

AlbumController – Build Add Album API

Create a DTO album in payload folder -> AlbumPayloadDTO.java

```
package org.studyeasy.SpringRestdemo.payload.auth.album;

import io.swagger.v3.oas.annotations.media.Schema;
import io.swagger.v3.oas.annotations.media.Schema.RequiredMode;
import jakarta.validation.constraints.NotBlank;
import lombok.AllArgsConstructor;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;
import lombok.ToString;

@Getter
@Setter
@ToString
@NoArgsConstructor
@AllArgsConstructor
public class AlbumPayloadDTO {

    @NotBlank
    @Schema(description = "Album name", example = "Travel", requiredMode =
RequiredMode.REQUIRED)
    private String name;

    @NotBlank
    @Schema(description = "Description for the Album", example = "description",
requiredMode = RequiredMode.REQUIRED)
    private String description;
}
```

Create a DTO album in payload folder -> AlbumViewDTO.java

```
package org.studyeasy.SpringRestdemo.payload.auth.album;

import io.swagger.v3.oas.annotations.media.Schema;
import io.swagger.v3.oas.annotations.media.Schema.RequiredMode;
import jakarta.validation.constraints.NotBlank;
import lombok.AllArgsConstructor;
import lombok.Getter;
import lombok.NoArgsConstructor;
```

```

import lombok.Setter;
import lombok.ToString;

@Getter
@Setter
@ToString
@NoArgsConstructor
@AllArgsConstructor
public class AlbumViewDTO {

    private long id;

    @NotBlank
    @Schema(description = "Album name", example = "Travel", requiredMode =
RequiredMode.REQUIRED)
    private String name;

    @NotBlank
    @Schema(description = "Description for the Album", example = "description",
requiredMode = RequiredMode.REQUIRED)
    private String description;
}

```

Create a enum in util folder -> AlbumSuccess.java and AlbumError.java

AlbumSuccess.java

```

package org.studyeasy.SpringRestdemo.util.constants;

public enum AlbumSuccess {
    ALUBM_ADDED
}

```

AlbumError.java

```

package org.studyeasy.SpringRestdemo.util.constants;

public enum AlbumError {
    ADD_ALBUM_ERROR
}

```

Updated AlbumController.java

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

import java.util.Optional;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.security.core.Authentication;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseStatus;
import org.springframework.web.bind.annotation.RestController;
import org.studyeasy.SpringRestdemo.model.Account;
import org.studyeasy.SpringRestdemo.model.Album;
import org.studyeasy.SpringRestdemo.payload.auth.album.AlbumPayloadDTO;
import org.studyeasy.SpringRestdemo.payload.auth.album.AlbumViewDTO;
import org.studyeasy.SpringRestdemo.service.AccountService;
import org.studyeasy.SpringRestdemo.service.AlbumService;
import org.studyeasy.SpringRestdemo.util.constants.AlbumError;

import io.swagger.v3.oas.annotations.Operation;
import io.swagger.v3.oas.annotations.parameters.RequestBody;
import io.swagger.v3.oas.annotations.responses.ApiResponse;
import io.swagger.v3.oas.annotations.security.SecurityRequirement;
import io.swagger.v3.oas.annotations.tags.Tag;
import jakarta.validation.Valid;
import lombok.extern.slf4j.Slf4j;

@RestController
@RequestMapping("api/v1/album")
@Tag(name = "Album Controller", description = "Controller for Album and Photo management")
@Slf4j
public class AlbumController {

    @Autowired
    private AccountService accountService;

    @Autowired
    private AlbumService albumService;

    @PostMapping(value = "/albums/add", consumes = "application/json", produces = "application/json")
```

```

        @ResponseStatus(HttpStatus.CREATED)
        @ApiResponse(responseCode = "400", description = "Please add valid name for
description")
        @ApiResponse(responseCode = "201", description = "Account added")
        @Operation(summary = "Add an album")
        @SecurityRequirement(name = "studyeasy-demo-api")
        public ResponseEntity<AlbumViewDTO> addAlbum(@Valid @RequestBody
AlbumPayloadDTO albumPayloadDTO,
            Authentication authentication) {
            try {
                Album album = new Album();
                album.setName(albumPayloadDTO.getName());
                album.setDescription(albumPayloadDTO.getDescription());

                // Extract Account
                String email = authentication.getName();

                Optional<Account> optionalAccount =
accountService.findByEmail(email);
                Account account = optionalAccount.get();
                album.setAccount(account);
                album = albumService.save(album);
                AlbumViewDTO albumViewDTO = new AlbumViewDTO(album.getId(),
album.getName(), album.getDescription());
                return ResponseEntity.ok(albumViewDTO);
            } catch (Exception e) {
                log.debug(AlbumError.ADD_ALBUM_ERROR.toString() + ": " +
e.getMessage());
                return ResponseEntity.status(HttpStatus.BAD_REQUEST).body(null);
            }
        }
    }
}

```

In SecurityConfig.java -> update below line

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

to

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

Updated application.properties

```

# Server port
Server.port=8080

# Database config
spring.datasource.url=jdbc:h2:file:./db/db
spring.datasource.driverClassName=org.h2.Driver
spring.datasource.username=admin
spring.datasource.password=pass987
spring.jpa.database-platform=org.hibernate.dialect.H2Dialect

# setup local h2 database console
spring.h2.console.enabled=true
spring.h2.console.path=/db-console
spring.h2.console.settings.web-allow-others=false
security.headers.frame=false

# local h2 tables setup on startup; use the value "none" in production!
spring.jpa.hibernate.ddl-auto=create-drop

# 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

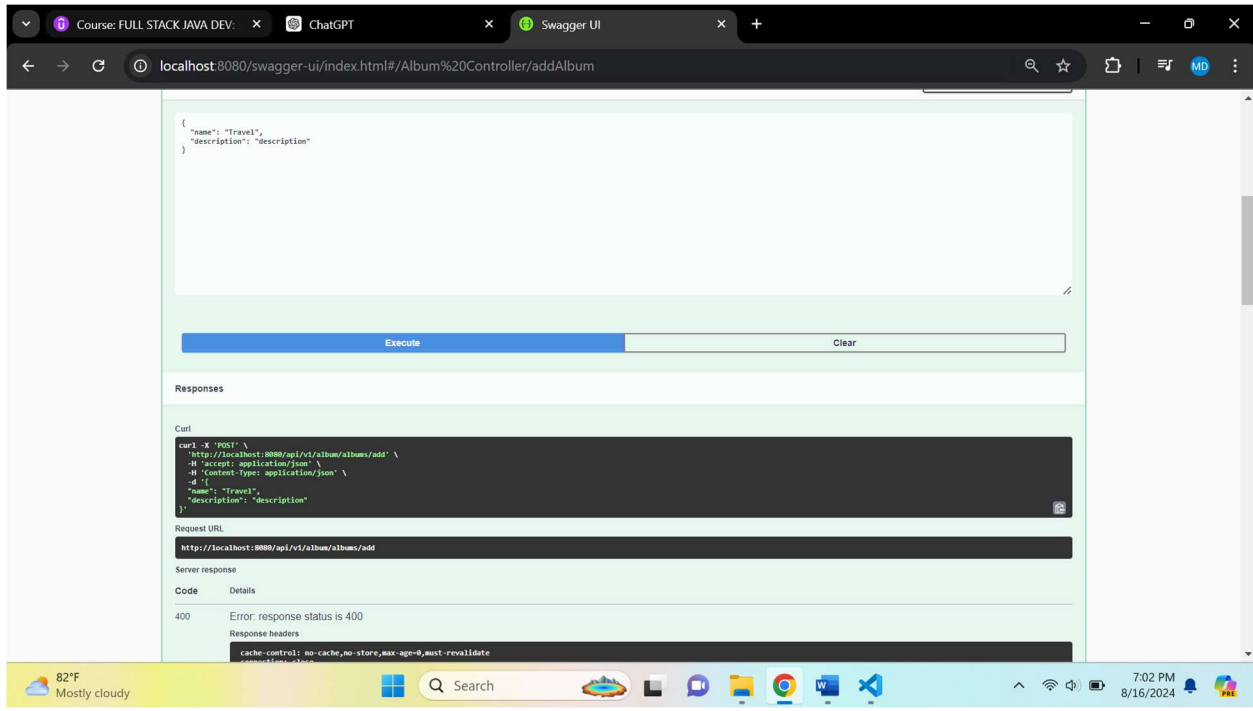
```

There was an error facing in controller - Wrong Import

```
import org.springframework.web.bind.annotation.RequestBody;
```

- **Incorrect Import:** In the latest version, the `@RequestBody` annotation is incorrectly imported from `io.swagger.v3.oas.annotations.parameters.RequestBody`. This is a Swagger annotation used for documentation purposes, not for binding HTTP request bodies to method parameters.
- **Correct Import:** The `@RequestBody` annotation should be imported from `org.springframework.web.bind.annotation.RequestBody`, as this is the one that binds the HTTP request body to the method parameter in Spring.

Output



Try to login by generating a token and authorize and then try to add album

Course: FULL STACK JAVA DEV: x ChatGPT x Swagger UI x +

localhost:8080/swagger-ui/index.html#/Album%20Controller/addAlbum

Album Controller

Controller for Album and Photo management

POST /api/v1/album/albums/add Add an Album

Parameters

No parameters

Request body **required** application/json

```
{
  "name": "Travel",
  "description": "Description"
}
```

Execute Clear

Responses

200 OK

```
{
  "id": "1",
  "name": "Travel",
  "description": "Description"
}
```

Request URL: http://localhost:8080/api/v1/album/albums/add

Server response

Code: 200

Response body

```
{
  "id": "1",
  "name": "Travel",
  "description": "Description"
}
```

add ed

82°F Mostly cloudy 7:04 PM 8/16/2024

Db console

Course: FULL STACK JAVA DEV: x ChatGPT x Swagger UI x H2 Console x +

localhost:8080/db-console/login.do?sessionId=8e1ecb0d6a0b0a797405e189c02e2612

Run Run Selected Auto complete Clear SQL statement:

SELECT * FROM ALBUM

SELECT * FROM ALBUM:

ACCOUNT_ID	ID	DESCRIPTION	NAME
1	1	description	Travel

(1 row, 5 ms)

Edit

82°F Mostly cloudy 7:05 PM 8/16/2024

