

java springboot restful 33 AlbumController – Upgrade View Albums API

Upgrading of listing all the albums API – How to get list of photos

Updated albums method

```
// List all the albums based on the logged in users
@GetMapping(value = "/albums", produces = "application/json")
@ApiResponse(responseCode = "200", description = "List of albums")
@ApiResponse(responseCode = "401", description = "Token missing")
@ApiResponse(responseCode = "403", description = "Token Error")
@Operation(summary = "List album api")
@SecurityRequirement(name = "studyeasy-demo-api")
public List<AlbumViewDTO> albums(Authentication authentication) {
    String email = authentication.getName();
    Optional<Account> optionalAccount = accountService.findByEmail(email);
    Account account = optionalAccount.get();
    List<AlbumViewDTO> albums = new ArrayList<>();
    for (Album album : albumService.findByAccount_id(account.getId())) {

        List<PhotoDTO> photos = new ArrayList<>();
        for(Photo photo : photoService.findByAlbumId(album.getId())){
            String link =
"/albums/"+album.getId()+"/photos/"+photo.getId()+"/download-photo";
            photos.add(new PhotoDTO(photo.getId(), photo.getName(),
photo.getDescription(), photo.getFileName(),link));
        }

        albums.add(new AlbumViewDTO(album.getId(), album.getName(),
album.getDescription(), photos));
    }
    return albums;
}
```

Updated AlbumViewDTO.java

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

import java.util.List;

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;

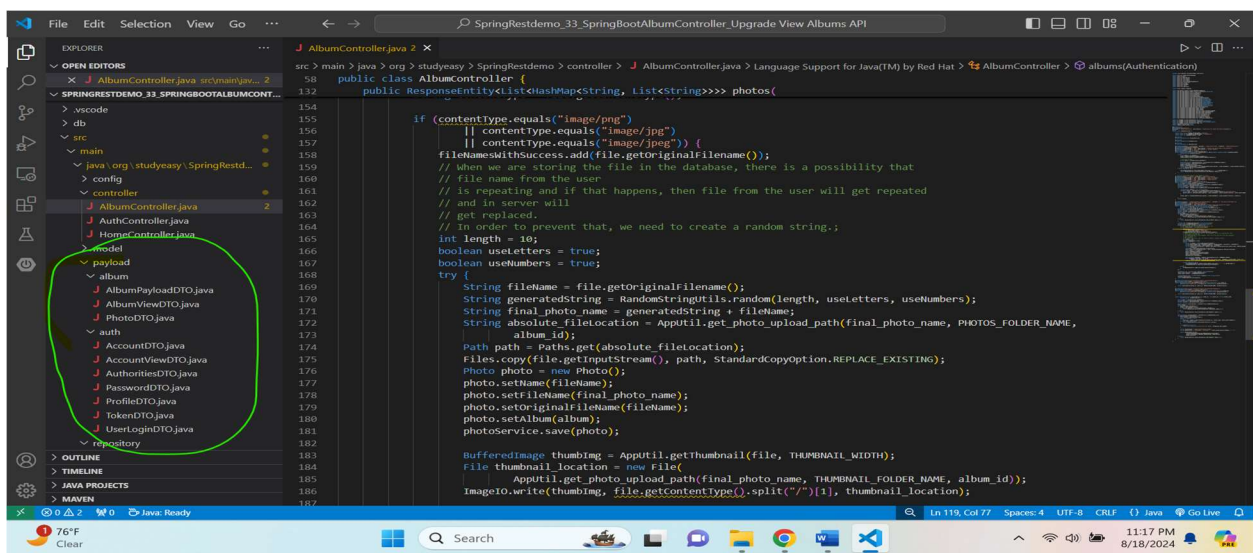
    @NotBlank
    @Schema(description = "Album name", example = "Travel", requiredMode =
RequiredMode.REQUIRED)
    private String name;

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

    private List<PhotoDTO> photos;
}

```

Change the Structure



Add a DTO in payload -> album -> PhotoDTO.java

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

import lombok.AllArgsConstructor;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;
import lombok.ToString;

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

    private long id;

    private String name;

    private String description;

    private String fileName;

    private String download_link;

}
```

Add below code in PhotoService.java

```
public List<Photo> findByAlbumId(long id){
    return photoRepository.findByAlbum_id(id);
}
```

Add this code in PhotoRepository.java

```
List<Photo> findByAlbum_id(long id);
```