



API 명세서

API 기준문서 [Image Generation]

1. 사용자 관리 (User Management)

기능	URL	Method	Request Body	API Response
회원가입	/api/signup	POST	{ "email": "string", "password": "string" }	<p>201 Created: { "success": true, "status": 201, "message": "회원가입 성공", "data": null, "timestamp": "string", "path": "/api/signup" }</p> <p>409 Conflict: { "success": false, "status": 409, "message": "이미 사용 중인 이메일입니다.", "timestamp": "string", "path": "/api/signup" }</p>
닉네임 중복 체크	/api/nickname	GET	{ "nickname": "string" }	<p>200 OK: { "success": true, "status": 200, "message": "닉네임 사용가능", "data": "nickname", "timestamp": "string", "path": "/api/check-nickname" }</p> <p>409 Conflict: { "success": false, "status": 409, "message": "이미 사용중인 닉네임입니다.", "data": null, "timestamp": "string", "path": "/api/check-nickname" }</p>
로그인	/api/user/login	POST	<p>{ "username": "string", "password": "string" }</p> <p>form data로 요청 보낼 것.</p>	<p>200 OK: { "success": true, "status": 200, "message": "로그인 성공", "data": { "nickname": "string" }, "timestamp": "string", "path": "/api/user/login" }</p> <p>401 Unauthorized: { "success": false, "status": 401, "message": "로그인 실패: 자격 증명에 실패하였습니다.", "data": null, "timestamp": "string", "path": "/api/user/login" }</p>

2. 방 관리 (Room Management)

기능	URL	Method	Request Body	API Response
방 생성	/rooms	POST	{ "id": "string", "name": "string", "maxCapacity": 0, "password": "string", "link": "string", "ownerId": "string" }	<p>201 Created:</p> <p>{ "success": true, "status": 201, "message": "방이 생성 되었습니다.", "data": { "id": "session", "name": "hi", "users": ["string"] } }</p>

				<pre>], "owner": "string", "maxCapacity": 6, "password": "string", "link": "string" }, "timestamp": "2025-01-22T13:42:13.158564", "path": "/rooms" }</pre>
방 입장	/rooms/{roomId}	POST	Request Param : String userId	<pre>400 BadRequest : { "success": false, "status": 400, "message": "방 정원이 다 찹니다.", "data": null, "timestamp": "2025-01-22T13:33:41.725140500", "path": "/rooms/sessionid" } 409 Confilct : { "success": false, "status": 409, "message": "이미 입장 되어 있는 방입니다.", "data": null, "timestamp": "2025-01-22T13:53:50.523724900", "path": "/rooms/sessionid" } 404 NotFount : { "success": false, "status": 404, "message": "잘못된 방 ID입니다.", "data": null, "timestamp": "2025-01-22T13:56:13.481571500", "path": "/rooms/sessionid1" } 200 OK : { "success": true, "status": 200, "message": "방 입장 성공", "data": null, "timestamp": "2025-01-22T13:56:47.693348800", "path": "/rooms/session" }</pre>
방 삭제	/rooms	DELETE	Request Param : String roomId	<pre>404 NotFount : { "success": false, "status": 404, "message": "해당 roomId를 가진 방이 없습니다.", "data": null,</pre>

			<div>"timestamp": "2025-01-22T14:02:48.296778", "path": "/rooms" }</div> <div>200 OK :</div> <div>{ "success": true, "status": 200, "message": "방 삭제 성공", "data": null, "timestamp": "2025-01-22T14:03:30.125095600", "path": "/rooms" }</div>
방 전체 목록 조회	/rooms	GET	<div>200 OK:</div> <div>{ "success": true, "status": 200, "message": "전체 방 정보 가져오기 성공", "data": [{ "id": "string", "name": "string", "users": ["string"], "owner": "string", "maxCapacity": 0, "password": "string", "link": "string" }, { "id": "sessionid", "name": "초보들 오셈", "users": ["sdf", "asd", "qwe", "a"], "owner": "sdf", "maxCapacity": 4, "password": "password", "link": "link" }], "timestamp": "2025-01-22T14:11:05.219642700", "path": "/rooms" }</div> <div>404 NOT FOUND :</div> <div>{ "success": false, "status": 404, "message": "생성되어 있는 방이 없습니다.", "data": null, "timestamp": "2025-01-22T14:17:26.049359900", "path": "/rooms" }</div>

				<div>404 NOT FOUND :</div> <div><pre>{ "success": false, "status": 404, "message": "해당 roomId를 가진 방을 찾을 수 없습니다.", "data": null, "timestamp": "2025-01-23T13:57:24.345907100", "path": "/rooms/asd" }</pre></div> <div>404 NOT FOUND :</div> <div><pre>{ "success": false, "status": 404, "message": "해당 사용자가 방에 존재하지 않습니다.", "data": null, "timestamp": "2025-01-23T13:58:20.163825700", "path": "/rooms/sessionid" }</pre></div>
방 나가기	/rooms/{roomId}	PATCH	RequestParam : userId	<div>200 OK :</div> <div><pre>{ "success": true, "status": 200, "message": "방장이 나갔습니다. 새로운 방장은 qwe1234입니다.", "data": "qwe1234", "timestamp": "2025-01-23T14:00:57.282640800", "path": "/rooms/sessionid" }</pre></div> <div>200 OK :</div> <div><pre>{ "success": true, "status": 200, "message": "방장이 방을 나가고 방이 삭제되었습니다.", "data": null, "timestamp": "2025-01-23T14:01:43.110917", "path": "/rooms/sessionid" }</pre></div>

3. 게임 관리 (Game Management)

기능	URL	Method	Request Body	API Response
게임 시작 (게임 시작 시 방장이 받은 카드 데이터를 반환)	/game	POST	<pre>{ "bossId": "string", "player": ["string"] "drawingStyle" : 0 }</pre>	201 created: <pre>{ "success": true, "status": 201, "message": "게임 생성", "data": { "gameId": "848fabf6-1f3d-4830-9216-</pre>

				<pre>a9400f5cf748", "status": { "userId": "bossid", "storyCards": [{ "id": 26, "keyword": "유령", "attribute": "인물" }, { "id": 123, "keyword": "시합/시험", "attribute": "사건" }, { "id": 147, "keyword": "사악함", "attribute": "상태" }, { "id": 49, "keyword": "옷", "attribute": "사물" }], "endingCard": { "id": 22, "content": "왕은 자신의 계약을 이행했고 모두가 행복해졌습니다." } }, "timestamp": "2025-02-05T15:51:12.644644500", "path": "/game" } 400 Bad Request: { "success": false, "status": 400, "message": "플레이어 수가 2명 미만입니다..", "data": null, "timestamp": "2025-02-05T15:52:34.423145300", "path": "/game" }</pre>
게임 시작 (게임 시작 시 방장이 받은 카드 데이터를 반환) 시연용 로직	/game/test	POST	<pre>{ "bossId": "string", "player": ["string"] "drawingStyle" : 0 }</pre>	<pre>201 created: { "success": true, "status": 201, "message": "게임 생성", "data": { "gameId": "848fabf6-1f3d-4830-9216-a9400f5cf748", "status": { "userId": "bossid", "storyCards": [{</pre>

				<pre>"id": 26, "keyword": "유령", "attribute": "인물" }, { "id": 123, "keyword": "시합/시험", "attribute": "사건" }, { "id": 147, "keyword": "사악함", "attribute": "상태" }, { "id": 49, "keyword": "옷", "attribute": "사물" }], "endingCard": { "id": 22, "content": "왕은 자신의 계약을 이행했고 모두가 행복해졌습니다." } }, "timestamp": "2025-02-05T15:51:12.644644500", "path": "/game" }</pre> <pre>400 Bad Request: { "success": false, "status": 400, "message": "플레이어 수가 2명 미만입니다..", "data": null, "timestamp": "2025-02-05T15:52:34.423145300", "path": "/game" }</pre>
게임 데이터 삭제 (게임 종료) isForceStopped true : 비정상 종료 false : 정상 종료 전원 패배 시 비정상 종료 게임 중 모두 나갈시 비정상 종료	/game	delete	{ "gameId" : "String", "isForceStopped" : true }	<pre>400 Bad_request: { "success": false, "status": 400, "message": "해당 gameId는 존재하지 않습니다.", "data": null, "timestamp": "2025-02-03T13:30:19.745321600", "path": "/game" }</pre> <pre>500 Internal Server Error: { "success": false, "status": 500, "message": "GPU 서버 통신 중 오류 발생",</pre>

				<div>"data": null, "timestamp": "2025-02-03T13:30:19.745321600", "path": "/game" }</div> <div>200 OK : { "success": true, "status": 200, "message": "비정상종료로 인한 게임 데이터 삭제 성공", "data": null, "timestamp": "2025-02-11T13:44:37.174859700", "path": "/game" }</div> <div>201 Created: { "data" : { "bookId" : "Stirng", "title" : "String", "bookCover" : 바이너리데이터" } }</div>
player별 보유 카드 데이터 조회	/game	GET	Param userId, gameId	<div>400 Bad Request : { "success": false, "status": 400, "message": "해당 userId는 게임에 존재 하지 않습니다.", "data": null, "timestamp": "2025-02-04T20:04:26.214310800", "path": "/game/6f0bdeb6-b3a4-4656-b078-8995e32b3790" }</div> <div>200 OK : { "success": true, "status": 200, "message": "플레이어 카드 상태 반환 성공", "data": { "userId": "QWE", "storyCards": [{ "id": 21, "keyword": "경비병/보초", "attribute": "인물" }, {</div>

				<pre>"id": 100, "keyword": "다리", "attribute": "장소" }, { "id": 124, "keyword": "시간이 흐름", "attribute": "사건" }, { "id": 141, "keyword": "용감한/용기있 는", "attribute": "상태" }], "endingCard": { "id": 11, "content": "이것이 바로 그 왕국이 이토록 특이한 이름을 갖게된 사연입니다." } }, "timestamp": "2025-02- 04T20:07:51.642319200", "path": "/game/6f0bdeb6- b3a4-4656-b078- 8995e32b3790" }</pre>
EndingCard 리롤 (반환 데이터로 리롤된 Ending Card 정보 보냄)	/game/shuffle	PATCH	Param: userId, gameId	<pre>400 Bad Request: { "success": false, "status": 400, "message": "해당 userId는 존재하지 않습니다.", "data": null, "timestamp": "2025-02- 04T20:10:40.640762", "path": "/game/shuffle/6f0bdeb6- b3a4-4656-b078- 8995e32b3790" } 200 OK : { "success": true, "status": 200, "message": "EndingCard 리롤 성공", "data": { "id": 8, "content": "매일마다 그는 자 신이 반향한 댓가를 보면서 눈 물을 흘렸습니다." }, "timestamp": "2025-02- 09T00:11:14.362219300", "path": "/game/shuffle" }</pre>

4. SCENE 관리 (이미지 생성)

기능	URL	Method	Request Body	API Response
Card 이미지 생성	/scene	POST		400 Bad Request: { "success": false, "status": 400, "message": "존재하지 않는 gameld입니다.", "data": null, "timestamp": "2025-02-08T13:29:43.437571800", "path": "/scene/storyCard" }
				400 Bad Request: { "success": false, "status": 400, "message": "해당 게임에 존재하지 않는 userId입니다.", "data": null, "timestamp": "2025-02-08T13:30:13.272209500", "path": "/scene/storyCard" }
			{ "gameld": "string", "userId": "string", "userPrompt": "string", "turn" : 0 }	500 Internal Server Error: { "success": false, "status": 500, "message": "GPU 서버 통신 중 오류 발생", "data": null, "timestamp": "2025-02-08T13:30:13.272209500", "path": "/scene/storyCard" }
				500 Internal Server Error: { "success": false, "status": 500, "message": "이미지 받기 실패", "data": null, "timestamp": "2025-02-08T13:30:13.272209500", "path": "/scene/storyCard" }
				201 Created: Content-Type: image/png 이미지 파일 반환
프롬프트 필터링	/scene/filtering	POST	{ "userId" : "String", "gameld" : "String", "userPrompt" : "String" }	200 OK: { "success": true, "status": 200, "message": "Prompt 필터링 성공", "data": 30,

				<pre>"timestamp": "2025-02-11T13:23:51.882507100", "path": "/scene/filtering" } 400 Bad Request: { "success": false, "status": 400, "message": "Prompt에 중복된 카드가 사용되었습니다.", "data": null, "timestamp": "2025-02-08T13:18:24.670710200", "path": "/scene/filtering" } 400 Bad Request: { "success": false, "status": 400, "message": "Prompt에 욕설이 사용되었습니다.", "data": null, "timestamp": "2025-02-11T10:41:18.319038500", "path": "/scene/filtering" } 400 Bad Request: { "success": false, "status": 400, "message": "Prompt에 소유한 카드가 사용되지 않았습니다.", "data": null, "timestamp": "2025-02-11T13:24:41.745874300", "path": "/scene/filtering" }</pre>
투표 결과에 따라 생성된 이미지 데이터 삭제 (true면 찬성, false면 반대) (cardId == -1 이면 엔딩카드를 보낸 것) (삭제된 scene 데이터를 리턴함 - 프론트에서 휴지통에 넣기 위함)	/scene/vote	POST	<pre>{ "gameId" : "String", "userId" : "String", "isAccepted" : true, "cardId" : 0 }</pre>	200 OK: { "success": true, "status": 200, "message": "투표 결과에 따라 삭제됨", "data": { "id": "2f73c46b-dee9-432f-b42c-2c84a81cdd4b", "gameId": "0b456664-7547-4316-b7c7-6c504fea8dad", "userId": "qwe", "sceneOrder": 1, "prompt": "유령이 죽었다.", "image": null }, "timestamp": "2025-02-

			<pre>08T23:15:14.859917800", "path": "/scene" } 400 Bad Request : { "success": false, "status": 400, "message": "아직 저장된 scene이 없습니다.", "data": null, "timestamp": "2025-02- 08T23:16:45.709507700", "path": "/scene" } 409 Bad Request : { "success": false, "status": 409 , "message": "투표 결과 찬성 으로 삭제되지 않음", "data": null, "timestamp": "2025-02- 08T23:16:45.709507700", "path": "/scene/vote" }</pre>
--	--	--	---

5. BOOK 관리 (베스트 셀러)

기능	URL	Method	Request Body	API Response
book 1개의 데이터 조회 (책 목록에서 하나를 클릭하 면 그 book 데이터를 불러 옴) 조회하면 조회수+1 이 요청을 통해 scene 데이 터를 가져올 수 있음(지연로 딩 때문)	/book	GET	param: id : "String"	<p>200 OK:</p> <pre>{ "success": true, "status": 200, "message": "book 데이터 반환 성공", "data": { "title": "book13", "sceneResponseList": [order : int, imageUrl : String, prompt : String], "bookCover": "string" }, "timestamp": "2025-02- 14T11:02:16.699760900", "path": "/book" }</pre> <p>404 NOT FOUND:</p> <pre>{ "success": false, "status": 404, "message": "해당 Id의 book이 없습니다.", "data": null,</pre>

				<pre>"timestamp": "2025-02-14T11:02:40.649596400", "path": "/book" }</pre>
책 리스트 페이지네이션 1,2,3위를 제외한 리스트들을 보여줌. mode 0 : 조회수 순 1 : 최신순	/book/{page}	GET	param: mode : int	<p>200 OK(조회수 순):</p> <pre>{ "success": true, "status": 200, "message": "book 데이터를 조회수순으로 불러왔습니다.", "data": [{ "id": "0475e722-64b9-4294-bea2-8c823ce972c6", "title": "book5", "imageUrl": "string", "scenes": [], "viewCnt": 0, "createdAt": "2025-02-13T17:12:58.011518" }, { "id": "0a523571-ef94-46d7-8ec0-66330d86e7a6", "title": "book6", "imageUrl": "string", "scenes": [], "viewCnt": 0, "createdAt": "2025-02-13T17:13:00.843859" }, { "id": "c491e0bd-a1b6-415a-8f8d-7453223ad53c", "title": "book9", "imageUrl": "string", "scenes": [], "viewCnt": 0, "createdAt": "2025-02-13T17:13:10.571183" }], "timestamp": "2025-02-14T12:25:53.655822300", "path": "/book/1" }</pre> <p>200 OK(최신순):</p> <pre>{ "success": true, "status": 200, "message": "book 데이터 최신순으로 불러왔습니다.", "data": [{ "id": "edb7bb9c-dd4a-4d76-aedc-93202257973c", "title": "최신 책", </pre>

				<pre>"imageUrl": "이미지주소", "scenes": [], "viewCnt": 0, "createdAt": "2025-02-14T09:30:29.260966" }, { "id": "28ae74ce-7327-42c5-99b7-3f0eb49cf6e3", "title": "book12", "imageUrl": "string", "scenes": [], "viewCnt": 0, "createdAt": "2025-02-13T17:13:19.748722" }, { "id": "21d762bc-384f-4309-8546-ce227aee306c", "title": "book11", "imageUrl": "string", "scenes": [], "viewCnt": 0, "createdAt": "2025-02-13T17:13:17.136616" }], "timestamp": "2025-02-14T12:57:02.409705400", "path": "/book/1" }</pre>
조회수 1,2,3위 책의 데이터 가져오기 (이 데이터로 표지, 제목 등 데이터 사용 가능) 안에 내용은 /book 으로 조회	/book/top3	GET		<pre>200 OK: { "success": true, "status": 200, "message": "1~3위 책 데이터 반환", "data": [{ "id": "f3b4346f-a4d3-4f8c-8e2e-199b1374d901", "title": "책3", "imageUrl": "책3", "scenes": [], "viewCnt": 6, "createdAt": "2025-02-14T12:59:50.769761" }, { "id": "1c54d553-3b82-4ba1-b03c-6bd99ea102d0", "title": "책1", "imageUrl": "책1", "scenes": [], "viewCnt": 3, "createdAt": "2025-02-14T12:59:42.053239" }, </pre>

			<pre>{ "id": "d718f211-5282-4a39-896f-79d9a320fee5", "title": "책2", "imageUrl": "책2", "scenes": [], "viewCnt": 0, "createdAt": "2025-02-14T12:59:46.734521" }], "timestamp": "2025-02-14T13:00:22.242433400", "path": "/book/top3" }</pre>
--	--	--	---