> This collection features end-to-end tests that ensure the Intergalactic Bank API components function in an expected sequence. Check Description out the [Functional testing collection](https://www.postman.com/templates/f26ad070-d626-4d75-b151-7cbf1a48ed11/Functional-testing) and [Integration testing collection](https://www.postman.com/templates/6f788c17-067a-4074-a528-d07df1df9529/Integration-testing-%23example) for other test cases for this API. ## ** Set started** The fictional Intergalactic Bank Services team is responsible for testing the end-to-end workflows in this collection. You can use this collection to validate specific workflows for your team or figure out ways to increase your test coverage and make your applications more resilient. To test the Intergalactic Bank API, you can: 1. **Review tests and scripts:** Check out the [tests](https://learning.postman.com/docs/writing-scripts/test-scripts/) in the Scripts tab of a collection, folder, or request. You'll also find additional steps in the Pre-request Script, such as for passing data between requests or building up POST request payloads. 2. **Run tests:** To run the collection, select Run collection from the collection menu. If you are interested in testing one of the use cases, select Run folder from the use case folder menu. Keep in mind that many of these tests contain logic that passes information between requests, and they are meant to be run in order from top to bottom. If you send the requests manually or out of order, your results may vary. 3. **Review test results:** Many of the test cases contain error messages to help debug service errors. If you're using this collection as a pre-merge check, ensure that all tests are passing before submitting (and no, deleting the failing test cases isn't an option 😉). Check out the additional sections below or select View complete documentation. For context-sensitive help with a folder or request, look for the documentation icon. ## Method your team can use this collection - **Add a mock server to speed up development:** Adding functionality to this service? Since this collection has example responses, you can add a mock server that will return the expected results. Start writing tests against the mock, and once your additions are live, you can swap out the URLs and have a live test suite. - **Integrate tests with your CI/CD pipeline:** This comprehensive test suite is a great addition to your existing CI/CD pipeline. Generate the Postman CLI command from the Collection Runner and add this step to your existing checks each time you make a commit affecting this service. - **Add a monitor:** If your team is contributing to or relying on this service, it's a good idea to keep tabs on the status. Adding a monitor to this collection can give you confidence that the service is up and running and alert you to any breaking changes that may have occurred. ## Q What we're testing - A very common workflow is the transfer of funds between accounts. We have two scenarios: one which creates all resources and successfully transfers funds, and one which tries the transfer with insufficient funds. - During these workflows, we are testing for data correctness, data types, and that data is successfully passed through the system during creation and retrieval. ## 🔒 A note on authorization This API uses an API key-based authorization. In this collection, we set the authorization at the collection level and generate a new API Key for each test run. ## 👀 View and share run results Interested in seeing previous run results? We're happy to share, and have a few ways for you to stay in the loop: - **Runs tab:** View past collection runs in the `Runs` tab of this collection. Here, you can see the results of past runs, including who ran them and whether it was via the Collection Runner or the CLI. You can also share individual run reports from this page. - **Monitors:** If you have a monitor set up for this collection, you can see historical run information when you select the Monitors tab in the left sidebar. You can also have the results sent to Slack or other services when you set up a corresponding integration. ### > Increase test coverage The Intergalactic Bank Services team wants your team to have everything they need to successfully test our services with their applications. Is your team utilizing this API in a use case not represented here? Reach out to us on our internal Slack by tagging `@bank-services`. Or add an [inline comment] (https://learning.postman.com/docs/collaborating-in-postman/working-with-your-team/discussing-your-work/) to this collection with your test case suggestions. Time Sun Sep 14 2025 03:25:04 GMT+0600 (Bangladesh Standard Time) Exported with Newman v6.2.1 **Failed Total** 0 1 Iterations 21 0 Requests 63 0 Prerequest Scripts 63 0 Test Scripts 73 0 Assertions Total run duration 10.4s Total data received 1.79KB (approx) Average response time 402ms 0 **Total Failures** Requests **Process a VALID transaction** [Setup][1] Generate a New API Key This request generates a new API key and saves it to the collection variables if the request is successful. Description Method **GET** URL https://template.postman-echo.com/api/v1/auth 2.6s Mean time per request 43B Mean size per request Total passed tests 4 Total failed tests 0 200 Status code Tests Fail count Name Pass count Folder-level status check 0 1 0 Should return a object 1 Should contain the 'apiKey' property in the response 1 0 1 0 Should return a string for the 'apiKey' property [2] Create A New Account (FromAccount) Generates a new account ('fromAccount'), that the funds will be transferred from, and saves the following Description information if the request is successful: - Account owner from the request body (saved as collection variable `e2e_user_name_from`) - Account id from the response body (saved as collection variable `e2e_user_account_id_from`) Method **POST** URL https://template.postman-echo.com/api/v1/accounts 289ms Mean time per request 26B Mean size per request Total passed tests 4 Total failed tests 0 200 Status code Tests Pass count Fail count Name 0 Folder-level status check 1 Should have a valid JSON format response body 0 1 Should have the required 'account' object in the response 1 0 Should have the 'id' property in the account object and is a non-negative integer 0 1 [3] Create A New Account (ToAccount) Description Generates another new account ('toAccount'), that the funds will be transferred to, and saves the following information if the request is successful: - Account owner from the request body (saved as collection variable `e2e_user_name_to`) - Account id from the response body (saved as collection variable `e2e_user_account_id_to`) **POST** Method URL https://template.postman-echo.com/api/v1/accounts Mean time per request 290ms Mean size per request 26B Total passed tests Total failed tests 0 Status code 200 Tests Pass count Fail count Name Folder-level status check 0 1 Should have a valid JSON format response body 1 0 Should have the required 'account' object in the response 0 1 Should have the 'id' property in the account object and is a non-negative integer 0 1 [4] Get All User Accounts To Confirm Creation Of Users Description Get a list of all accounts to ensure the accounts were created and saved correctly. Method **GET** URL https://template.postman-echo.com/api/v1/accounts 289ms Mean time per request 272B Mean size per request Total passed tests 4 Total failed tests 0 200 Status code Tests Fail count Name Pass count Folder-level status check 1 0 Should contain an accounts 'array' in the response 1 0 Should return the correct account data for the 'fromAccount' user 1 0 0 Should return the correct account data for the 'toAccount' user [5] Create A New Transaction Between The Users Description Create a new transaction between the `fromAccount` and `toAccount` **POST** Method URL https://template.postman-echo.com/api/v1/transactions 300ms Mean time per request 44B Mean size per request 2 Total passed tests Total failed tests 0 200 Status code Tests Name Pass count Fail count Folder-level status check 1 0 1 0 Should display a successful transaction message [6] Get All Transactions To The ToAccount Description Get a list of all transactions to and from the 'toAccount' Method **GET** URL https://template.postman-echo.com/api/v1/transactions?toAccountId=2952644 287ms Mean time per request 155B Mean size per request Total passed tests 3 0 Total failed tests Status code 200 Tests Pass count Name Fail count Folder-level status check 1 0 Should contain a transactions 'array' in the response 1 0 1 0 Should return the correct transaction data for the 'toAccount' user [7] Get A Single Transaction Get the details of an individual transaction, using the first transaction id from the list of transactions obtained in Description step \[6\]. Method **GET** URL https://template.postman-echo.com/api/v1/transactions/793358 285ms Mean time per request 152B Mean size per request 3 Total passed tests lotal failed tests Status code 200 Tests Fail count Name Pass count Folder-level status check 0 1 Should contain a transaction 'object' in the response 1 0 Should return the correct transaction data for the 'toAccount' user 1 0 [8] Get Users Current Balance - ToAccount Description Confirm the balance of the `toAccount` matches the expected value after the transaction. Method **GET** URL https://template.postman-echo.com/api/v1/accounts/2952644 Mean time per request 290ms Mean size per request 140B Total passed tests 6 0 Total failed tests Status code 200 Tests Pass count Name Fail count Folder-level status check 0 1 0 Should contain a account 'object' in the response 1 Should contain the correct account id in the response 1 0 0 Should contain the correct owner in the response 1 1 0 Should contain the correct balance in the response Should contain the correct currency in the response 0 1 [9] Get Users Current Balance - FromAccount Confirm the balance of the 'fromAccount' matches the expected value after the transaction. Description Method **GET** URL https://template.postman-echo.com/api/v1/accounts/2952643 Mean time per request 286ms Mean size per request 141B Total passed tests 6 0 Total failed tests Status code 200 Tests Fail count Name Pass count Folder-level status check 1 0 0 Should contain a account 'object' in the response 1 1 0 Should contain the correct account id in the response 1 0 Should contain the correct owner in the response 0 Should contain the correct balance in the response 1 0 Should contain the correct currency in the response 1 [Teardown][10] Delete The User Account - ToAccount Description Delete the `toAccount` account to clean up after this test case. DELETE Method URL https://template.postman-echo.com/api/v1/accounts/2952644 Mean time per request 291ms Mean size per request 16B Total passed tests Total failed tests 0 Status code 200 Tests Name Pass count Fail count Folder-level status check 1 0 [Teardown][11] Delete The User Account - FromAccount Description Delete the `fromAccount` account and clear out the collection variables we created during this test case. Method DELETE URL https://template.postman-echo.com/api/v1/accounts/2952643 Mean time per request 302ms Mean size per request 16B Total passed tests 1 Total failed tests 0 Status code 200 Tests Name Pass count Fail count 0 Folder-level status check 1 Attempt an INVALID transaction [Setup][1] Generate a New API Key Description This request generates a new API Key, and saves it to the collection variables if the request was successful. Method **GET** URL https://template.postman-echo.com/api/v1/auth Mean time per request 285ms Mean size per request 43B Total passed tests 4 0 Total failed tests Status code 200 Tests Fail count Name Pass count 0 Status check 1 0 Should return a object 1 Should contain the 'apiKey' property in the response 1 0 0 Should return a string for the 'apiKey' property 1 [2] Create A New Account (FromAccount) Description Generates a new account ('fromAccount'), that the funds will be transferred from, and saves the following information if the request is successful: - Account owner from the request body (saved as collection variable `e2e_user_name_from`) - Account id from the response body (saved as collection variable `e2e_user_account_id_from`) **POST** Method URL https://template.postman-echo.com/api/v1/accounts 290ms Mean time per request 26B Mean size per request Total passed tests 4 Total failed tests 0 200 Status code Tests Pass count Fail count Name Status check 1 0 Should have a valid JSON format response body 1 0 0 Should have the required 'account' object in the response 1 0 Should have the 'id' property in the account object and is a non-negative integer 1 [3] Create A New Account (ToAccount) Description Generates another new account ('toAccount'), that the funds will be transferred to, and saves the following information if the request is successful: - Account owner from the request body (saved as collection variable `e2e_user_name_to`) - Account id from the response body (saved as collection variable `e2e_user_account_id_to`) **POST** Method URL https://template.postman-echo.com/api/v1/accounts Mean time per request 287ms Mean size per request 26B Total passed tests 4 0 Total failed tests Status code 200 Tests Name Pass count Fail count Status check 0 1 Should have a valid JSON format response body 0 1 0 Should have the required 'account' object in the response 1 Should have the 'id' property in the account object and is a non-negative integer 0 1 [4] Get All User Accounts To Confirm Creation Of Users Description Get a list of all accounts to ensure the accounts were created and saved correctly. Method **GET** URL https://template.postman-echo.com/api/v1/accounts Mean time per request 286ms 273B Mean size per request Total passed tests 4 Total failed tests 0 200 Status code Tests Pass count Fail count Name Status check 0 1 Should contain an accounts 'array' in the response 1 0 Should return the correct account data for the 'fromAccount' user 1 0 Should return the correct account data for the 'toAccount' user 1 0 [5] Create A New Transaction with insufficient funds Description Create a new transaction between the 'fromAccount' and 'toAccount'. We are expecting this to fail as there are insufficient funds in the `fromAccount`. **POST** Method URL https://template.postman-echo.com/api/v1/transactions 297ms Mean time per request 101B Mean size per request Total passed tests 3 0 Total failed tests 400 Status code Tests Name Pass count Fail count Status code is 400 0 1 0 Should return an error: invalidBody 1 0 Response has the required fields 1 [6] Get All Transactions To The ToAccount Description Get a list of transactions to and from the `toAccount`, and ensure there are no transactions. Method **GET** URL https://template.postman-echo.com/api/v1/transactions?toAccountId=2952646 287ms Mean time per request Mean size per request 19B Total passed tests 2 0 Total failed tests 200 Status code Tests Name Pass count Fail count Status check 0 1 Should contain an empty transactions 'array' in the response 1 0 [7] Check ToAccount Balance Unchanged Description Get more information about the `toAccount`, and check that the account balance remains unchanged after the failed transaction. **GET** Method https://template.postman-echo.com/api/v1/accounts/2952646 URL Mean time per request 289ms 140B Mean size per request Total passed tests 6 Total failed tests 0 200 Status code Tests Fail count Pass count Name 0 Status check 1 Should contain a account 'object' in the response 1 0 Should contain the correct account id in the response 0 1 1 0 Should contain the correct owner in the response Should contain the correct balance in the response 1 0 Should contain the correct currency in the response 1 0 [8] Check FromAccount Balance Unchanged Description Get more information about the `fromAccount`, and confirm the balance of the `toAccount` remains unchanged after the failed transaction. Method **GET** URL https://template.postman-echo.com/api/v1/accounts/2952645 Mean time per request 285ms Mean size per request 141B Total passed tests 6 Total failed tests 0 Status code 200 Tests Name Pass count Fail count Status check 0 1 1 0 Should contain a account 'object' in the response 0 Should contain the correct account id in the response 1 Should contain the correct owner in the response 1 0 0 Should contain the correct balance in the response 1 Should contain the correct currency in the response 1 0 [Teardown][9] Delete The User Account - ToAccount Delete the `toAccount` account to clean up after this test case. Description Method DELETE URL https://template.postman-echo.com/api/v1/accounts/2952646 Mean time per request 296ms Mean size per request 16B Total passed tests 1 Total failed tests 0 Status code 200 Tests Pass count Fail count Name Status check 1 0 [Teardown][10] Delete The User Account - FromAccount Description Delete the `fromAccount` account and clear out the collection variables we created during this test case. Method DELETE URL https://template.postman-echo.com/api/v1/accounts/2952645 Mean time per request 296ms Mean size per request 16B Total passed tests 1 Total failed tests 0 Status code 200 Tests Name Pass count Fail count Status check 1 0

Newman Report

End-to-End Tests

Collection