

번호	기능명	API Method	End Point	입력 데이터	응답 데이터	예외	비고	우선순위 [1]	개발	테스트	배포	사용
0	공통	X	login, reissue 을 제외한 모든 요청	-header Authorization: Bearer Access	-	401 -body { "message": "Access Token is Expired", "data": null }	login을 제외한 모든 요청의 Header에 Authorization 을 추가	1				
		POST	/reissue	-Cookie refresh: RefreshToken	- header Authorization: AccessToken -body { "status": "success", "message": "Tokens successfully reissued", "data": { } }	- 401 -body { "status": "error", "message": "Refresh Token expired" }	Access Token 이 만료될 경우 Refresh Token 을 헤더에 담아 AccessToekn 재발급 요청	1				
1	로그인	GET	/oauth2/authorization/[도메인]	X	- cookie Authorization: RefresToken		전에 회원가입을 했다면 로그인 성공 아니면 회원가입 페이지(필명 검증)로 이동	1				
2	필명 중복 확인	GET	/users/pen-names/{penName}	X	{ "message": "PenName is not Duplication", "data": null }	- 409 { "message": "PenName already exists", "data": null }	해당 필명이 존재하는지 확인	1				
3	로그아웃	POST	/logout	-Cookie refresh: RefreshToken	{ "message": "Logout successful", "data": null }	- 400 { "message": "~~~", "data": null } -500 { "message": "~~~", "data": null }	해당 기기 로그인 종료 (RefreshToken 제거)	1				
4	탈퇴	DELET	/users	X	- body { "status": "success", "data": {}, "message": "Signout Success" }			1				
5	로그인 유저 프로필 조회	GET	{ "message": "Find Profile Success", "data": { "penName": "필명", "introduction": "작가 소개", "imageUrl": "이미지 경로", "privateStudio": { "studioId": 72, "name": "private", "description": "private" }, "tags": [{ "tagId": 3, "name": "현대소설", "color": "#C70039" }, { "tagId": 4, "name": "고전소설", "color": "#900C3F" }] } }	X	{ "message": "Find Profile Success", "data": { "penName": "필명", "introduction": "작가 소개", "imageUrl": "이미지 경로", "privateStudio": { "studioId": 72, "name": "private", "description": "private" }, "tags": [{ "tagId": 3, "name": "현대소설", "color": "#C70039" }, { "tagId": 4, "name": "고전소설", "color": "#900C3F" }] } }	- 403 Unauthorized { "message": "Usenot found", "data": null } - 403 Unauthorization { "message": "Profile not found", "data": null } - 404 ResourceNotFound { "message": "Studio not found", "data": null }						

6	프로필 변경	PUT	/users/profiles	<pre>{ "penName": "필명", "introduction": "작가 소개", "imageUrl": "이미지 경로", "tags": [{ "tagId": "1"}, { "tagId": "2"}] }</pre>	<pre>- body { "status": "success", "data": { "penName": "필명", "introduction": "작가 소개", "tags": [{ "tagId": "1"}, { "tagId": "2"}], "imageUrl": "이미지 경로" }, "message": "Profile Update Success" }</pre>	<pre>-409 Conflict { "data": null, "message": "Failed to update profile" } -400 Resource Not Found { "message": "해당 태그 존재하지 않음", "data": null }</pre>	작가 본인의 프로필을 변경한다 필명 중복 확인 이후 변경 가능	1				
7	상대 프로필 조회	GET	/users/profiles/{userId}	X	<pre>{ "message": "해당 유저 조회 성공", "data": { "userId": 106, "email": "siokim001@gmail.com", "isDelete": false, "penName": "필명", "introduction": "작가 소개", "imageUrl": "이미지 경로", "tags": [{ "tagId": 3, "name": "현대소설", "color": "#C70039" }, { "tagId": 4, "name": "고전소설", "color": "#900C3F" }] } }</pre>	<pre>-404 Not Found { "message": "User not found", "data": null } - 409 ConflictException { "message": "User is deleted", "data": null }</pre>						
8	스튜디오 생성	POST	/studios	<pre>{ "name": "title", "description": "desc" }</pre>	<pre>{ "message": "스튜디오 생성 성공", "data": null }</pre>	<pre>- 403 Unauthorized { "message": "로그인 유저 정보 조회 실패", "data": null } - 500 InternerServerException { "message": "DB 저장 실패", "data": null }</pre>	토큰으로 관리자의 요청만 수락 description null 가능	1				
9	참여한 스튜디오 조회	GET	/studios		<pre>{ "message": "참여 스튜디오 목록 조회 성공", "data": [{ "studioid": 73, "name": "title", "description": "desc" }, {}, {}] }</pre>		토큰으로 사용자 아이디에 맞는 studio 리턴 ROLE_PRIVATE인 개인 스튜디오만 조회	1				
10	개별 스튜디오 조회	GET	/studios/{stuioId}	X	<pre>{ "message": "스튜디오 상세 조회 성공", "data": { "studioid": 85, "name": "새로운 스튜디오", "description": "oo" } }</pre>	<pre>- 404 ResourceNotFound { "message": "해당 스튜디오 존재하지 않음", "data": null } - 403 Forbidden { "message": "Studio 접근 권한 없음", "data": null }</pre>						
11	스튜디오설정 변경	PUT	/studios/{studio_id}	<pre>{ "name": "수정할 제목", "descrip"</pre>	<pre>{ "message": "스튜디오 설정 변경 성공", "data": { "studioid": 85, "name": "새로운 이름", "description": "새로운 설명" } }</pre>	<pre>- 403 Forbidden { "message": "Studio 관리 권한 없음", "data": null } - 404 ResourceNotFound { "message": "해당 스튜디오 존재하지 않음", "data": null }</pre>	<pre>- 403 Forbidden { "message": "Studio 관리 권한 없음", "data": null } - 404 ResourceNotFound { "message": "해당 스튜디오 존재하지 않음", "data": null }</pre>	1				

12	스튜디오 삭제	DELETE	/studios/{studio_id}		{ "message": "스튜디오 삭제 성공", "data": null }	-403 Forbidden { "message": "Studio 관리 권한 없음", "data": null }	코드 상으로 Character DB 내용을 삭제함 이제 CASECADE 가 적용되 있으니 안해도 됨 (일단 주석 처리했음)	1			
13	스튜디오 탈퇴	DELETE	/studios/{studio_id}/members	X	{ "message": "스튜디오 탈퇴 성공", "data": null }	-403 Forbidden { "message": "OWNER 사용자는 추방할 수 없음", "data": null } -404 ResourceNotFound { "message": "해당 유저 Studio 에 존재하지 않음", "data": null }					
내 스튜디오 Role 조회				GET	/studios/{studioId}/my	X	{ "message": "유저 권한 조회 성공", "data": "ROLE_OWNER" }	-403 Forbidden { "message": "스튜디오 가입되지 않은 유저", "data": null }			
14	멤버 목록 조회	GET	/studios/{studio_id}/members		{ "message": "멤버 목록 조회 성공", "data": [{ "userId": 106, "penName": "익명의 작가_86f84cf3-1d03-2d17-f72d-0256cb368c82", "imageUri": "", "role": "ROLE_MEMBER" }, {}, {}] }			1			
15	멤버 권한 설정	PUT	/studios/{studioId}/members/{memberId}	{ "role" : "ROLE_OWNER" }	{ "data": null, "message": "멤버 권한 설정 변경 성공" }	-403 Forbidden { "data": null, "message": "설정할 수 없는 권한 : {ROLE}" } -404 ResourceNotFound { "data": null, "message": "해당 유저 스튜디오에 없음 " }	토큰으로 관리자의 요청만 수락 (ROLE_OWNER) 요청 X ROLE_MEMBER ROLE_VIEWER (ROLE_PRIVATE, ROLE_REQUESTER) 요청	1			
16	멤버 추방	DELETE	/studios/{studio_id}/members/{member_id}		{ "data": null "message": "멤버 추방 성공", }	-403 Forbidden { "message": "OWNER 사용자는 추방할 수 없음", "data": null } -404 ResourceNotFound { "message": "해당 유저 Studio 에 존재하지 않음", "data": null }	토큰으로 관리자의 요청만 수락 연관 정보 삭제 로직 고민 필요 db도 수정 필요할 듯	1			
17	멤버 초대 링크 생성	POST	/invitations/code/{studioId}	중복 생성 -> 수정했음	{ "data": "http:[도메인]/api/invitations/code/{studioId}/invitation/{invitationCode}", "message": "모집 코드 생성 성공" }	-404 ResourceNotFound { "data": null, "message": "해당 스튜디오 존재 X" } -404 ResourceNotFound { "data": null, "message": "해당 스튜디오 가입 X" } -409 Conflict { "data": null, "message": "개인 스튜디오에 초대 불가" }	토큰으로 관리자의 요청만 수락 보안 링크 생성, 유효기간 관리 -> 7일	3			
18	멤버 초대 링크 조회	GET	/invitations/code/{studioId}	중복 생성 문제로 유니크 하지 않음 -> DB 에 초대코드에 Studio_id	{ "data": "http:[도메인]/api/invitations/code/{studioId}/invitation/{invitationCode}", "message": "모집 코드 조회성공" }		토큰으로 관리자의 요청만 수락	3			
19	초대 링크 삭제	DELETE	/invitations/code/{invitationCode}		{ "data": null, "message": "모집 코드 삭제 성공" }		토큰으로 관리자의 요청만 수락				

20	초대 링크로 가입	PUT	초대링크		<pre>{ "data": null, "message": "초대 코드로 가입 성공" }</pre>	<pre>- 404 ResourceNotFound { "data": null, "message": "해당 코드 존재 X" } - 404 ResourceNotFound { "data": null, "message": "해당 코드 만료" } - 409 Conflict { "data": null, "message": "초대를 이미 존재함" } - 409 Conflict { "data": null, "message": "개인 스튜디오에 초대 불가" } - 404 ResourceNotFound { "data": null, "message": "해당 스튜디오 존재 X" }</pre>					
21	팀원 모집 공고 작성	POST	/invitations/{studioId}	<pre>{ "title": "title1", "description": "title2", "tags": [{"tagId": 1}, {"tagId": 2}] }</pre>	<pre>{ "data": null, "message": "모집글 생성 성공" }</pre>	<pre>- 409 Conflict { "data": null, "message": "초대를 이미 존재함" } - 409 Conflict { "data": null, "message": "개인 스튜디오에 초대 불가" } - 404 ResourceNotFound { "data": null, "message": "해당 스튜디오 존재 X" }</pre>	토큰으로 관리자의 요청만 수락 중복된 태그는 제거하고 하나만 허용 (HashS	1			
22	팀원 모집 공고 수정	PUT	/invitations/{studioId}	<pre>{ "title": "title1", "description": "title2", "tags": [{"tagId": 1}, {"tagId": 2}] }</pre>	<pre>{ "data": null, "message": "모집글 수정 성공" }</pre>	<pre>- 404 ResourceNotFound { "data": null, "message": "수정할 모집글 없음" } - 404 ResourcesNotFound { "data": null, "message": "태그가 존재하지 않음" }</pre>	토큰으로 관리자의 요청만 수락				
23	팀 모집 공고 삭제	DELETE	/invitations/{studioId}	X	<pre>{ "data": null "message": "모집글 삭제 성공", }</pre>	<pre>- 404 ResourceNotFound { "data": null, "message": "삭제할 모집글 없음" }</pre>	토큰으로 관리자의 요청만 수락				
24	팀 모집 게시글 조회	GET	/invitations	<p>Parameter page=0 (현재 페이지) -> 0부터, size=2 (페이지 당 content 수) - [sort=xxx[.desc]asc]]</p> <p>+ totalPages = 개수 (1, 2, 3) + totalElements = 전체 conten + size = 조회 content 수 + content = 게시물 배열</p>	<pre>{ "message": "모집글 전체 조회 ", "data": { "totalPages": 3, "totalElements": 5, "size": 2, "content": [{ "studioId": 75, "title": "studio1 팀원모집 ", "tags": [] }, { "studioId": 80, "title": "스튜디오 2 팀원모집 ", "tags": [] }] } }</pre>		1				

29	캐릭터 목록 조회	GET	/studios/{studio_id}/characters	X	<pre>{ "message": "Get Characters Success", "data": [{ "id": 13, "name": "변경 ", "description": "변경 캐릭터 설명", "imageUrl": "https://storyboat-character.s3.ap-northeast-2.amazonaws.com/ddd.jpg", "tags": null }, { "id": 15, "name": "캐릭터 1", "description": "캐릭터 설명 ㅇㅇ", "imageUrl": "https://storyboat-character.s3.ap-northeast-2.amazonaws.com/26b27ab8-ac5a-42b2-8d3c-3166da166ff5ddd.jpg", "tags": "1, 2, 3" }, { "id": 16, "name": "캐릭터 1", "description": "캐릭터 설명 ㅇㅇ", "imageUrl": "https://storyboat-character.s3.ap-northeast-2.amazonaws.com/8fd6308d-5315-4079-854d-24f8cdcc6d3bddd.jpg", "tags": "1, 2, 3" }, { "id": 17, "name": "캐릭터 1", "description": "캐릭터 설명 ㅇㅇ", "imageUrl": "https://storyboat-character.s3.ap-northeast-2.amazonaws.com/8fd6308d-5315-4079-854d-24f8cdcc6d3bddd.jpg", "tags": "1, 2, 3" }] }</pre>					1	
30	캐릭터 생성	POST	/studios/{studio_id}/characters	-> Form 데이터? name: 캐릭터 명 1 description =캐릭터 설명 1 tags: 호러, + file: 이미지 파일	<pre>{ "message": "캐릭터 생성 성공", "data": null }</pre>	<pre>- 404 ResourceNotFound { "message": "해당 스튜디오 존재하지 않음", "data": null } - 400 ResourceNotFound { "message": "업로드할 파일 없음", "data": null } - 500 Interner Servier Error { "message": "캐릭터 존재하지 않음", "data": null } - 400 ResourceNotFound { "message": "캐릭터 존재하지 않음", "data": null } - 403 Forbidden { "message": "Studio 접근 권한 없음", "data": null }</pre>	대그... 처리 로직을 잘 모르겠음				1
31	캐릭터 삭제	DELETE	/studios/{studio_id}/characters/{chara	X	<pre>{ "message": "캐릭터 삭제 성공", "data": null }</pre>	<pre>- 404 ResourceNotFound { "message": "캐릭터 존재하지 않음", "data": null } - 403 Forbidden { "message": "Studio 접근 권한 없음", "data": null }</pre>					1
32	캐릭터 수정	PUT	/studios/{studio_id}/characters/{chara	{ "name": "캐릭터 명", "description": "캐릭터 설명", "file": "파일", "tags": "일, 둘, 삼", }	<pre>{ "message": "캐릭터 수정 성공", "data": null }</pre>	<pre>{ "message": "찾을 수 없는 캐릭터입니다 ", "data": null } - 500 Interner Servier Error { "message": "S3 파일 업로드 중 오류가 발생", "data": null }</pre>					1

33	캐릭터 내보내기	POST	/studios/{studio_id}/characters/{chara_id}	X	{ "message": "캐릭터 업로드 성공", "data": null }	- 400 ResourceNotFound { "message": "캐릭터 존재하지 않음", "data": null } - 403 { "message": "캐릭터 전송 권한 없음", "data": null } - 403 Forbidden { "message": "Studio 접근 권한 없음", "data": null } - 403 Forbidden { "message": "캐릭터 전송 권한 없음", "data": null }	개인 저장소의 캐릭터를 템 저장소의 캐릭터로 업로드! (뒤에 studio_id 가 올라갈 studio)	2			
34	아이디어 리스트 조회	GET	/studios/{studio_id}/ideas	X	{ "message": "아이디어 목록 조회 성공", "data": [{ "ideald": 53, "title": "아이디어 1", "content": "내용 1" }, { "ideald": 54, "title": "아이디어 2", "content": "내용 2" }] }	- 403 Forbidden { "message": "Studio 접근 권한 없음", "data": null }		1			
35	아이디어 생성	POST	/studios/{studio_id}/ideas	{ "title": "아이디어 노트 제목", "content": "노트 내용" }	{ "message": "아이디어 생성 성공", "data": null }	- 403 Forbidden { "message": "Studio 수정 권한 없음", "data": null } - 400 { "message": "Studio 없음", "data": null }		1			
36	아이디어 상세보기	GET	/studios/{studio_id}/ideas/{idea_id}	X	{ "message": "아이디어 조회 성공", "data": { "ideald": 53, "title": "아이디어 1", "content": "내용 1" } }	- 403 Forbidden { "message": "Studio 접근 권한 없음", "data": null } - 400 ResourceNotFound { "message": "아이디어 존재하지 않음", "data": null }		1			
37	아이디어 수정	PUT	/studios/{studio_id}/ideas/{idea_id}	{ "title": "아이디어 노트 제목", "content": "노트 내용" }	{ "message": "아이디어 수정 성공", "data": { "ideald": 53, "title": "새 아이디어 2", "content": "새 내용 2" } }	- 403 Forbidden { "message": "Studio 수정 권한 없음", "data": null } - 400 ResourceNotFound { "message": "아이디어 존재하지 않음", "data": null }		1			

38	아이디어 삭제	DELETE	/studios/{studio_id}/ideas/{idea_id}	X	{ "message": "아이디어 삭제 성공", "data": null }	- 403 Forbidden { "message": "Studio 수정 권한 없음", "data": null } - 400 ResourceNotFound { "message": "아이디어 존재하지 않음", "data": null }		1			
39	스토리 박스 목록 조회	GET	/studios/{studio_id}/stories	Parameter page=0 (현재 페이지) -> 0부터, size=2 (페이지 당 content 수) - [sort=xxx[,desc asc]] + totalPages = 개수 (1, 2, 3) + totalElements = 전체 content 수 + size = 조회 content 수 + content = 게시물 배열	{ "message": "스토리 목록 조회 성공", "data": { "totalPages": 1, "totalElements": 1, "size": 20, "content": [{ "storyId": 15, "title": "스토리 제목" }] } }		토큰으로 권한 확인 ROLE_REQUESTER 이하는 확인 불가능	1			
40	스토리 박스 생성	POST	/studios/{studio_id}/stories	{ "title": "스토리 제목" }	{ "data": {}, "message": "스토리 생성 성공" }	- 404 ResourceNotFound { "data": null, "message": "스튜디오를 찾을 수 없음" }	토큰으로 권한 확인 ROLE_REQUESTER, ROLE_VIEWER 이하는	1			
41	스토리 박스 삭제	DELETE	/studios/{studio_id}/stories/{story_id}	X	{ "message": "스토리 삭제 성공", "data": null }	- 401 UnauthorizedException { "data": null, "message": "권한 없음" } - 404 ResourceNotFoundException { "data": null, "message": "스튜디오를 찾을 수 없음" } - 404 ResourceNotFoundException { "data": null, "message": "스토리 찾을 수 없음" }	OWNER, PRIVATE만 삭제 가능				
42	부 조회 (폴렛 편집 화	GET	/studios/{studio_id}/stories/{story_id}	X	{ "data": {???}, "message": "스토리 조회 성공" }	- 404 ResourceNotFoundException { "data": null, "message": "스토리 찾을 수 없음" }	VIEWER 이상 권한만 가능	1			
43	스토리 수정(저장)	PUT	/studios/{studio_id}/stories/{story_id}	{ json }	{ "data": {}, "message": "스토리 저장 성공" }	- 404 ResourceNotFoundException { "data": null, "message": "스튜디오를 찾을 수 없음" } - 403 Forbidden { "data": null, "message": "권한 없음", } - 500 InternalServerErrorException { "data": null, "message": "MongoDB 저장 오류" }	MEMBER 이상 권한만 가능	1			

44	토리 업로드 (개인 -> 단)	POST	/studios/{from_studio_id}/upload/{story_id}	X	{ "status": "success", "data": {}, "message": "스토리 업로드 성공" }	- 404 ResourceNotFoundException { "data": null, "message": "스토리 찾을 수 없음" } - 404 ResourceNotFoundException { "data": null, "message": "스튜디오 찾을 수 없음" } - 500 InternalServerErrorException { "data": null, "message": "MongoDB 저장 오류" }	개인 저장소의 Story를 템 저장소의 Story로 업로드! (뒤에 studio_id 가 올라갈 studio)	2			
45	토리 수정 (저장) 이력 조회	GET	/studios/{studio_id}/stories/{story_id}	Parameter page=0 (현재 페이지) -> 0부터, size=2 (페이지 당 content 수) - [sort=xxx[,desc asc]] + totalPages = 개수 (1, 2, 3) + totalElements = 전체 content 수 + size = 조회 content 수 + content = 게시물 배열	{ "message": "히스토리 불러오기 성공", "data": { "totalPages": 1, "totalElements": 3, "size": 20, "content": [{ "storyId": "66b1b5406e4a4202250a18ca", "dateTime": "2024-08-06T00:54:26.197", "penName": "필명", "imageUrl": "이미지 경로" }, { "storyId": "66b0f5b294b85612040acef3", "dateTime": "2024-08-06T00:13:34.993", "penName": "필명", "imageUrl": "이미지 경로" }, { "storyId": "66b0ec1e38dea314eaf75b7f", "dateTime": "2024-08-06T00:09:14.969", "penName": "필명", "imageUrl": "이미지 경로" }] } }	수정자 필명, 수정 시간	2				
46	토리 해당 버전 미리보기	GET	/studios/{studioId}/stories/{studioStoryId}/histories/{storyId}		{ "message": "Histories Find Success", "data": "{\r\n \"templ\": \"templ\"\r\n}" }			2			
47	스토리 롤백	PUT	/studios/{studioId}/stories/{studioStoryId}/histories/{storyId}		{ "message": "Rollback Success", "data": null }			2			
48	태그 리스트 조회	GET	/tags		{ "message": "태그 목록 조회 성공", "data": [{ "tagId": 2, "name": "소설", "color": "#FF5733" }, { "tagId": 3, "name": "현대소설", "color": "#C70039" }, { "tagId": 4, "name": "고전소설", "color": "#900C3F" }, { "tagId": 5, "name": "판타지", "color": "#FFC300" }] }						

