TUUM API Test Plan

Purpose: This document includes strategy and approach for testing Tuum Sandbox API. Testing will be conducted using Postman. The test plan tests the flow from creating a person to creating an account for the person to creating the first transaction and checking the balance of the person's account.

Scope

The scope of this API test plan includes testing the following:

- Validation of API endpoints
- Error handling and response codes.
- Security testing (authorisation)

Test Environment

- API Base URLs:
  - https://auth-api.sandbox.tuumplatform.com
  - https://person-api.sandbox.tuumplatform.com
  - https://account-api.sandbox.tuumplatform.com
- Postman Collection: Included in repository as a JSON file

Test Execution Flow

The tests will be executed in the following order:

1. Setup Pre-Conditions → Authentication
2. Test Execution → Use the provided Postman Collection to run tests
3. Negative Scenarios
4. Post Execution → Conclusion

Test Scenarios & Test Cases

Precondition:

- Authorise the employee. To do so, open the Postman collection and from the collection, open the Authentication API folder. In the folder, click on **POST** Authorise Employee. Click SEND. When the post-request script is run, the token is automatically saved under the collection variables and will be used in every other request forward.

API Endpoint	Method	Test Scenario	Expected Result	Status (Pass/Fail)
/api/v1/employees/authorise	POST	Authorise employee	200 OK, response body contains the token. Test scripts pass.	PASS

Test execution:

1. User creates a person to be used in the tests moving forward. Open Person API folder and **POST** Create Person request. Click SEND. When the post-request script is run, the personId is automatically saved under the collection variables and will be used in every other request forward.

API Endpoint	Method	Test Scenario	Expected Result	Status (Pass/Fail)
/api/v2/persons	POST	Create a person	200 OK, response body contains the personId. Test scripts pass.	PASS

2. User finds the person just created to check if they have an open account. Open Account API (Find Info) folder and **GET** Get Person Information request. Click SEND.

API Endpoint	Method	Test Scenario	Expected Result	Status (Pass/Fail)
/api/v1/persons/:personId	GET	Get person information	200 OK, response body	PASS

			contains the hasActiveAccount parameter. The parameter has the value “**false**”. Test scripts pass.	

3. User creates a personId specific account to be used in the tests moving forward. Open Account API (Create Account) folder and **POST** Create Account request. Click SEND. When the post-request script is run, the accountId is automatically saved under the collection variables and will be used in every other request forward.

API Endpoint	Method	Test Scenario	Expected Result	Status (Pass/Fail)
/api/v4/persons/:personId/accounts	**POST**	Create an account	200 OK, response body contains the accountId. Test scripts pass.	**PASS**

4. Assert that the person just created now has an open account. Open Account API (Find Info) folder and **GET** Get Person Information request. Click SEND.

API Endpoint	Method	Test Scenario	Expected Result	Status (Pass/Fail)
/api/v1/persons/:personId	**GET**	Get person information	200 OK, response body contains the hasActiveAccount parameter. The parameter has the value “**true**”. Test scripts pass.	**PASS**

5. Check that the account just created for the person has a balance of 0 (zero) since the account was opened with no funds in the body. Open Account API (Account Balance) folder and **GET** Find Account Balances request. Click SEND.

API Endpoint	Method	Test Scenario	Expected Result	Status (Pass/Fail)
/api/v1/accounts/:accountId/balances?currencyCode=EUR	**GET**	Get account balance	200 OK, response body contains the balanceId and balanceAmount. The balanceAmount has the value "**0**". Test scripts pass.	**PASS**

6. User creates the first transaction to the account. Open Account API (Transactions) folder and **POST** Create Account Transaction request. First make sure that valueDate is the date of the day when the account was created and testing is conducted. Click SEND.

API Endpoint	Method	Test Scenario	Expected Result	Status (Pass/Fail)
api/v5/accounts/:accountId/transactions	**POST**	Create a transaction	200 OK, response body contains the accounTransactiontId and correct accountId and amount. Test scripts pass.	**PASS**

7. Check once again the balance of the account. Open Account API (Account Balance) folder and **GET** Find Account Balances request. Click SEND.

API Endpoint	Method	Test Scenario	Expected Result	Status (Pass/Fail)
/api/v1/accounts/:accountId/balances?currencyCode=EUR	**GET**	Get account balance	200 OK, response body contains the balanceId and balanceAmount. The balanceAmount matches the amount value in the previous step. Test scripts pass.	**PASS**

8. Check the transaction history of the account. Open Account API (Search Transactions) folder and **GET** Find Account Transactions request. Click SEND.

API Endpoint	Method	Test Scenario	Expected Result	Status (Pass/Fail)
/api/v4/accounts/:accountId/transactions/search?dateFrom=2024-09-01&dateTo=2024-10-31&dateType=POSTING_DATE&pageNumber=1&pageSize=10	**GET**	Get account transactions	200 OK, response body contains the accounTransactiontId and correct accountId and amount that matches the Create Transaction response. Test scripts pass.	**PASS**

Negative test values:

This part tests the above-mentioned API endpoint with negative values, so values that must be specific but are not, or values that are required but missing.

Authorisation:

- Try to authorise the employee with an empty username and password in the body. Open the Authentication API folder. In the folder, click on **POST** Authorise Employee Empty Credentials. Click SEND. The test scripts pass in this case since they expect the 400 status code and error fields.

API Endpoint	Method	Test Scenario	Expected Result	Status (Pass/Fail)
/api/v1/employees/authorise	**POST**	Authorise employee	200 OK, response body contains the token. Test scripts pass.	**FAIL**

- Try to authorise the employee with the wrong password in the body. Open the Authentication API folder. In the folder, click on **POST** Authorise Employee Wrong Password. Click SEND. The test scripts pass in this case since they expect the 400 status code and error fields.

API Endpoint	Method	Test Scenario	Expected Result	Status (Pass/Fail)
/api/v1/employees/authorise	**POST**	Authorise employee	200 OK, response body contains the token. Test scripts pass.	**FAIL**

Create person:

- Try to create a person with **invalid** personTypeCode. The only accepted values are P and L. Insert X. Open the Person API folder. In the folder, click on **POST** Create Person Invalid personTypeCode. Click SEND. The test scripts pass in this case since they expect the 400 status code and error fields.

API Endpoint	Method	Test Scenario	Expected Result	Status (Pass/Fail)
/api/v2/persons	**POST**	Create a person	200 OK, response body contains the token. Test scripts pass.	**FAIL**

- Try to create a person with an **empty** personTypeCode. The only accepted values are P and L. Leave it empty. Open the Person API folder. In the folder, click on **POST** Create Person Empty personTypeCode. Click SEND. The test scripts pass in this case since they expect the 500 status code and error fields.

API Endpoint	Method	Test Scenario	Expected Result	Status (Pass/Fail)
/api/v2/persons	**POST**	Create a person	200 OK, response body contains the token. Test scripts pass.	**FAIL**

- Try to create a person with no authentication token. The token is removed from the header. Open the Person API folder. In the folder, click on **POST** Create Person No Authentication Token. Click SEND. The test scripts pass in this case since they expect the 401 status code.

API Endpoint	Method	Test Scenario	Expected Result	Status (Pass/Fail)
/api/v2/persons	POST	Create a person	200 OK, response body contains the token. Test scripts pass.	FAIL

Create an account:

- Try to create an account without accountTypeCode. Open Account API (Create Account) folder and POST Create Account Without accountTypeCode request. Click SEND. The test scripts pass in this case since they expect the 400 status code and error fields.

API Endpoint	Method	Test Scenario	Expected Result	Status (Pass/Fail)
/api/v4/persons/:personId/accounts	POST	Create an account	200 OK, response body contains the accountId. Test scripts pass.	FAIL

- Try to create an account without accountNumber value. Open Account API (Create Account) folder and POST Create Account Without Account Number request. Click SEND. The test scripts pass in this case since they expect the 400 status code and error fields.

API Endpoint	Method	Test Scenario	Expected Result	Status (Pass/Fail)

/api/v4/persons/:personId/accounts	POST	Create an account	200 OK, response body contains the accountId. Test scripts pass.	FAIL

Create transaction:

- Try to create a transaction to the account when the valueDate is in the past and before the account was created. Open Account API (Transactions) folder and POST Create Account Transaction Value Date Before Account Activation Date request. First make sure that valueDate is set in the past before the account activation date. Click SEND. The test scripts pass in this case since they expect the 400 status code and error fields.

API Endpoint	Method	Test Scenario	Expected Result	Status (Pass/Fail)
api/v5/accounts/:accountId/transactions	POST	Create a transaction	200 OK, response body contains the accounTransactiontId and correct accountId and amount. Test scripts pass.	FAIL

Conclusion

The testing of Tuum Sandbox API was successfully executed according to the test plan. API endpoints, including person creation, account creation, and transaction processing, were validated. The response codes functioned as expected in both positive and negative test scenarios. Security measures related to authorization were effectively tested. Overall, the

API performed reliably. The negative test cases also produced the correct error responses. Therefore, Tuum API meets the requirements.