Update Album API

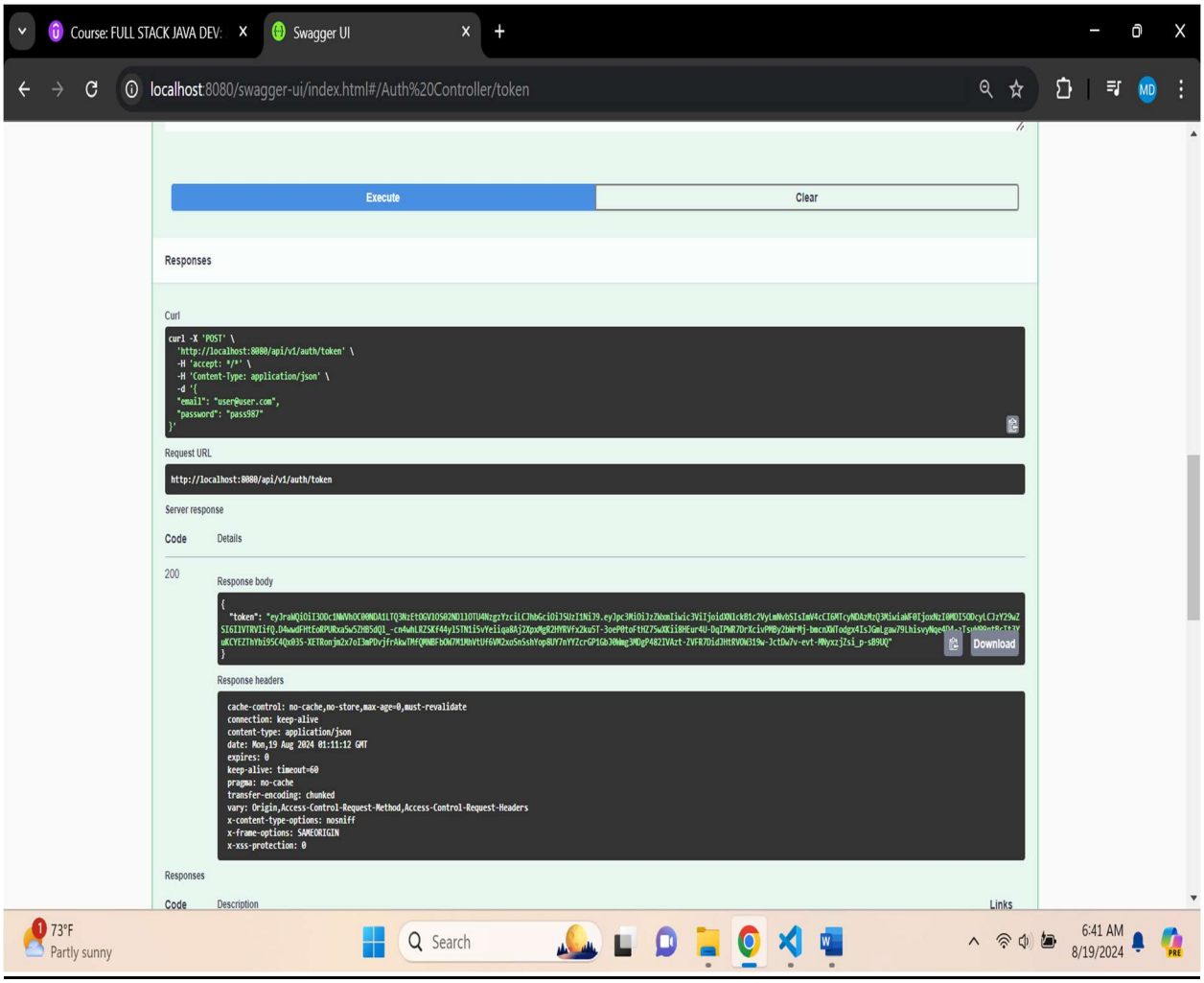
Add below code in AlbumController.java

```
//Update Album
    @PutMapping(value = "/albums/{album_id}/update", consumes =
"application/json", produces = "application/json")
    @ResponseStatus(HttpStatus.CREATED)
    @ApiResponse(responseCode = "400", description = "Please add valid name a
description")
    @ApiResponse(responseCode = "204", description = "Album update")
    @Operation(summary = "Update an Album")
    @SecurityRequirement(name = "studyeasy-demo-api")
    public ResponseEntity<AlbumViewDTO> update_Album(@Valid @RequestBody
AlbumPayloadDTO albumPayloadDTO,
            @PathVariable long album_id, Authentication authentication) {
        try {

            String email = authentication.getName();
            Optional<Account> optionalAccount =
accountService.findByEmail(email);
            Account account = optionalAccount.get();

            Optional<Album> optionaAlbum = albumService.findById(album_id);
            Album album;
            if (optionaAlbum.isPresent()) {
                album = optionaAlbum.get();
                if (account.getId() != album.getAccount().getId()) {
                    return
ResponseEntity.status(HttpStatus.FORBIDDEN).body(null);
                }
            } else {
                return ResponseEntity.status(HttpStatus.BAD_REQUEST).body(null);
            }


            album.setName(albumPayloadDTO.getName());
            album.setDescription(albumPayloadDTO.getDescription());
            album = albumService.save(album);
            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(),
```

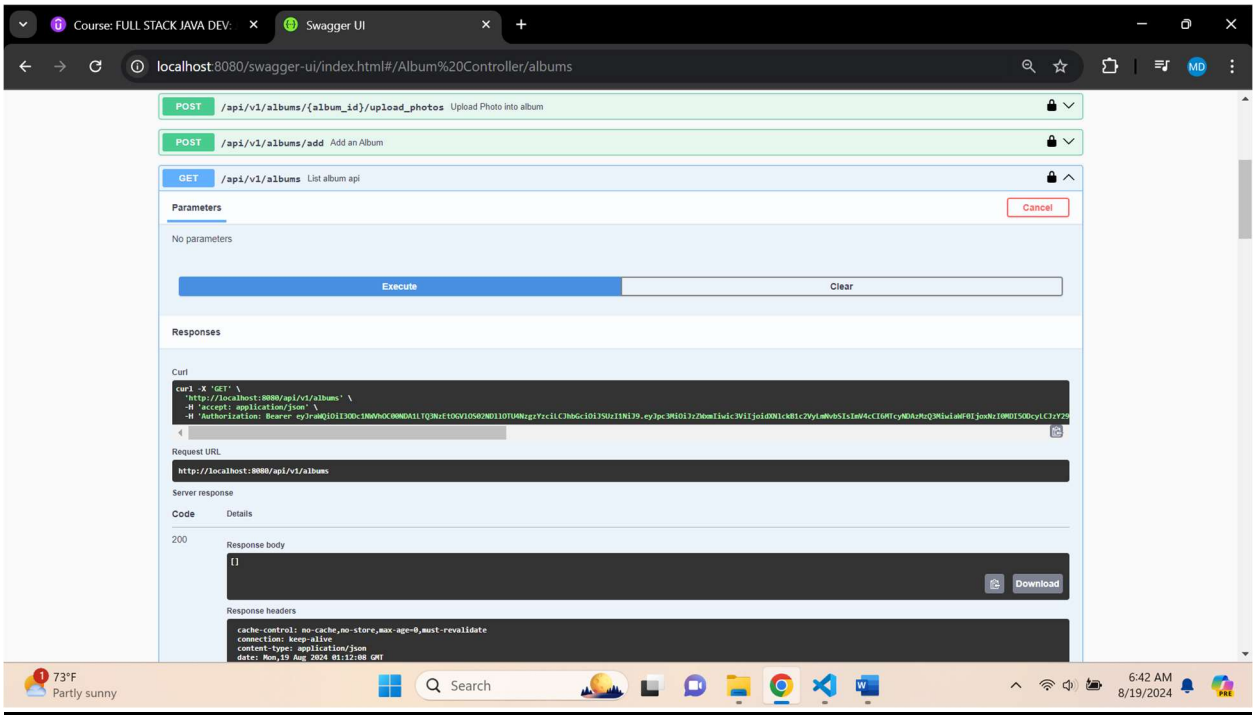
```
                photo.getFileName(), link));

            }
            AlbumViewDTO albumViewDTO = new AlbumViewDTO(album.getId(),
album.getName(), album.getDescription(), photos);
            return ResponseEntity.ok(albumViewDTO);

        } catch (Exception e) {
            return ResponseEntity.status(HttpStatus.BAD_REQUEST).body(null);
        }
    }
```

Output

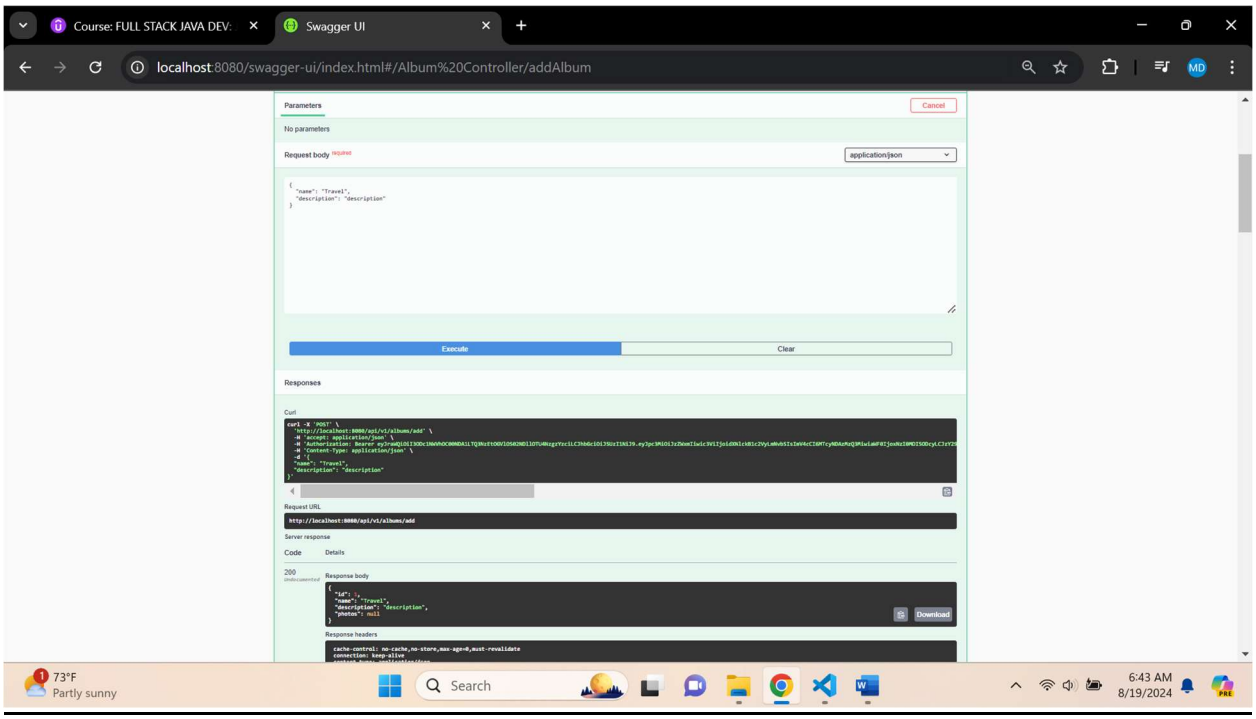
Token Generated to authrozie for user



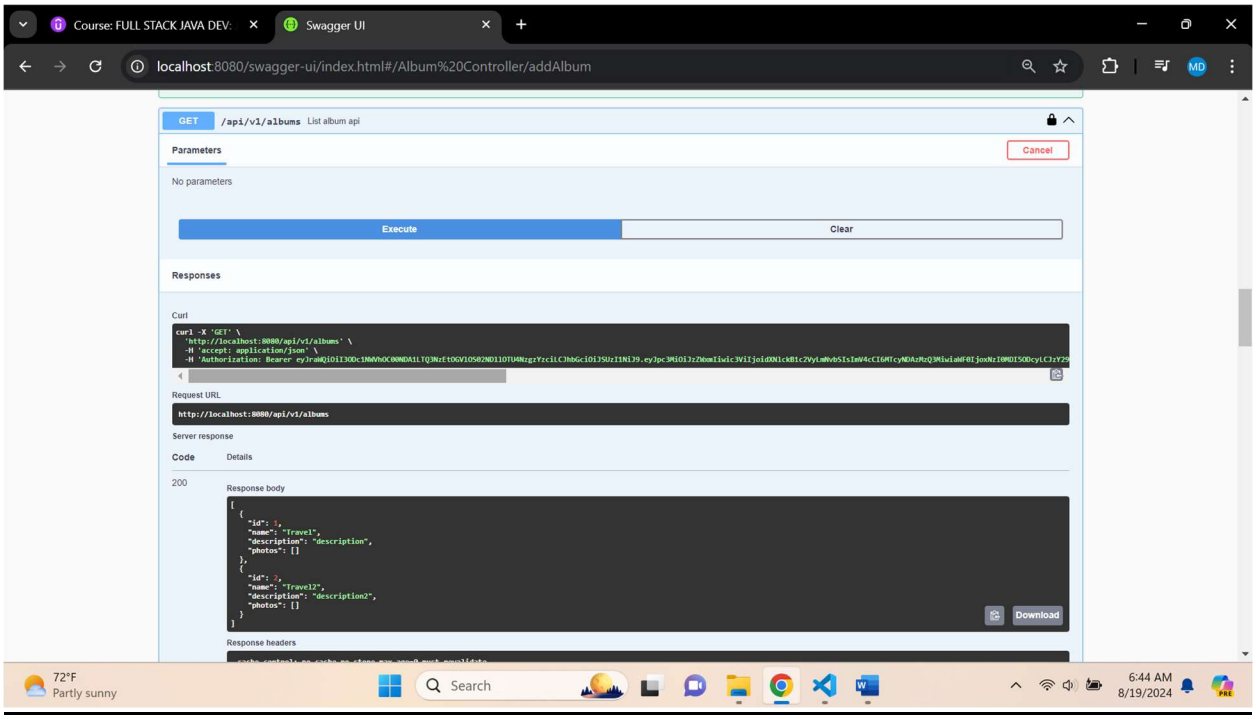
List of Albums - Starting



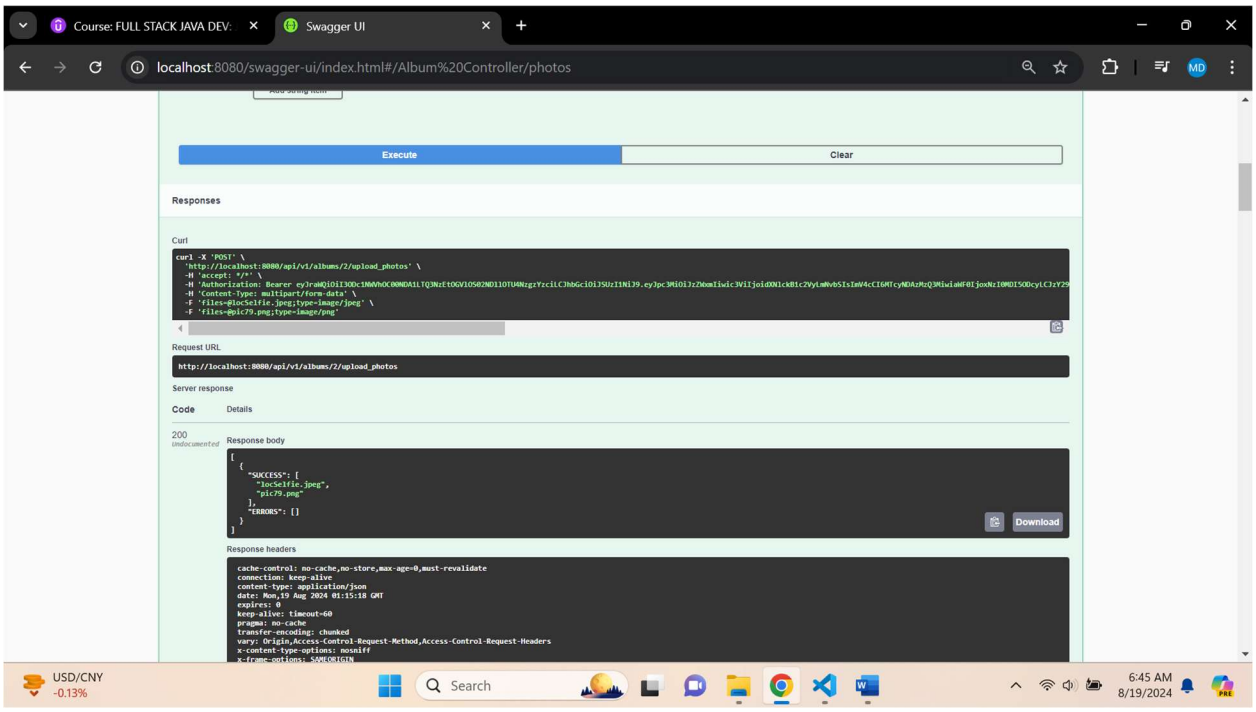
Add Album – More than 2



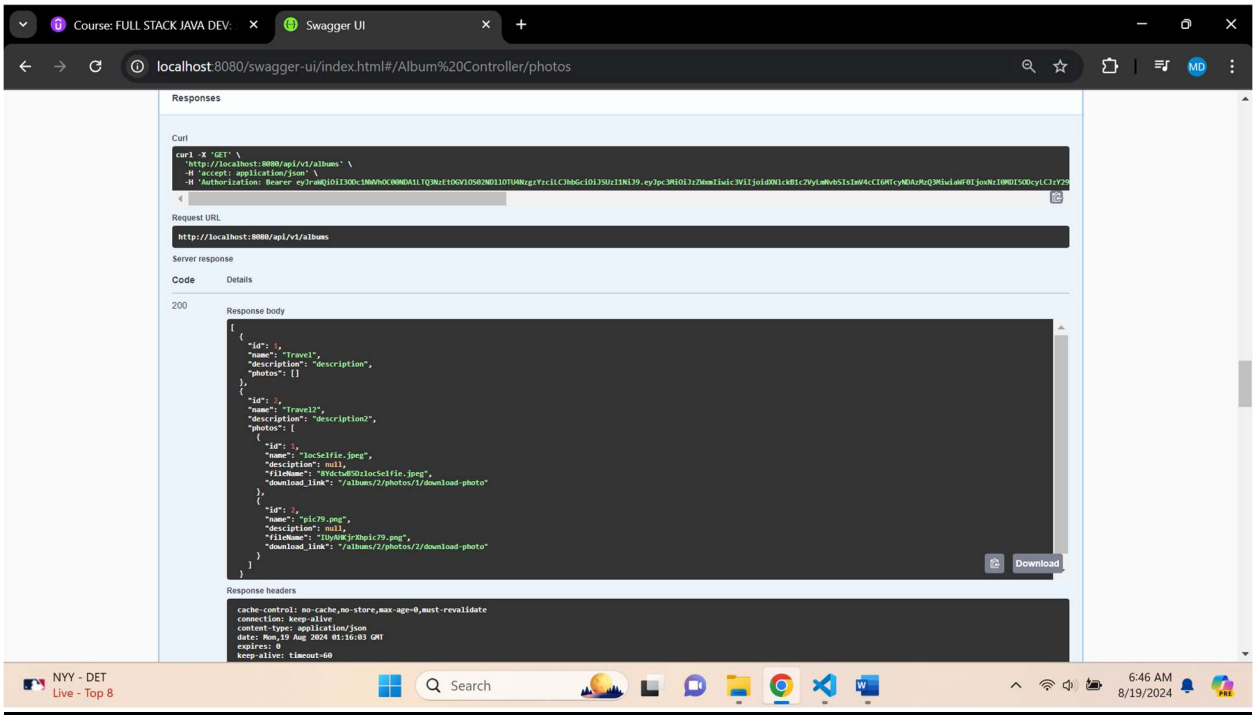
List of Albums with no photo uploaded



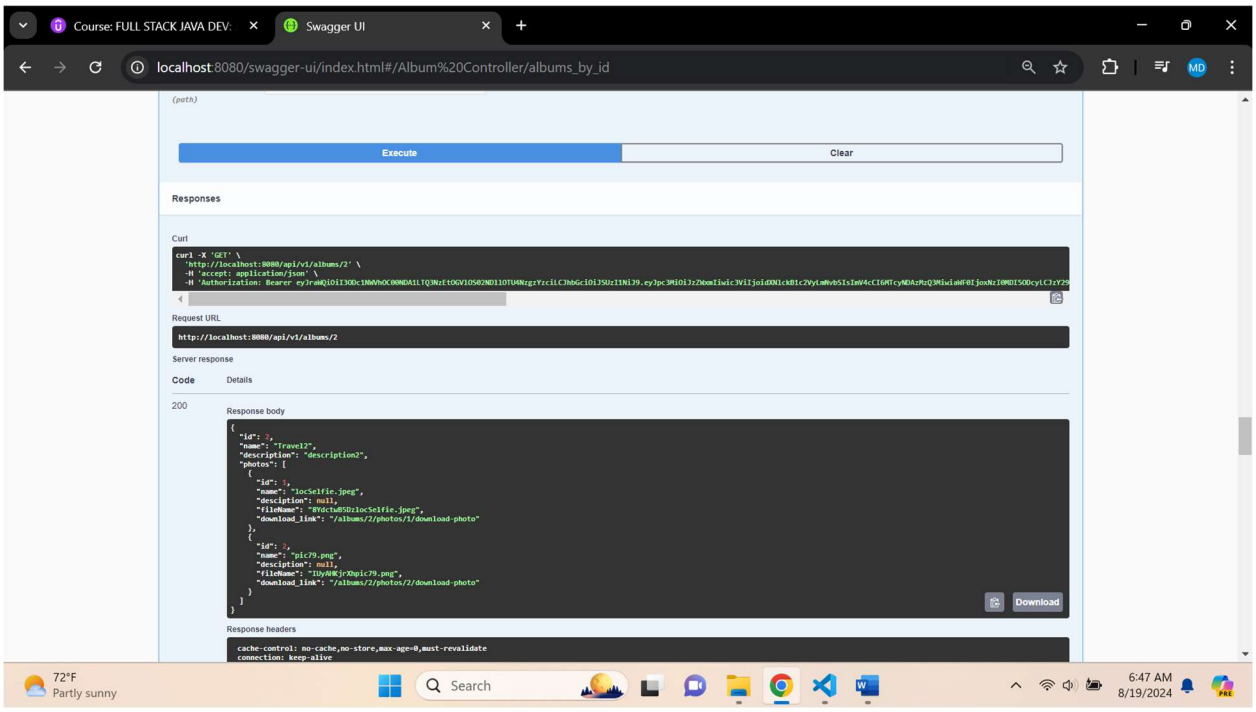
Updating Photo into album (id =2)



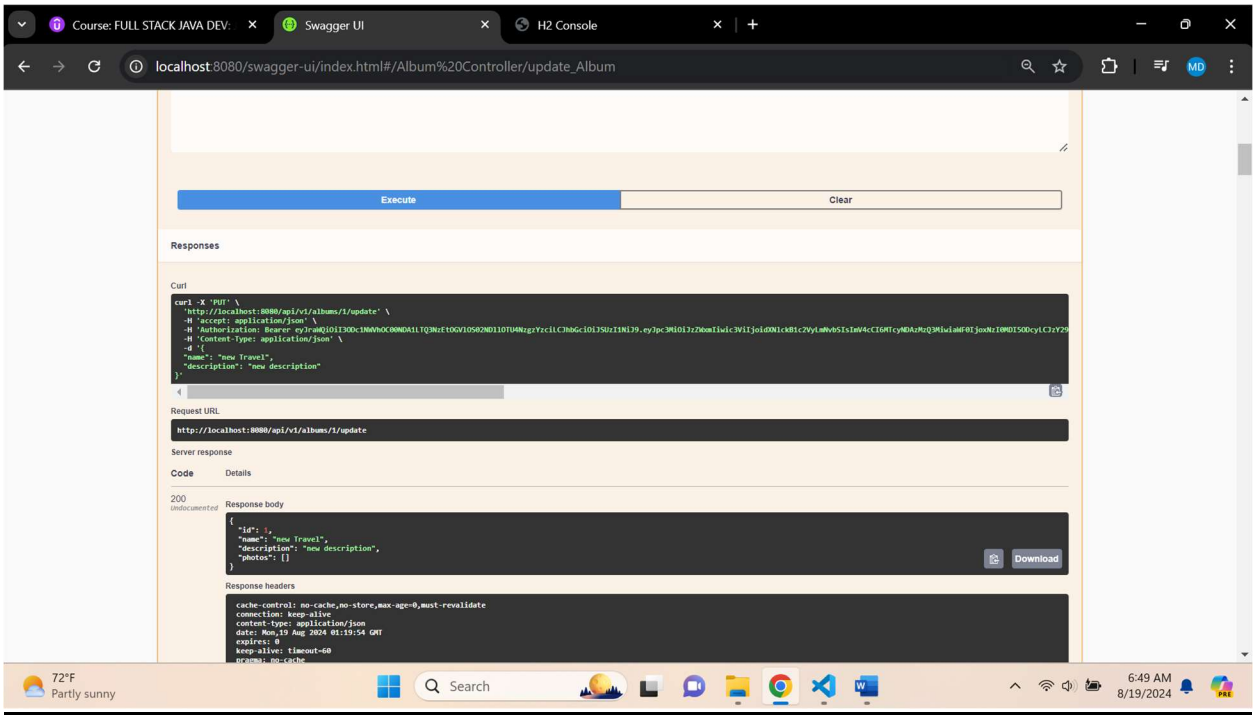
List of Albums with photo uploaded – id = 2



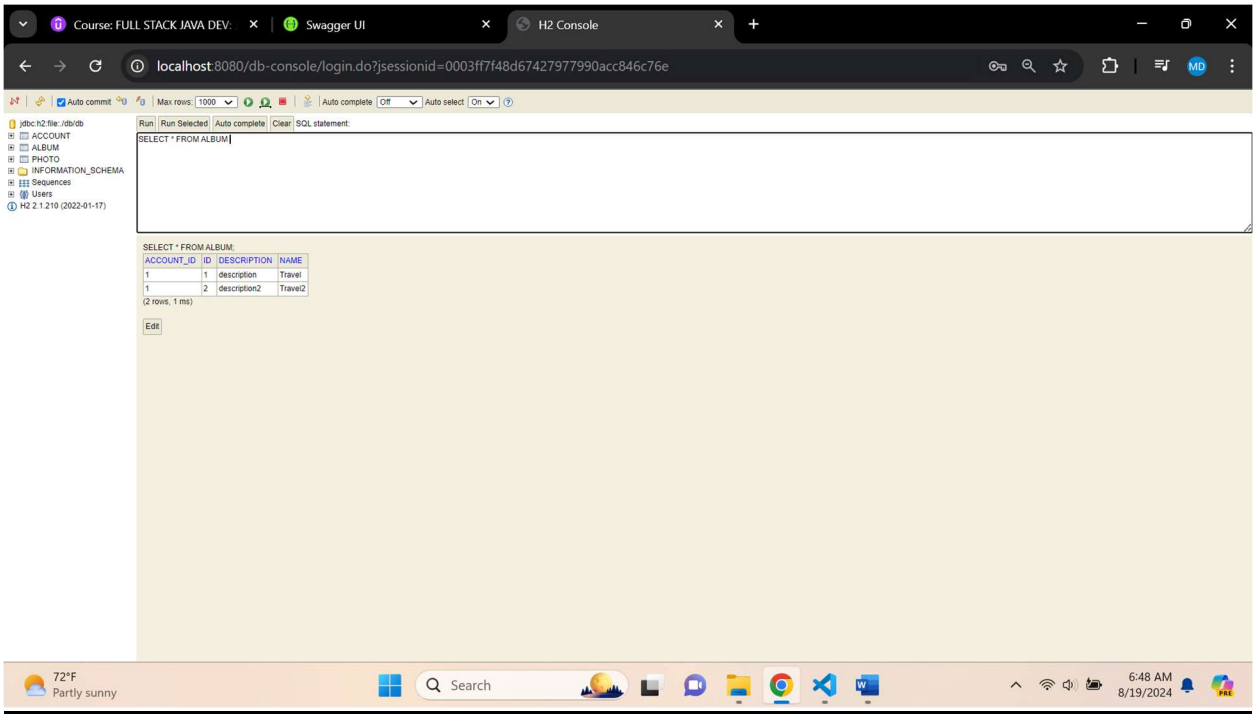
List album by album id



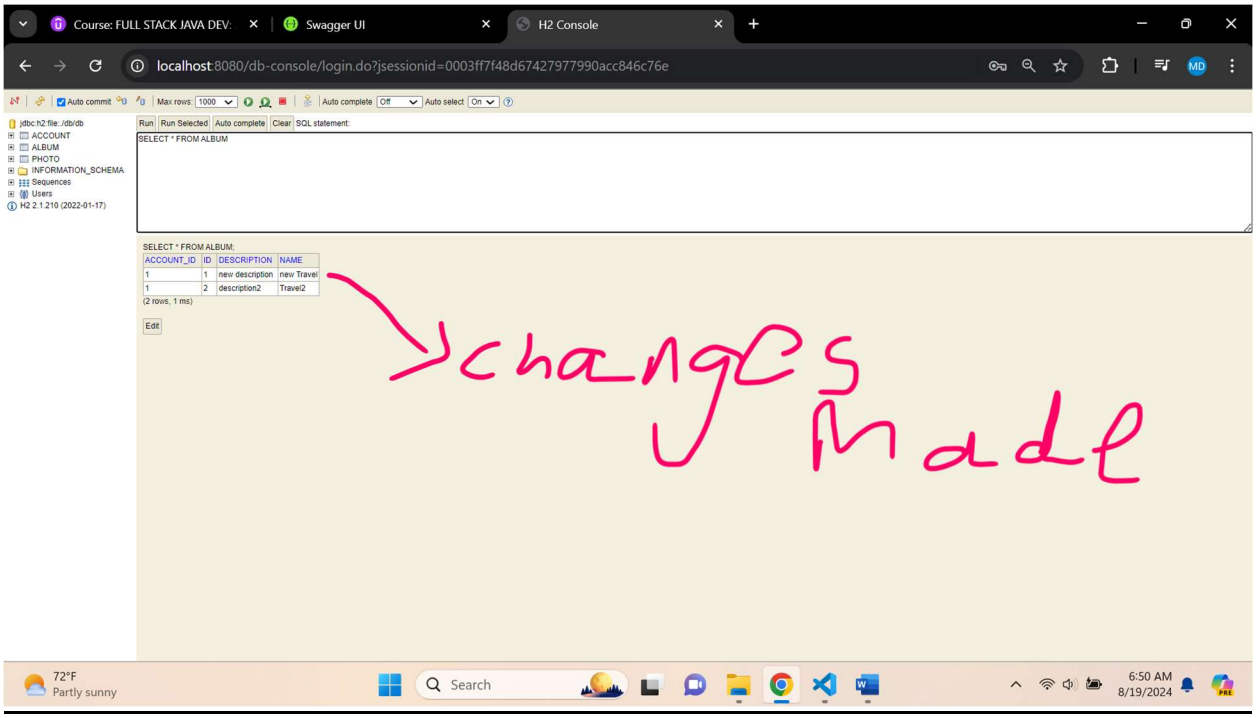
Updating Album



Db Console – Album – Before Update



Db Console – Album – After Update



Final Frontend API Swagger View for AlbumController