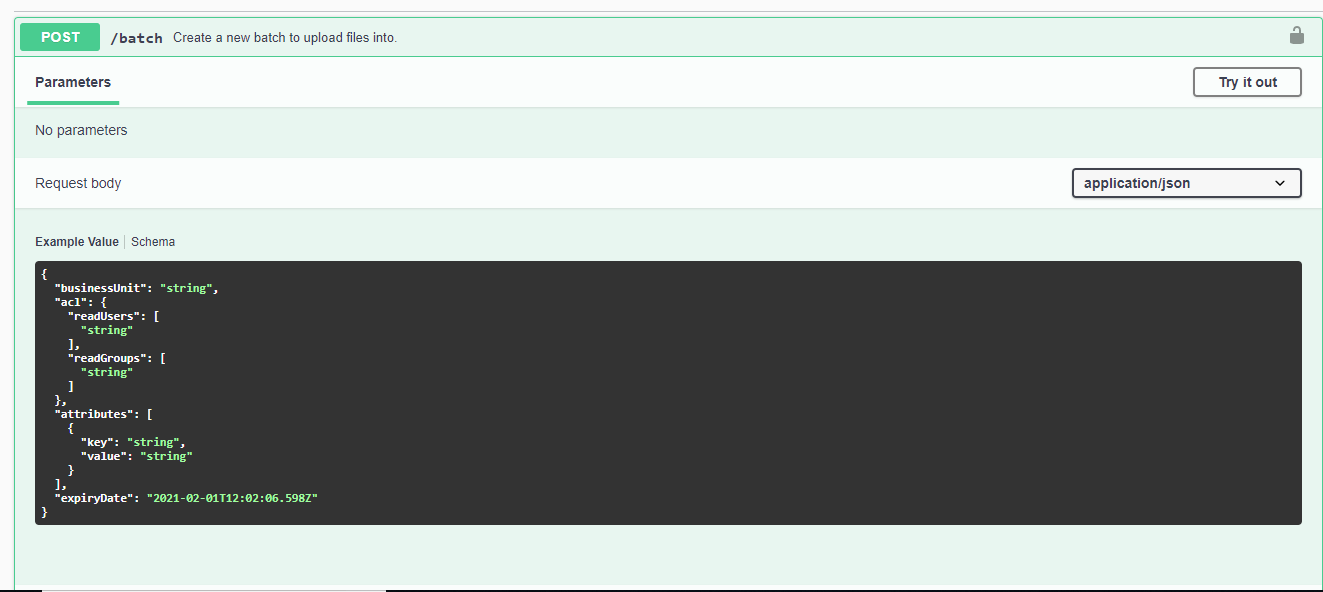
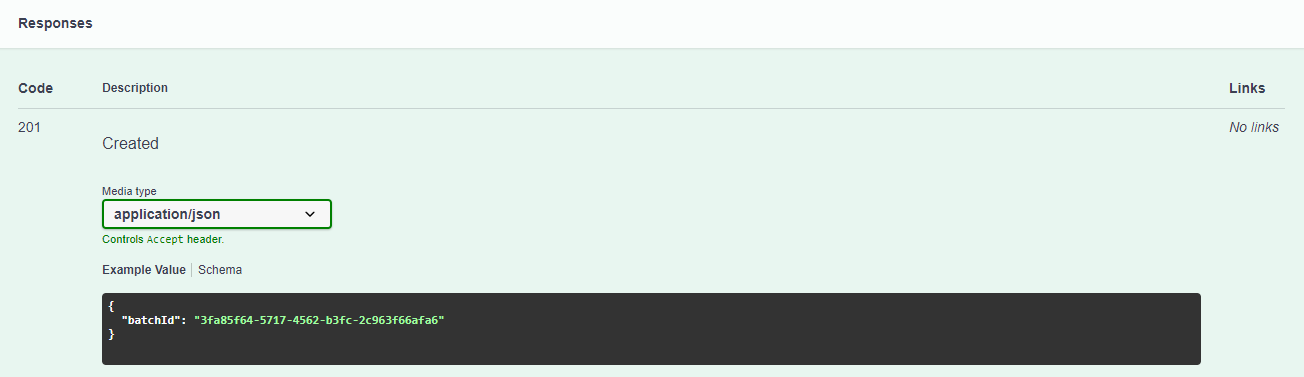
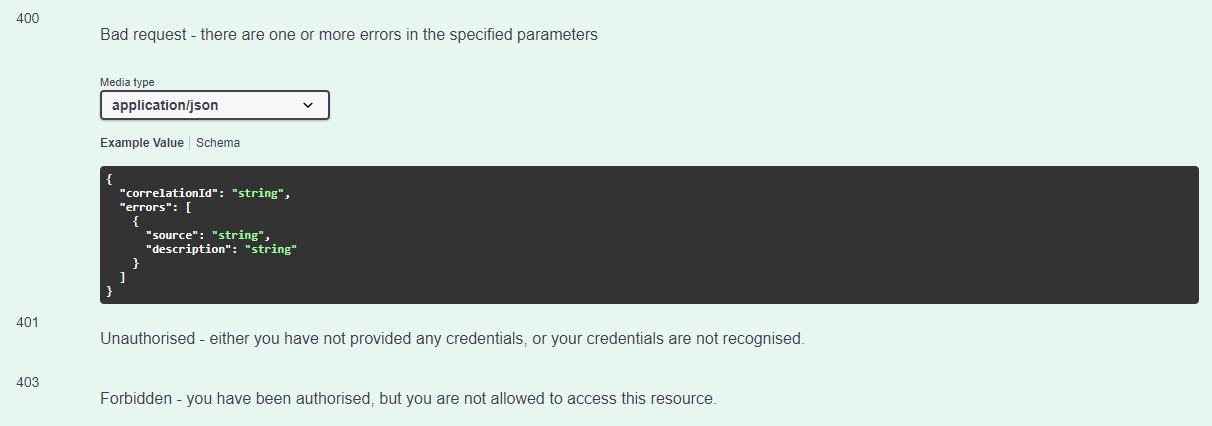
Create Dotnet Core Web API application and add below endpoint.

1. Create batch







The request body data should be in the below format:

{

"businessUnit": "string",

"acl": {

"readUsers": [

"string"

],

"readGroups": [

"string"

]

},

"attributes": [

{

"key": "string",

"value": "string"

}

],

"expiryDate":"2022-02-15T15:20:52Z",

"files": [

{

"filename": "string",

"fileSize": 0,

"mimeType": "string",

"hash": "string",

"attributes": [

{

"key": "string",

"value": "string"

}

]

}

]

}

Requirement:

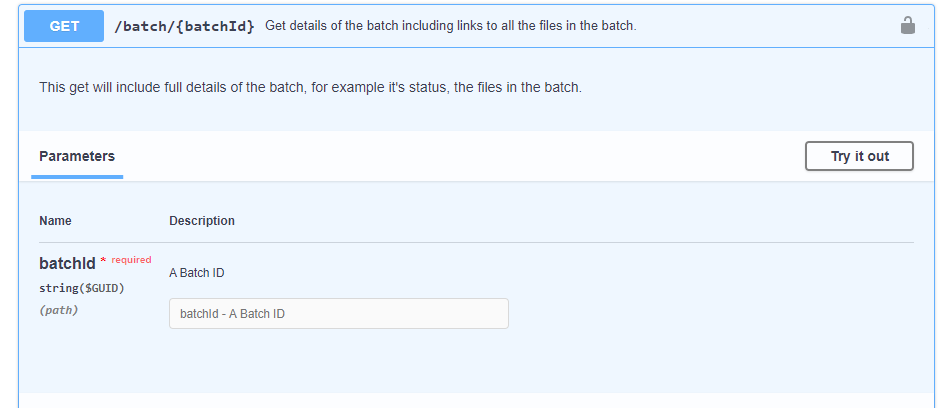
1. Create api endpoint as defined above and adhere above requirement
2. Persist details on table.

Apart from above details below are additional requirement

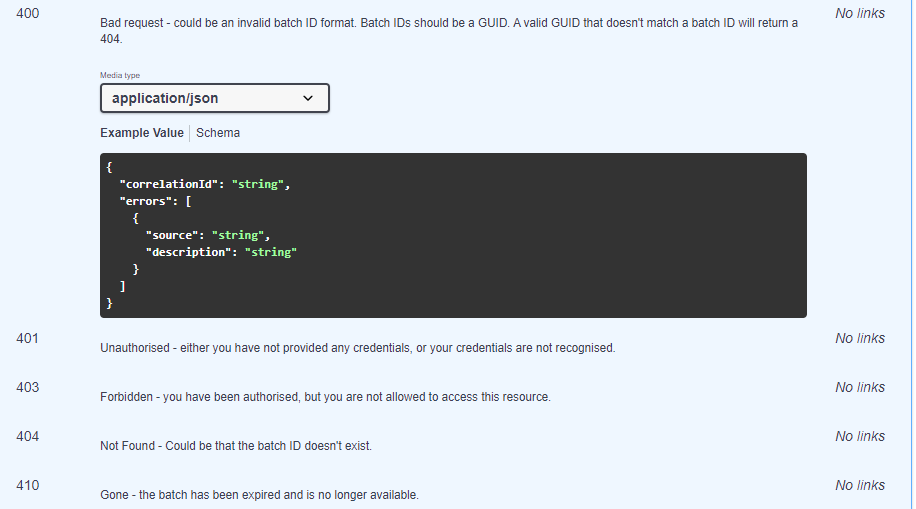
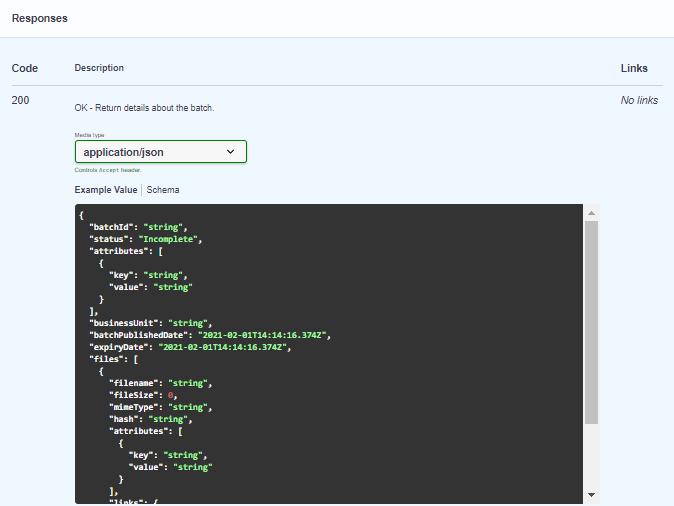
1. Make Business unit configurable and validate every request with it if not present give bad request.
2. Business unit should be validate against empty, if not present.
3. Atleast one file should be exists in batch request body data.
4. For batch and file attributes if present on request should have both key and value if not it should respond bad request.
5. Expiry date should not be past date. No validations for format.
6. Bad request response should be in format given above.
7. Created Batch should in GUID format.
8. Write UT for services, repository and controller.

Note: Don’t bother about authorisation for now.

2) Get Batch details



Responses:



Requirement:

1. This api endpoint should return details created from above endpoint
2. If any validations on parameter, the response should be 400 status code.
3. If the specified valid batchId not exists in DB, the response should be 404.

Note: Response object will not be same but should have all the relevant data used when creating batch.

Expectation:

1. Create dotnet core webapi application and add above endpoint
2. Add swagger support to your application and add details for your endpoint
3. Add Data Persistence layer(Entity frame work)
4. Use fluent validation for Validation
5. Add Unit testing for your api
6. Appropriate logging - Optional
7. Follow Microsoft coding convention - https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/inside-a-program/coding-conventions