

## Getting Started with Auth Controller

Update Auth Controller

```
@RestController
public class AuthController {

    private final AuthenticationManager authenticationManager;
    private final TokenService tokenService;

    public AuthController(TokenService tokenService, AuthenticationManager authenticationManager) {
        this.tokenService = tokenService;
        this.authenticationManager = authenticationManager;
    }

    @PostMapping("/token")
    @ResponseBody
    public Token token(@RequestBody UserLogin userLogin) throws AuthenticationException {
        Authentication authentication = authenticationManager
            .authenticate(new
UsernamePasswordAuthenticationToken(userLogin.email(), userLogin.password()));
        return new Token(tokenService.generateToken(authentication));
    }
}
```

To

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

import
org.springframework.security.authentication.UsernamePasswordAuthenticationToken;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.security.authentication.AuthenticationManager;
import org.springframework.security.core.Authentication;
import org.springframework.security.core.AuthenticationException;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseBody;
import org.springframework.web.bind.annotation.ResponseStatus;
import org.springframework.web.bind.annotation.RestController;
```

```

import org.studyeasy.SpringRestdemo.payload.auth.Token;
import org.studyeasy.SpringRestdemo.payload.auth.UserLogin;
import org.studyeasy.SpringRestdemo.service.TokenService;

import io.swagger.v3.oas.annotations.tags.Tag;
import lombok.extern.slf4j.Slf4j;

@RestController
@RequestMapping("/auth")
@Tag(name= "Auth Controller", description="Controller for Account management")
@Slf4j
public class AuthController {

    private final AuthenticationManager authenticationManager;
    private final TokenService tokenService;

    public AuthController(TokenService tokenService, AuthenticationManager authenticationManager) {
        this.tokenService = tokenService;
        this.authenticationManager = authenticationManager;
    }

    @PostMapping("/token")
    @ResponseStatus(HttpStatus.OK)
    public ResponseEntity<Token> token(@RequestBody UserLogin userLogin) throws AuthenticationException {
        try{
            Authentication authentication = authenticationManager
                .authenticate(new
UsernamePasswordAuthenticationToken(userLogin.email(), userLogin.password()));
            return ResponseEntity.ok(new
Token(tokenService.generateToken(authentication)));
        }catch(Exception e){
            log.debug("null");
            return new ResponseEntity<>(new Token(null), HttpStatus.BAD_REQUEST);
        }

    }

}

```

Create a new folder -> util -> Create a new folder -> constants -> Create 2 enums files

## AccountSuccess.java

```
package org.studyeasy.SpringRestdemo.util.constants;

public enum AccountSuccess {
    ACCOUNT_ADDED
}
```

## AccountError.java

```
package org.studyeasy.SpringRestdemo.util.constants;

public enum AccountError {
    TOKEN_GENERATION_ERROR,
    ADD_ACCOUNT_ERROR
}
```

Update this line AuthController below line

```
log.debug("null");
to
```

```
log.debug(AccountError.TOKEN_GENERATION_ERROR.toString() + ": " +
e.getMessage());
```

Add this in application.properties

```
# Logging setting
logging.level.org.studyeasy=DEBUG
logging.pattern.console=%d [%level] %c{1.} [%t] %m%n
logging.file.name = appLog.log
logging.pattern.file=%d [%level] %c{1.} [%t] %m%n
```

Rename the file Token.java to TokenDTO.java and DTO means Data Transfer Object

Rename the file UserToken.java to UserToken DTO.java and DTO means Data Transfer Object

```
requestMatchers("/*").permitAll()
```

The screenshot shows the Swagger UI interface for a REST API. The top navigation bar includes tabs for "Course: FULL STACK JAVA DEV..." and "Swagger UI". The browser address bar displays "localhost:8080/swagger-ui/index.html#/Auth%20Controller/token".

### Responses

**Curl**

```
curl -X 'POST' \
'http://localhost:8080/auth/token' \
-H 'accept: */*' \
-H 'Content-Type: application/json' \
-d '{
  "email": "user@user.com",
  "password": "pass987"
}'
```

**Request URL**

```
http://localhost:8080/auth/token
```

**Server response**

Code	Details
200	<p><b>Response body</b></p> <pre>{   "token": "eyJraWQ1OjIyZmYzMDcyODMzMzc1LjQ3MmE1YmNmYXBvJmNkZWZlYVZlIiLCBjbGciOiJSU0UiLiNiIj39.eYpc3MI0I3Z26omIwiC3VIIjoIdXN1ckBic2VylmVnbS1mW4CTGHcyMzIZmK3OCwIamF8TjoxMzIZm3I3Mzc4LjC3YZ29wZSI0IjE3PTETVVFmFI39_k_gbrdeawt1oIIIVsgIvi_jnTKsy805AbhpMwaVIrShb5I3XMduelm_bdxRZ1DvftppjIUBcAgJk2IEZX4gPEESdLxwIITgtqc_7C3BTf6GFAlHtMeXdcjYbDbucITvcsc9ed0I1VCgZ7YFq88c2QD_Zsn-g_gzIca1b0TM5KzyP2XDQ2mfcbcs8In50vasifjGrW7a_ptICJgrzmno7P_PbasSDvrHrtcbdyAFPaqelQbnIP33wV13b3-9-7FHyeylKDplpnKJxdDVLipQyBCVgh2eozIFsmhgIvnXZuhkfgpl3X_hIqtBM1B6dyQDNw0Igs3-http" }</pre> <p><b>Response headers</b></p> <pre>cache-control: no-cache,no-store,max-age=0,must-revalidate connection: keep-alive content-type: application/json date: Fri,09 Aug 2024 18:16:18 GMT expires: 0 keep-alive: timeout=60 pragma: no-cache transfer-encoding: chunked vary: Origin,Access-Control-Request-Method,Access-Control-Request-Headers x-content-type-options: nosniff x-ssr-protection: 0</pre>

The bottom status bar shows the system clock at 11:47 PM on 8/9/2024.