.printStackTrace();
24         }
25     }
26 }
27
```

OUTPUT

Console output:

```
<terminated> SpringLearnApplication [Java Application] D:\studies\sem 5\sts-4.23.1.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre
Parsed Date is : Tue Oct 10 00:00:00 IST 2023
```

Hello World RESTful Web Service

```java
application.properties    HelloController.java    SpringLearnApplication.java ×    spring-learn/pom.xml
1  package com.cognizant.springlearn;
2
3  import org.springframework.boot.SpringApplication;
4  import org.springframework.boot.autoconfigure.SpringBootApplication;
5
6  @SpringBootApplication
7  public class SpringLearnApplication {
8
9      public static void main(String[] args) {
10         SpringApplication.run(SpringLearnApplication.class, args);
11     }
12 }
13
```

```java
application.properties    HelloController.java ×    SpringLearnApplication.java    spring-learn/pom.xml
1  package com.cognizant.springlearn.controller;
2
3  import org.slf4j.Logger;
7
8  @RestController
9  public class HelloController {
10
11     private static final Logger LOGGER = LoggerFactory.getLogger(HelloController.class);
12
13     @GetMapping("/hello")
14     public String sayHello() {
15         LOGGER.info("START sayHello()");
16         String message = "Hello World!!";
17         LOGGER.info("END sayHello()");
18         return message;
19     }
20 }
21
```
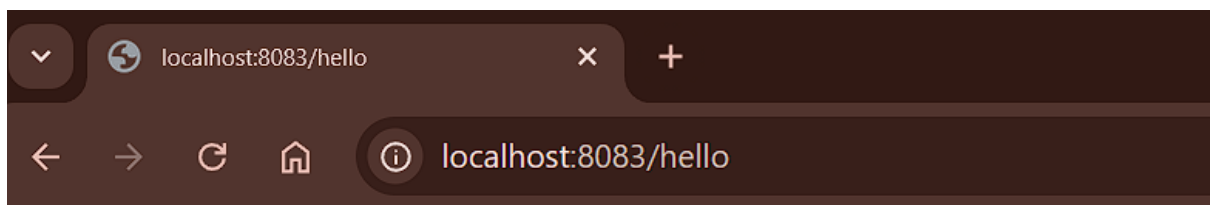
```properties
application.properties ×    HelloController.java    SpringLearnApplication.java    spring-learn/pom.xml
1  spring.application.name=spring-learn
2  server.port=8083
3
```

```
.   ____          _            __ _ _
/\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
\\/  ___)| |_)| | | | | || (_| |  ) ) ) )
'  |____| .__|_| |_|_| |_\__, | / / / /
=========|_|==============|___/=/_/_/_/

 :: Spring Boot ::                (v3.5.3)

2025-07-13T18:40:26.514+05:30  INFO 4248 --- [spring-learn] [  restartedMain] c.c.springlearn.SpringLearnApplication    : Starting SpringLearnApplica
2025-07-13T18:40:26.529+05:30  INFO 4248 --- [spring-learn] [  restartedMain] c.c.springlearn.SpringLearnApplication    : No active profile set, fall
2025-07-13T18:40:26.621+05:30  INFO 4248 --- [spring-learn] [  restartedMain] .e.DevToolsPropertyDefaultsPostProcessor  : Devtools property defaults
2025-07-13T18:40:26.621+05:30  INFO 4248 --- [spring-learn] [  restartedMain] .e.DevToolsPropertyDefaultsPostProcessor  : For additional web related
2025-07-13T18:40:28.205+05:30  INFO 4248 --- [spring-learn] [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer   : Tomcat initialized with por
2025-07-13T18:40:28.253+05:30  INFO 4248 --- [spring-learn] [  restartedMain] o.apache.catalina.core.StandardService    : Starting service [Tomcat]
2025-07-13T18:40:28.254+05:30  INFO 4248 --- [spring-learn] [  restartedMain] o.apache.catalina.core.StandardEngine     : Starting Servlet engine: [A
2025-07-13T18:40:28.350+05:30  INFO 4248 --- [spring-learn] [  restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/]        : Initializing Spring embedde
2025-07-13T18:40:28.351+05:30  INFO 4248 --- [spring-learn] [  restartedMain] w.s.c.ServletWebServerApplicationContext  : Root WebApplicationContext:
2025-07-13T18:40:28.988+05:30  INFO 4248 --- [spring-learn] [  restartedMain] o.s.b.d.a.OptionalLiveReloadServer        : LiveReload server is runnin
2025-07-13T18:40:29.063+05:30  INFO 4248 --- [spring-learn] [  restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer   : Tomcat started on port 8083
2025-07-13T18:40:29.078+05:30  INFO 4248 --- [spring-learn] [  restartedMain] c.c.springlearn.SpringLearnApplication    : Started SpringLearnApplicat
2025-07-13T18:40:40.220+05:30  INFO 4248 --- [spring-learn] [nio-8083-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/]        : Initializing Spring Dispatc
2025-07-13T18:40:40.220+05:30  INFO 4248 --- [spring-learn] [nio-8083-exec-1] o.s.web.servlet.DispatcherServlet         : Initializing Servlet 'dispa
2025-07-13T18:40:40.225+05:30  INFO 4248 --- [spring-learn] [nio-8083-exec-1] o.s.web.servlet.DispatcherServlet         : Completed initialization in
2025-07-13T18:40:40.319+05:30  INFO 4248 --- [spring-learn] [nio-8083-exec-1] c.c.s.controller.HelloController          : START sayHello()
2025-07-13T18:40:40.321+05:30  INFO 4248 --- [spring-learn] [nio-8083-exec-1] c.c.s.controller.HelloController          : END sayHello()
```

OUTPUT

Hello World!!

REST - Country Web Service

```xml
<!-- http://www.springframework.org/schema/beans/spring-beans.xsd (xsi:schemaLocation) -->
1  <!-- src/main/resources/country.xml -->
2  <beans xmlns="http://www.springframework.org/schema/beans"
3         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4         xsi:schemaLocation="
5             http://www.springframework.org/schema/beans
6             http://www.springframework.org/schema/beans/spring-beans.xsd">
7
8      <bean id="in" class="com.cognizant.springlearn.model.Country">
9          <property name="code" value="IN"/>
10         <property name="name" value="India"/>
11     </bean>
12
13 </beans>
14
```

```java
1  package com.cognizant.springlearn.model;
2
3  public class Country {
4      private String code;
5      private String name;
6
7      // Getters and setters
8      public String getCode() { return code; }
9      public void setCode(String code) { this.code = code; }
10
11     public String getName() { return name; }
12     public void setName(String name) { this.name = name; }
13 }
14
```

```java
1
2
3  package com.cognizant.springlearn.controller;
4
5  import com.cognizant.springlearn.model.Country;
6  import org.springframework.context.ApplicationContext;
7  import org.springframework.context.support.ClassPathXmlApplicationContext;
8  import org.springframework.web.bind.annotation.RequestMapping;
9  import org.springframework.web.bind.annotation.RestController;
10
11 @RestController
12 public class CountryController {
13
14     @RequestMapping("/country")
15     public Country getCountryIndia() {
16         ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");
17         return (Country) context.getBean("in");
18     }
19 }
20
```

OUTPUT

localhost:8083/country

localhost:8083/country

Pretty-print ☐

{"code":"IN","name":"India"}

REST - Get country based on country code

country.xml × Country.java   CountryController.java   CountryService.java

```xml
          http://www.springframework.org/schema/beans/spring-beans.xsd (xsi:schemaLocation)
 1 <beans xmlns="http://www.springframework.org/schema/beans"
 2        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 3        xsi:schemaLocation="
 4           http://www.springframework.org/schema/beans
 5           http://www.springframework.org/schema/beans/spring-beans.xsd">
 6
 7     <bean id="countryList" class="java.util.ArrayList">
 8         <constructor-arg>
 9             <list>
10                 <bean class="com.cognizant.springlearn.model.Country">
11                     <property name="code" value="IN"/>
12                     <property name="name" value="India"/>
13                 </bean>
14                 <bean class="com.cognizant.springlearn.model.Country">
15                     <property name="code" value="US"/>
16                     <property name="name" value="United States"/>
17                 </bean>
18                 <bean class="com.cognizant.springlearn.model.Country">
19                     <property name="code" value="CN"/>
20                     <property name="name" value="China"/>
21                 </bean>
22             </list>
23         </constructor-arg>
24     </bean>
25
26 </beans>
27
```

country.xml   Country.java × CountryController.java   CountryService.java

```java
 1 package com.cognizant.springlearn.model;
 2
 3 public class Country {
 4     private String code;
 5     private String name;
 6
 7     // Getters and setters
 8     public String getCode() { return code; }
 9     public void setCode(String code) { this.code = code; }
10
11     public String getName() { return name; }
12     public void setName(String name) { this.name = name; }
13 }
14
```

country.xml   Country.java   CountryController.java × CountryService.java

```java
 1 package com.cognizant.springlearn.controller;
 2
 3 import com.cognizant.springlearn.model.Country;
 4 import com.cognizant.springlearn.service.CountryService;
 5 import org.springframework.beans.factory.annotation.Autowired;
 6 import org.springframework.web.bind.annotation.*;
 7
 8 @RestController
 9 public class CountryController {
10
11     @Autowired
12     private CountryService countryService;
13
14     @GetMapping("/countries/{code}")
15     public Country getCountry(@PathVariable String code) {
16         return countryService.getCountry(code);
17     }
18 }
19
```

```java
package com.cognizant.springlearn.service;

import com.cognizant.springlearn.model.Country;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public class CountryService {

    public Country getCountry(String code) {
        ApplicationContext context = new ClassPathXmlApplicationContext("country.xml");
        List<Country> countryList = (List<Country>) context.getBean("countryList");

        return countryList.stream()
                .filter(c -> c.getCode().equalsIgnoreCase(code))
                .findFirst()
                .orElse(null); // or throw exception if preferred
    }
}
```


Output


```
{"code":"IN","name":"India"}
```

Create authentication service that returns JWT

```java
package jwt_auth_service.jwt_auth_service.config;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import org.springframework.security.web.SecurityFilterChain;

@Configuration
@EnableWebSecurity
public class SecurityConfig {

    @Bean
    public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {
        http
                .csrf(csrf -> csrf.disable())
                .authorizeHttpRequests(auth -> auth.anyRequest().permitAll());
        return http.build();
    }
}
```



```java
package jwt_auth_service.jwt_auth_service.util;

import io.jsonwebtoken.Jwts;
import io.jsonwebtoken.SignatureAlgorithm;
import io.jsonwebtoken.security.Keys;

import javax.crypto.SecretKey;
import java.util.Date;

public class JwtUtil {

    private static final SecretKey SECRET_KEY = Keys.secretKeyFor(SignatureAlgorithm.HS256);
    private static final long EXPIRATION_TIME = 600_000;

    public static String createJWT(String username) {
        return Jwts.builder()
                .setSubject(username)
                .setIssuedAt(new Date())
                .setExpiration(new Date(System.currentTimeMillis() + EXPIRATION_TIME))
                .signWith(SECRET_KEY)
                .compact();
    }
}
```



```
spring.application.name=jwt-auth-service
server.port=8091
logging.level.org.springframework.security=DEBUG
```

```
JS jwt-auth-service    Version control                                    JwtAuthServiceApplication

Project                          application.properties      pom.xml (jwt-auth-service)      SecurityConfig.java      FilterChainProxy.class      JwtR

  jwt_auth_service.jwt_auth_service     2   <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    config                              32   <dependencies>
      SecurityConfig                    33       <dependency>
    controller                          34           <groupId>org.springframework.boot</groupId>
      AuthController                    35           <artifactId>spring-boot-starter-security</artifactId>
    model                               36       </dependency>
      JwtResponse                       37       <dependency>
    service                             38           <groupId>org.springframework.boot</groupId>
      JwtService                        39           <artifactId>spring-boot-starter-web</artifactId>
    util                                40       </dependency>
      JwtUtil                           41
    JwtAuthServiceApplication           42       <dependency>
  resources                             43           <groupId>io.jsonwebtoken</groupId>
    static                              44           <artifactId>jjwt-api</artifactId>
    templates                           45           <version>0.11.5</version>
    application.properties              46       </dependency>
  test                                  47       <dependency>
  target                                48           <groupId>io.jsonwebtoken</groupId>
  .gitattributes                        49           <artifactId>jjwt-impl</artifactId>
  .gitignore                            50           <version>0.11.5</version>
  HELP.md                               51           <scope>runtime</scope>
  mvnw                                  52       </dependency>
  mvnw.cmd                              53       <dependency>
  pom.xml                               54           <groupId>io.jsonwebtoken</groupId>
  External Libraries                    55           <artifactId>jjwt-jackson</artifactId>
  Scratches and Consoles                56           <version>0.11.5</version>
                                        57           <scope>runtime</scope>
                                        58       </dependency>

Run    JwtAuthServiceApplication       project > licenses > license
```

```
2025-07-13T14:45:46.172+05:30  INFO 27500 --- [jwt-auth-service] [   restartedMain] o.s.b.d.a.OptionalLiveReloadServer   : LiveReload server is running on port 35729
2025-07-13T14:45:46.223+05:30  INFO 27500 --- [jwt-auth-service] [   restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port 8091 (http) with
  context path '/'
2025-07-13T14:45:46.238+05:30  INFO 27500 --- [jwt-auth-service] [   restartedMain] j.j.JwtAuthServiceApplication        : Started JwtAuthServiceApplication in 3.501
  seconds (process running for 4.18)
2025-07-13T14:45:51.663+05:30  INFO 27500 --- [jwt-auth-service] [nio-8091-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/]   : Initializing Spring DispatcherServlet
  'dispatcherServlet'
2025-07-13T14:45:51.664+05:30  INFO 27500 --- [jwt-auth-service] [nio-8091-exec-1] o.s.web.servlet.DispatcherServlet    : Initializing Servlet 'dispatcherServlet'
2025-07-13T14:45:51.665+05:30  INFO 27500 --- [jwt-auth-service] [nio-8091-exec-1] o.s.web.servlet.DispatcherServlet    : Completed initialization in 1 ms
2025-07-13T14:45:51.690+05:30 DEBUG 27500 --- [jwt-auth-service] [nio-8091-exec-1] o.s.security.web.FilterChainProxy    : Securing GET /authenticate
2025-07-13T14:45:51.700+05:30 DEBUG 27500 --- [jwt-auth-service] [nio-8091-exec-1] o.s.security.web.FilterChainProxy    : Secured GET /authenticate
Username: user
Password: pwd
2025-07-13T14:45:51.836+05:30 DEBUG 27500 --- [jwt-auth-service] [nio-8091-exec-1] o.s.s.w.a.AnonymousAuthenticationFilter : Set SecurityContextHolder to anonymous
  SecurityContext
```

```
Terminal    Command Prompt    Command Prompt (2)    +

C:\Users\22itb\OneDrive\Desktop\Week - 2\Week 4\jwt-auth-service\jwt-auth-service>curl -u user:pwd http://localhost:8091/authenticate
{"token":"eyJhbGci0iJIUzI1NiJ9.eyJzdWIi0iJ1c2VyIiwiaWF0IjoxNzUyMzk4Njk2LCJleHAi0jE3NTIz0Tky0TZ9.N2eQIKGlpMk4qgzUja4UCBmJHttxOsVdd3rZhE2Q_vw"}
C:\Users\22itb\OneDrive\Desktop\Week - 2\Week 4\jwt-auth-service\jwt-auth-service>
```

Overview   GET http://localhost:8091/au ● +

No environment ∨

http://localhost:8091/authenticate

Save ∨   Share

GET ∨   http://localhost:8091/authenticate

Send ∨

Params   Authorization ●   Headers (10)   Body ●   Scripts   Tests   Settings

Cookies

Auth Type

Basic Auth ∨

Username   user

Password   •••   👁 🔒

The authorization header will be automatically generated
when you send the request. Learn more about Basic Auth
authorization.

Body   Cookies   Headers (11)   Test Results   ⟳

200 OK · 12 ms · 476 B · ⊕ · ⋯

Raw ∨   ▷ Preview   Visualize ∨

1   {"token":"eyJhbGci0iJIUzI1NiJ9.eyJzdWIi0iJ1c2VyIiwiaWF0IjoxNzUyMzk3NzUwLCJleHAi0jE3NTIzOTgzNTB9.YKUEKQEw6NJsH1KW6ysm4vaM6iirs269fjCfy76UMq4"}