List of All Apps

Test case ID	Test case name	Description	Precondition	API End point	Request methoc	Request Headers	Test steps	Expected Result	
TC001	Verify successful retrieval with a valid API key	Verify that the API returns a 200 OK status and the correct response body when a valid API key is provided.	Valid X-API-Key is provided.	/api/v1/apps	GET	X-API-Key: valid-api-key	Send a request with a valid API key. Check the status code of the response. Verify the response body.	HTTP Status Code: 200 OK, and a list of apps is returned in JSON format with all fields correctly populated.	
TC002	Verify response contains expected fields for each app	Ensure that each app object contains the expected fields: key, name, description, logo, categories, and appId.	A valid X-API-Key is provided.	/api/v1/apps	GET	X-API-Key: valid-api-key	Send a valid request. Parse the response body. Check each item for the presence of fields: key, name, description, logo, categories, appId.	Each app object in the response contains all required fields correctly populated.	
TC003	Verify response time is within acceptable limits	Ensure that the API responds within an acceptable time frame.	A valid X-API-Key is provided, and the server is not under heavy load.	/api/v1/apps	GET	X-API-Key: valid-api-key	Send a valid request. Measure the response time.	The response time is less than 2 seconds.	
TC004	Verify response contains valid and non-empty fields	Ensure that the response returns all apps with valid and non-empty fields.	Valid X-API-Key is provided.	/api/v1/apps	GET	X-API-Key: valid-api-key	Send a valid request. Validate that each app has non-empty values for all fields.	All fields in each app object are non-empty and valid.	
TC005	Verify uniqueness of appId	Ensure that the appId is unique across all items in the response.	Valid X-API-Key is provided.	/api/v1/apps	GET	X-API-Key: valid-api-key	Send a valid request. Check that all appIds are unique.	All appId values are unique.	
TC006	Verify case-insensitivity of categories	Verify that the API is case-insensitive when returning the categories.	Valid X-API-Key is provided.	/api/v1/apps	GET	X-API-Key: valid-api-key	Send a GET request to the endpoint. Check the case of category values in the response.	Categories are returned regardless of their case in the database, and matched correctly.	

Negative Test Cases (list of All Apps)									
TC007	Verify 401 Unauthorized for invalid/missing API key	Verify that the API returns a 401 Unauthorized status when an invalid or missing API key is provided.	None	/api/v1/apps	GET	X-API-Key: invalid-api-key	Send a GET request with an invalid or missing API key.. Verify the response status code.	The API returns a 401 Unauthorized status.	
TC008	Verify 404 Not Found for invalid endpoint	Verify that the API returns a 404 Not Found status when an invalid endpoint is accessed.	None	/api/v1/nonexistent	GET	X-API-Key: valid-api-key	Send a GET request to an invalid endpoint. Verify the response status code.	The API returns a 404 Not Found status.	
TC009	Verify error for unsupported HTTP method	Verify that the API returns an appropriate error message when a request is made with an unsupported HTTP method (e.g., POST).	None	/api/v1/apps	POST	X-API-Key: valid-api-key	Send a POST request to the endpoint. Verify the response error message.	The API returns an error indicating that the method is not allowed.	
TC010	Verify 400 Bad Request for invalid query parameters	Verify that the API returns a 400 Bad Request status when invalid query parameters are added to the request.	None	/api/v1/apps?invalidparam=value	GET	X-API-Key: valid-api-key	1. Send a GET request with invalid query parameters. 2. Verify the response status code.	The API returns a 400 Bad Request status.	
TC011	Verify 500 Internal Server Error for server down	Verify that the API returns a 500 Internal Server Error when the server is down.	Server down	/api/v1/apps	GET	X-API-Key: valid-api-key	1. Simulate a server down situation. 2. Send a GET request. 3. Verify the response status code.	The API returns a 500 Internal Server Error	
TC012	Verify empty list response for no apps	Verify that the API returns an empty list if there are no apps available in the system.	No apps available	/api/v1/apps	GET	X-API-Key: valid-api-key	1. Ensure there are no apps in the system. 2. Send a GET request. 3. Verify the response.	The API returns an empty list.	
List of all Connected Accounts									

TC013	Validate 200 Status Code	Verify the API returns a 200 status code	Valid API key	/api/v1/connectedAccounts	GET	X-API-Key: <api-key>	Send GET request Validate 200 Status Code	200 OK	
TC014	Validate Response Structure	Verify the response contains the correct fields	Valid API key	/api/v1/connectedAccounts	GET	X-API-Key: <api-key>	Send GET request Validate Response Structure	Response includes items array with correct fields	
TC015	Validate Unique IDs	Verify that each id is unique	Valid API key	/api/v1/connectedAccounts	GET	X-API-Key: <api-key>	Send GET request Validate Unique IDs	Each id is unique	
TC016	Validate Connector ID,Status Field,CreatedAt Field,UpdatedAt Field, AppName, Logo URL	Verify that each connectorId is a valid UUID Verify that the status field contains valid values Verify that the createdAt field is a valid timestamp Verify that the updatedAt field is a valid timestamp Verify that appName is correctly populated Verify that the logo URL is valid	Valid API key	/api/v1/connectedAccounts	GET	X-API-Key: <api-key>	Send GET request Validate Connector ID	connectorId is a valid UUID Status is one of the expected values (INITIATED, ACTIVE) createdAt is a valid ISO timestamp updatedAt is a valid ISO timestamp appName matches expected app name logo is a valid URL	
Negative Test Cases(List of All Connected Accounts)									
TC017	Invalid API Key	Verify the API returns a 401 status code with an invalid API key	Invalid API key	/api/v1/connectedAccounts	GET	X-API-Key: invalid-key	Send GET request	401 Unauthorized	
TC018	Missing API Key	Verify the API returns a 403 status code when the API key is missing	No API key	/api/v1/connectedAccounts	GET	None	Send GET request	403 Forbidden	
TC019	Non-existent Endpoint	Verify the API returns a 404 status code for a non-existent endpoint	Valid API key	/api/v1/invalidEndpoint	GET	X-API-Key: <api-key>	Send GET request	404 Not Found	
TC020	Invalid HTTP Method	Verify the API returns a 400 status code when using an invalid HTTP method	Valid API key	/api/v1/connectedAccounts	POST	X-API-Key: <api-key>	Send POST request	400 Bad Request	

TC021	Missing Required Fields	Verify the API returns an error when required fields are missing in the request	Valid API key	/api/v1/connectedAccounts	GET	X-API-Key: <api-key>	Send GET request with missing fields	Error response due to missing fields	
Connect An Account									
TC022	Valid Integration ID	Verify the API with a valid integrationId and redirectUri.	API Key valid	/api/v1/connectedAccounts	POST	X-API-Key: <api-key>	Send POST request with valid integrationId and redirectUri.	Response should return status 200 with connectionStatus as INITIATED and valid connectionId, redirectUrl.	
TC023	Valid Redirect URI	Check the API with a valid redirectUri pointing to Google.	API Key valid	/api/v1/connectedAccounts	POST	X-API-Key: <api-key>	Send POST request with a valid redirectUri (https://google.com).	Response should contain redirectUrl based on the redirectUri provided.	
TC024	Minimum Length Integration ID	Verify the API with the minimum length of integrationId.	API Key valid	/api/v1/connectedAccounts	POST	X-API-Key: <api-key>	Send POST request with the minimum length integrationId (1 character).	Response should return status 400 with an error message regarding the minimum length violation.	
TC025	Maximum Length Redirect URI	Check the API with the maximum allowed length for redirectUri.	API Key valid	/api/v1/connectedAccounts	POST	X-API-Key: <api-key>	Send POST request with the maximum length redirectUri (e.g., 2048 characters).	Response should return status 200 and the connection should be initiated successfully.	
TC026	Successful Account Connection	Verify successful initiation of account connection.	API Key valid	/api/v1/connectedAccounts	POST	X-API-Key: <api-key>	Send POST request with valid integrationId and redirectUri.	Response should return connectionStatus as INITIATED, valid connectionId, and correct redirectUrl.	
Negative Test Cases(Connect an Account)									
TC027	Missing Integration ID	Verify the response when the integrationId is missing.	API Key valid	/api/v1/connectedAccounts	POST	X-API-Key: <api-key>	Send POST request without integrationId in the request body.	Response should return status 400 with an error message indicating missing required field.	

TC028	Invalid API Key	Verify the API with an invalid API Key.	None	/api/v1/connectedAccounts	POST	X-API-Key: <invalid-api-key>	Send POST request with an invalid API Key.	Response should return status 401 with an unauthorized access message.	
Get a Connected Account									
TC029	Valid Connected Account ID	Verify retrieval of details for a valid connectedAccountId	A valid connectedAccountId exists	/api/v1/connectedAccounts/{connectedAccountId}	GET	X-API-Key: <api-key>	1. Send a GET request with a valid connectedAccountId 2. Observe the response.	200 OK with account details returned in the response body	
TC030	Valid API Key	Verify access with a valid API key	A valid connectedAccountId exists	/api/v1/connectedAccounts/{connectedAccountId}	GET	X-API-Key: <valid-api-key>	1. Send a GET request with a valid connectedAccountId and valid API key 2. Observe the response.	200 OK with account details returned in the response body	
TC031	Recently Updated Account	Verify retrieval of recently updated account details	A valid, recently updated account	/api/v1/connectedAccounts/{connectedAccountId}	GET	X-API-Key: <api-key>	1. Send a GET request for an account that was recently updated 2. Observe the response.	200 OK with updated account details reflected in the response body	
TC032	Retrieve Connected Account Details	Verify functionality to retrieve connected account details	A valid connectedAccountId exists	/api/v1/connectedAccounts/{connectedAccountId}	GET	X-API-Key: <api-key>	1. Send a GET request for an existing connectedAccountId 2. Observe the returned account details.	200 OK with correct details in the response	
TC033	Check status Field	Verify the status field in the returned connected account	A valid connectedAccountId exists	/api/v1/connectedAccounts/{connectedAccountId}	GET	X-API-Key: <api-key>	1. Send a GET request for an existing connectedAccountId 2. Verify the status field in the response.	status field reflects the correct account status	
Negative test cases (Get a Connected Account)									
TC034	Invalid Connected Account ID	Verify behavior when an invalid connectedAccountId is provided	None	/api/v1/connectedAccounts/{invalidId}	GET	X-API-Key: <api-key>	1. Send a GET request with an invalid connectedAccountId 2. Observe the response.	404 Not Found with appropriate error message	

TC035	Missing API Key	Verify behavior when API key is missing	A valid connectedAccountId exists	/api/v1/connectedAccounts/{connectedAccountId}	GET	None	1. Send a GET request without providing the X-API-Key header 2. Observe the response.	401 Unauthorized with an error message about missing credentials	
TC036	Invalid API Key	Verify behavior when an invalid API key is provided	A valid connectedAccountId exists	/api/v1/connectedAccounts/{connectedAccountId}	GET	X-API-Key: invalid-key	1. Send a GET request with an invalid X-API-Key header 2. Observe the response.	401 Unauthorized with an error message about invalid credentials	
Delete a Connected Account									
TC037	Valid Connected Account ID	Verify deletion of a valid connected account	A valid connectedAccountId exists	/api/v1/connectedAccounts/{connectedAccountId}	DELETE	X-API-Key: <api-key>	1. Send a DELETE request with a valid connectedAccountId 2. Observe the response.	200 OK with "status": "success"	
TC038	Recently Created Account	Verify deletion of a recently created connected account	A valid, recently created account	/api/v1/connectedAccounts/{connectedAccountId}	DELETE	X-API-Key: <api-key>	1. Send a DELETE request for a recently created connectedAccountId 2. Observe the response.	200 OK with "status": "success"	
TC039	Deletion with Valid API Key	Verify deletion with a valid API key	A valid connectedAccountId exists	/api/v1/connectedAccounts/{connectedAccountId}	DELETE	X-API-Key: <valid-api-key>	1. Send a DELETE request with a valid API key and connectedAccountId 2. Observe the response.	200 OK with "status": "success"	
TC040	Delete Connected Account	Verify functionality to delete a connected account	A valid connectedAccountId exists	/api/v1/connectedAccounts/{connectedAccountId}	DELETE	X-API-Key: <api-key>	1. Send a DELETE request for an existing connectedAccountId 2. Observe the response.	200 OK with "status": "success"	
TC041	Deletion of Already Deleted Account	Verify behavior when trying to delete an already deleted account	A deleted connectedAccountId	/api/v1/connectedAccounts/{deletedAccountId}	DELETE	X-API-Key: <api-key>	1. Send a DELETE request for an already deleted connectedAccountId 2. Observe the response.	404 Not Found or appropriate error message	

Negative test cases(Delete a Connected Account)									
TC042	Invalid Connected Account ID	Verify behavior when an invalid connectedAccountId is provided	None	/api/v1/connectedAccounts/{invalidId}	DELETE	X-API-Key: <api-key>	1. Send a DELETE request with an invalid connectedAccountId 2. Observe the response.	404 Not Found with an error message	
TC043	Missing API Key	Verify behavior when API key is missing	A valid connectedAccountId exists	/api/v1/connectedAccounts/{connectedAccountId}	DELETE	None	1. Send a DELETE request without providing the X-API-Key header 2. Observe the response.	401 Unauthorized with an error message	
TC044	Invalid API Key	Verify behavior when an invalid API key is provided	A valid connectedAccountId exists	/api/v1/connectedAccounts/{connectedAccountId}	DELETE	X-API-Key: invalid-key	1. Send a DELETE request with an invalid X-API-Key header 2. Observe the response.	401 Unauthorized with an error message	
Get an Integration									
TC045	Valid Integration ID	Fetch integration data using a valid ID	Valid Integration ID present	/api/v1/integrations/{integrationId}	GET	X-API-Key: <api-key>	1. Send GET request with valid Integration ID 2. Validate response	200 OK with valid integration data	
TC046	Valid Response Structure	Check if response structure is correct	Valid Integration ID present	/api/v1/integrations/{integrationId}	GET	X-API-Key: <api-key>	1. Send GET request 2. Validate JSON structure	Response contains keys: id, name, authScheme	
TC047	Integration Enabled Status	Verify the enabled status of integration	Valid Integration ID present	/api/v1/integrations/{integrationId}	GET	X-API-Key: <api-key>	1. Send GET request 2. Validate the enabled field	enabled field matches expected value	
TC048	Authentication Scheme	Validate correct auth scheme is returned	Valid Integration ID present	/api/v1/integrations/{integrationId}	GET	X-API-Key: <api-key>	1. Send GET request 2. Validate authScheme field	authScheme matches the expected value	
TC049	ID Format Validation	Validate the format of Integration ID	None	/api/v1/integrations/invalidFormatID	GET	X-API-Key: <api-key>	1. Send GET request with invalid format 2. Validate response	400 Bad Request with validation error	
TC050	API Key Validation	Validate API key with correct length	API key has correct length	/api/v1/integrations/{integrationId}	GET	X-API-Key: <api-key>	1. Send GET request with valid API key 2. Validate response	200 OK with valid integration data	

Negative test cases (Get an integration)									
TC051	Invalid Integration ID	Fetch data with an invalid ID	Integration ID not present	/api/v1/integrations/invalidId	GET	X-API-Key: <api-key>	1. Send GET request with invalid ID 2. Validate error response	404 Not Found with error message	
TC052	Missing API Key	Check API behavior without API key	None	/api/v1/integrations/{integrationId}	GET	None	1. Send GET request without API key 2. Validate error response	401 Unauthorized	
Update an Integration									
TC053	Valid Update Request	Update the enabled field to false	Valid Integration ID present	/api/v1/integrations/{integrationId}	PATCH	Content-Type: application/json, X-API-Key	1. Send PATCH request with valid ID and request body 2. Validate response	200 OK with "status": "success"	
TC054	Validate enabled field	Ensure enabled field is updated correctly	Integration is enabled currently	/api/v1/integrations/{integrationId}	PATCH	Content-Type: application/json, X-API-Key	1. Send PATCH request with enabled: false 2. Verify the change in the database or subsequent GET request to confirm the update	enabled field is false	
TC055	Enable to Disable	Toggle enabled field from true to false	Integration is enabled	/api/v1/integrations/{integrationId}	PATCH	Content-Type: application/json, X-API-Key	1. Send PATCH request with enabled: false 2. Validate that the enabled field is set to false	"status": "success"	
TC056	Disable to Enable	Toggle enabled field from false to true	Integration is disabled	/api/v1/integrations/{integrationId}	PATCH	Content-Type: application/json, X-API-Key	1. Send PATCH request with enabled: true 2. Validate that the enabled field is set to true	"status": "success"	
TC057	Missing enabled field	Send request without the enabled field	None	/api/v1/integrations/{integrationId}	PATCH	Content-Type: application/json, X-API-Key	1. Send PATCH request without the enabled field 2. Validate error response	400 Bad Request	
Negative Test Cases- Update an integration									

TC058	Invalid Integration ID	Attempt to update with an invalid ID	Invalid Integration ID used	/api/v1/integrations/invalidId	PATCH	Content-Type: application/json, X-API-Key	1. Send PATCH request with invalid ID 2. Validate error response	404 Not Found or error message	
TC059	Missing API Key	Test API without API Key	None	/api/v1/integrations/{integrationId}	PATCH	Content-Type: application/json	1. Send PATCH request without API key 2. Validate error response	401 Unauthorized	
(List of all Integrations- as same as list of all apps)									
Create a new Integration									
TC060	Valid POST Request	Create an integration with valid data	Valid API Key	/api/v1/integrations	POST	Content-Type: application/json, X-API-Key	1. Send POST request with valid data 2. Validate response and new integration details	201 Created with integration details	
TC061	Minimal Required Fields	Create integration with minimal required fields	Valid API Key	/api/v1/integrations	POST	Content-Type: application/json, X-API-Key	1. Send POST request with minimal required fields 2. Validate response	201 Created with integration details	
TC062	Validates OAuth2 Scheme	Ensure integration uses OAuth2 scheme	Valid API Key	/api/v1/integrations	POST	Content-Type: application/json, X-API-Key	1. Send POST request with OAuth2 scheme 2. Validate OAuth2 fields are correctly set	201 Created with OAuth2 configuration	
TC063	Verify Integration Creation	Ensure that integration is created and details are correct	Valid API Key	/api/v1/integrations	POST	Content-Type: application/json, X-API-Key	1. Send POST request with valid data 2. Validate that the new integration is correctly created and accessible via GET request	201 Created with correct details	
TC064	Ensure OAuth2 Fields	Validate that OAuth2 specific fields are correctly set	Valid API Key	/api/v1/integrations	POST	Content-Type: application/json, X-API-Key	1. Send POST request with OAuth2 settings 2. Validate that client_id, client_secret, and base_url are set correctly	201 Created with OAuth2 configuration	
Negative Test Cases (Create a New Integration)									

TC065	Missing Required Field	Send POST request with a missing required field	Valid API Key	/api/v1/integrations	POST	Content-Type: application/json, X-API-Key	1. Send POST request missing the name field 2. Validate error response	400 Bad Request with error message	
TC066	Invalid API Key	Attempt to create integration with an invalid API key	Invalid API Key	/api/v1/integrations	POST	Content-Type: application/json, X-API-Key	1. Send POST request with invalid API key 2. Validate error response	401 Unauthorized	
TC067	Invalid JSON Format	Send request with improperly formatted JSON	Valid API Key	/api/v1/integrations	POST	Content-Type: application/json, X-API-Key	1. Send POST request with invalid JSON format 2. Validate error response	400 Bad Request with error message	
Get a Specific Action									
TC068	Valid Action Name	Retrieve details of a valid action	Valid API Key	/api/v1/actions/CreateSubtask	GET	X-API-Key: <api-key>	1. Send GET request with a valid action name 2. Validate the response contains correct action details	200 OK with correct action details	
TC069	Valid Action with All Params	Retrieve details with all possible parameters	Valid API Key	/api/v1/actions/CreateSubtask	GET	X-API-Key: <api-key>	1. Send GET request with a valid action and all parameters 2. Validate the response contains correct details	200 OK with complete action details	
TC070	Verify Action Details	Ensure the action details match the expected configuration	Valid API Key	/api/v1/actions/CreateSubtask	GET	X-API-Key: <api-key>	1. Send GET request with valid action name 2. Validate the returned details match the expected configuration	200 OK with correct action details	
TC071	Validate Response Structure	Ensure the response structure is correct	Valid API Key	/api/v1/actions/CreateSubtask	GET	X-API-Key: <api-key>	1. Send GET request with valid action name 2. Validate the structure of the response (e.g., parameters, response, etc.)	200 OK with correct response structure	

TC072	Invalid Action Name Format	Test with invalid format for action name (e.g., numbers only)	Valid API Key	/api/v1/actions/123456	GET	X-API-Key: <api-key>	1. Send GET request with action name in an invalid format 2. Validate the response contains an appropriate error message	400 Bad Request with error message	
TC073	Missing Required Parameters	Test response when required parameters are missing from the action details	Valid API Key	/api/v1/actions/CreateSubtask	GET	X-API-Key: <api-key>	1. Send GET request with valid action name 2. Validate that response correctly indicates any missing parameters	400 Bad Request with error message	
Negative Cases (get a specific action)									
TC074	Invalid Action Name	Try retrieving details using an invalid action name	Valid API Key	/api/v1/actions/InvalidAction	GET	X-API-Key: <api-key>	1. Send GET request with an invalid action name 2. Validate the response contains an appropriate error message	404 Not Found with error message	
TC075	Missing API Key	Try retrieving action details without providing the API Key	No API Key	/api/v1/actions/CreateSubtask	GET	None	1. Send GET request without the API key 2. Validate the response contains an appropriate error message	401 Unauthorized with error message	
TC076	Empty Action Name	Try retrieving action details with an empty action name	Valid API Key	/api/v1/actions/	GET	X-API-Key: <api-key>	1. Send GET request with an empty action name 2. Validate the response contains an appropriate error message	400 Bad Request with error message	
Get List of Actions(Same as List of Apps)									
Execute action									

TC077	Retrieve All Actions	Fetch all actions and verify successful retrieval	Valid API Key	/api/v1/actions	GET	X-API-Key: <api-key>	1. Send GET request 2. Validate the response status and content	200 OK with list of actions and metadata	
TC078	Actions List Pagination	Verify pagination details in the response	Valid API Key	/api/v1/actions	GET	X-API-Key: <api-key>	1. Send GET request 2. Check if page and totalPages fields are correctly populated	200 OK with correct pagination info	
TC079	Verify Enabled Actions	Check if enabled field is correctly set to true	Valid API Key	/api/v1/actions	GET	X-API-Key: <api-key>	1. Send GET request 2. Validate enabled field for actions	enabled field is true for actions	
TC080	Verify Response Structure	Ensure the structure of the response is correct	Valid API Key	/api/v1/actions	GET	X-API-Key: <api-key>	1. Send GET request 2. Validate response structure (e.g., items, page, totalPages)	200 OK with correct response structure	
TC081	Check Action Metadata	Verify if action metadata such as logo and appName are correct	Valid API Key	/api/v1/actions	GET	X-API-Key: <api-key>	1. Send GET request 2. Validate action metadata like logo and appName are correct	200 OK with correct metadata	
Negative Test cases(Execute actions)									
TC082	Missing API Key	Attempt to retrieve actions without API key	No API Key	/api/v1/actions	GET	None	1. Send GET request without API key 2. Validate the response contains an appropriate error message	401 Unauthorized with error message	
TC083	Invalid API Key	Attempt to retrieve actions with an invalid API key	Invalid API Key	/api/v1/actions	GET	X-API-Key: invalid-key	1. Send GET request with invalid API key 2. Validate the response contains an appropriate error message	401 Unauthorized with error message	

TC084	No Actions Available	Verify behavior when no actions are available	No actions in the system	/api/v1/actions	GET	X-API-Key: <api-key>	1. Send GET request when no actions are present 2. Validate the response reflects no data available	200 OK with empty list and totalPages as 0	