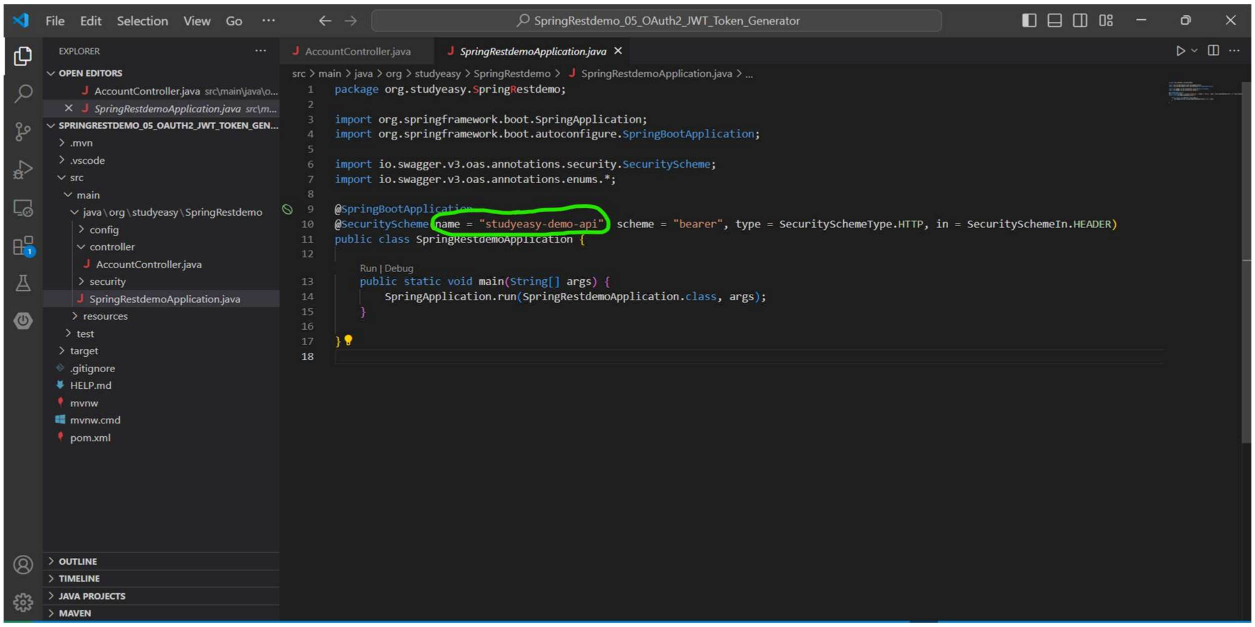
Java Spring Boot Restful – OAuth2 – JWT – Token Generator

How to secure?

```
@SecurityRequirement(name = "studyeasy-demo-api")
```


File Edit Selection View Go ···
SpringRestdemo_05_OAuth2_JWT_Token_Generator
EXPLORER
OPEN EDITORS
AccountController.java src\main\java\o...
SpringRestdemoApplication.java src\m...
SPRINGRESTDEMO_05_OAUTH2_JWT_TOKEN_GEN...
.mvn
.vscode
src
main
java \ org \ studyeasy \ SpringRestdemo
config
controller
AccountController.java
security
SpringRestdemoApplication.java
resources
test
target
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
OUTLINE
TIMELINE
JAVA PROJECTS
MAVEN
AccountController.java
SpringRestdemoApplication.java
src > main > java > org > studyeasy > SpringRestdemo > SpringRestdemoApplication.java > ...
Run | Debug

```
package org.studyeasy.SpringRestdemo;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

import io.swagger.v3.oas.annotations.security.SecurityScheme;
import io.swagger.v3.oas.annotations.enums.*;

@SpringBootApplication
@SecurityScheme(name = "studyeasy-demo-api", scheme = "bearer", type = SecuritySchemeType.HTTP, in = SecuritySchemeIn.HEADER)
public class SpringRestdemoApplication {

    public static void main(String[] args) {
        SpringApplication.run(SpringRestdemoApplication.class, args);
    }

}
```

Go to AccountController -> modify code

```
@GetMapping("/test")
    @SecurityRequirement(name = "studyeasy-demo-api")
    public String test(){
        return "Test api..";
    }
```

Tag – Add some description or name to a specific tag.

```
@Tag(name = "Test", description = "The Test API.")
```

```
@GetMapping("/test")
    @Tag(name = "Test", description = "The Test API.")
    @SecurityRequirement(name = "studyeasy-demo-api")
    public String test(){
        return "Test api..";
    }
```

Authentication Manager

```
@Bean
public AuthenticationManager authManager(UserDetailsService userDetailsService) {
       var authProvider = new DaoAuthenticationProvider();
       authProvider.setUserDetailsService(userDetailsService);
       return new ProviderManager(authProvider);
}
```

Create a controller in controller folder -> AuthController.java and in order to create a token you need to generate a service as well -> new folder -> service

New file -> TokenService.java

TokenService.java

```
package org.studyeasy.SpringRestdemo.service;

import java.time.Instant;
import java.time.temporal.ChronoUnit;
import java.util.stream.Collectors;

import org.springframework.security.core.Authentication;
import org.springframework.security.core.GrantedAuthority;
import org.springframework.security.oauth2.jwt.JwtClaimsSet;
import org.springframework.security.oauth2.jwt.JwtEncoder;
import org.springframework.security.oauth2.jwt.JwtEncoderParameters;


public class TokenService {

    private final JwtEncoder encoder;

    public TokenService(JwtEncoder encoder){
        this.encoder = encoder;
    }

    public String generateToken(Authentication authentication){
        Instant now = Instant.now();
        String scope = authentication.getAuthorities().stream()
                       .map(GrantedAuthority::getAuthority)
                       .collect(Collectors.joining(" "));
        JwtClaimsSet claims = JwtClaimsSet.builder()
                              .issuer("self")
                              .issuedAt(now)
                              .expiresAt(now.plus(1, ChronoUnit.HOURS))
                              .subject(authentication.getName())
                              .claim("scope", scope)
                              .build();
        return
this.encoder.encode(JwtEncoderParameters.from(claims)).getTokenValue();
    }
}
```

AuthController.java

```
package org.studyeasy.SpringRestdemo.controller;

import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class AuthController {

    @PostMapping("/token")
    public String token(){

    }
}
```