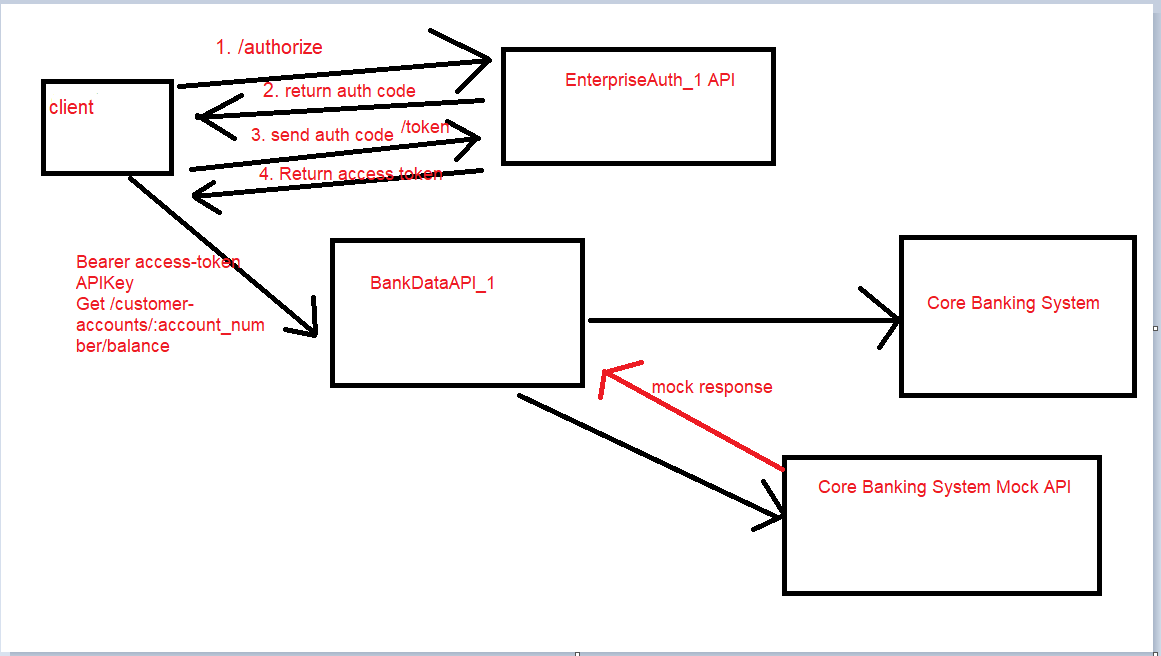
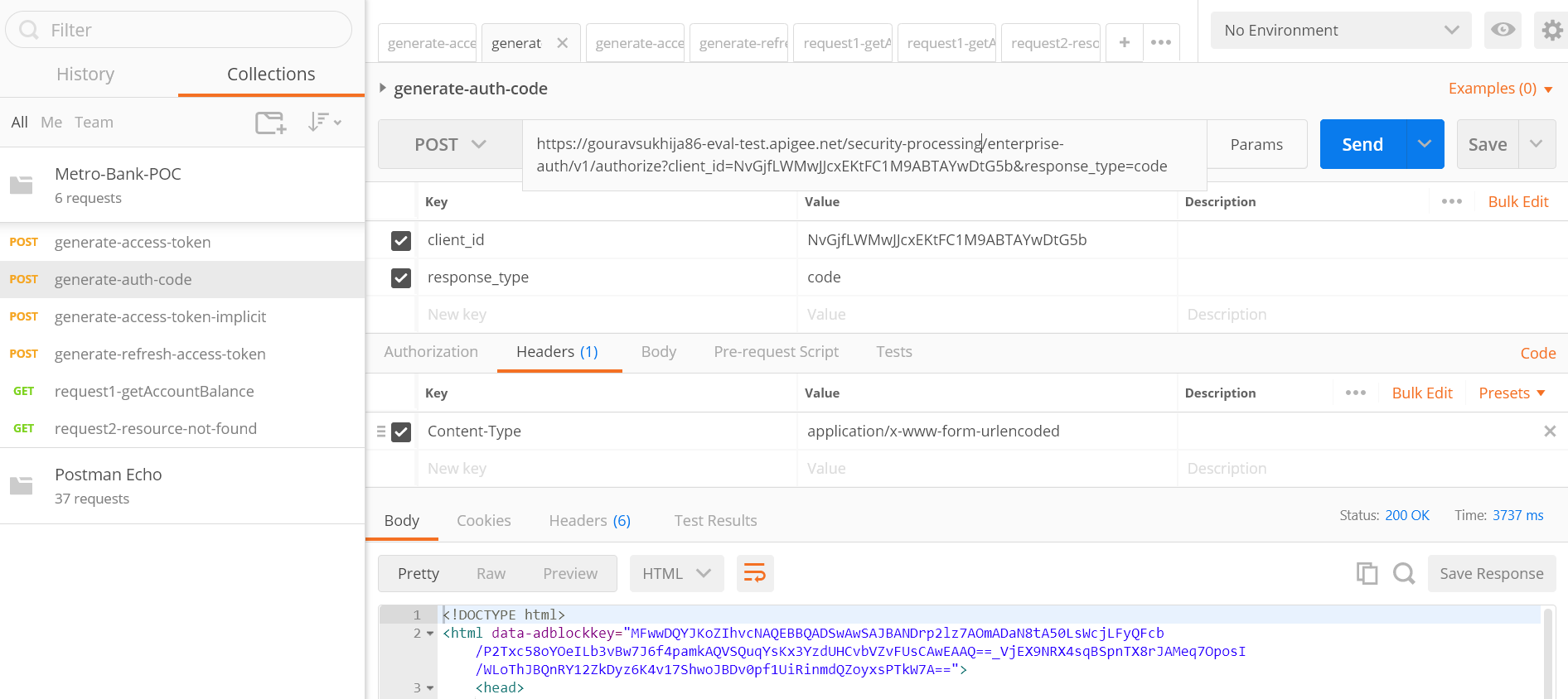
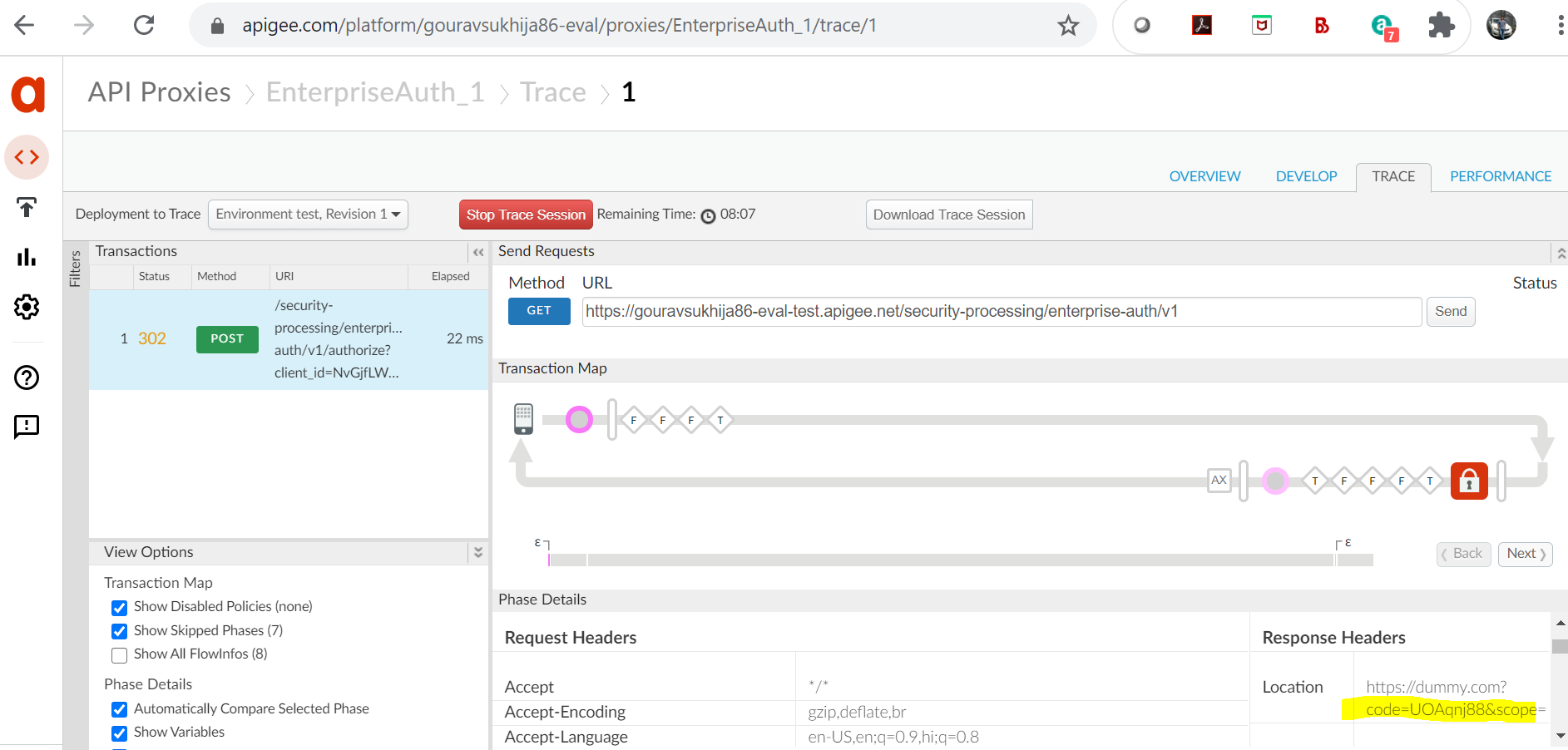
# High Level Design:

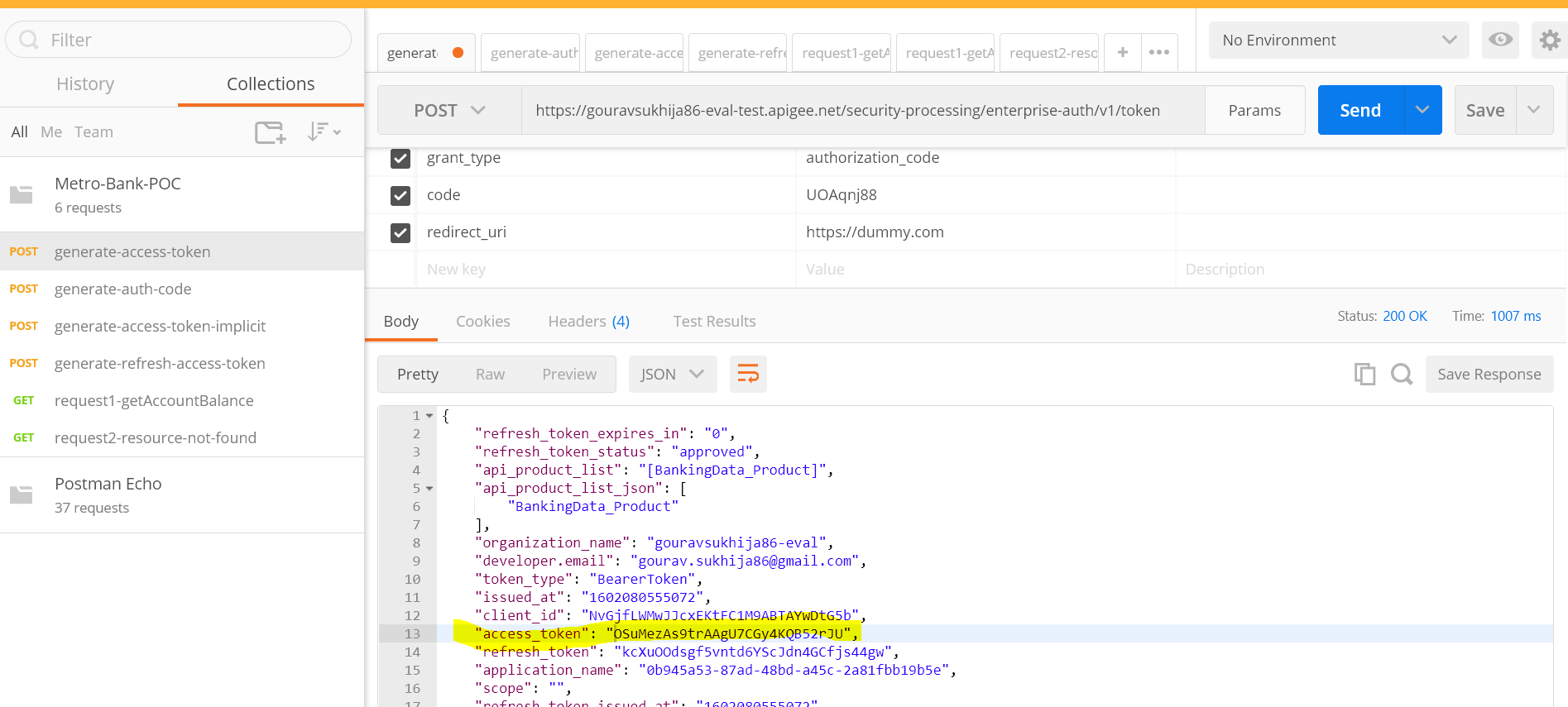


# End to End proof of implementation with screen shots:

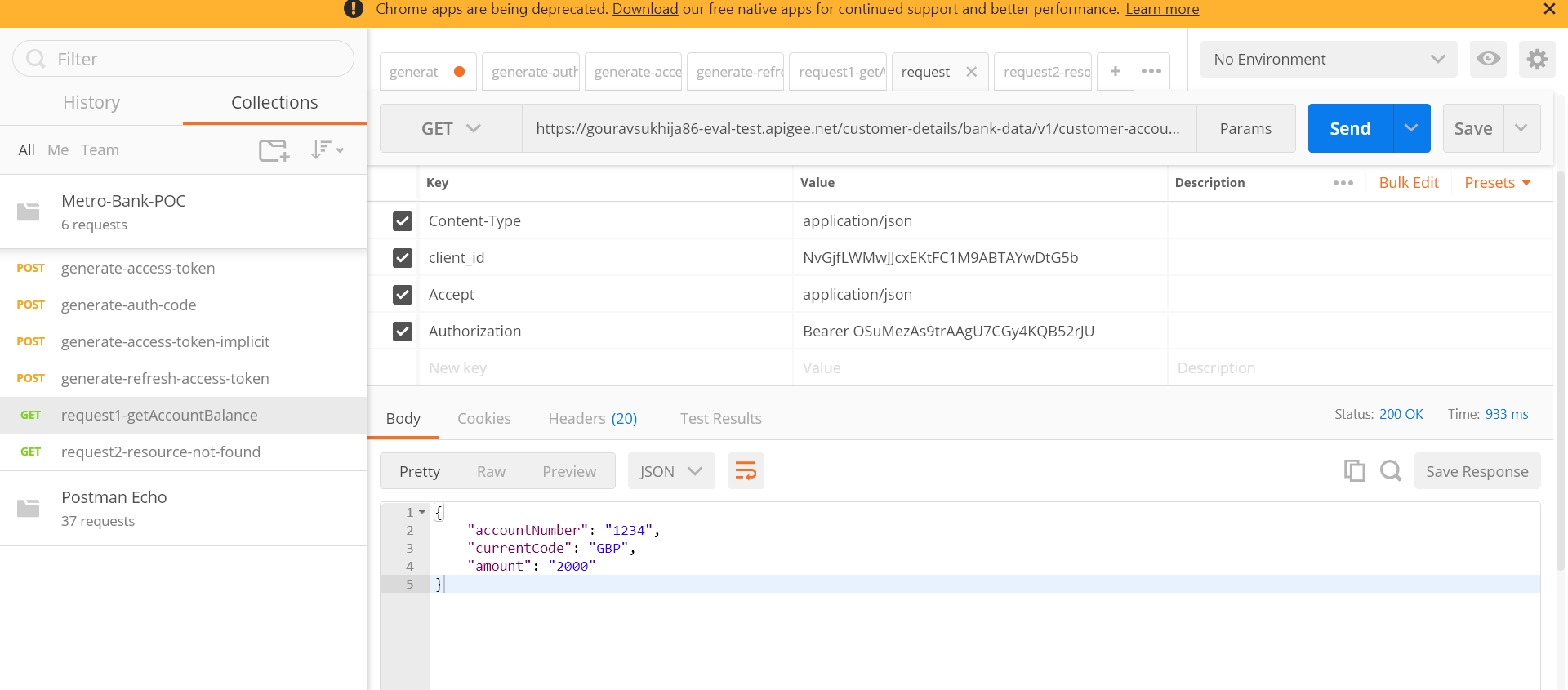
1. Generate Auth code: This call returns 302 response

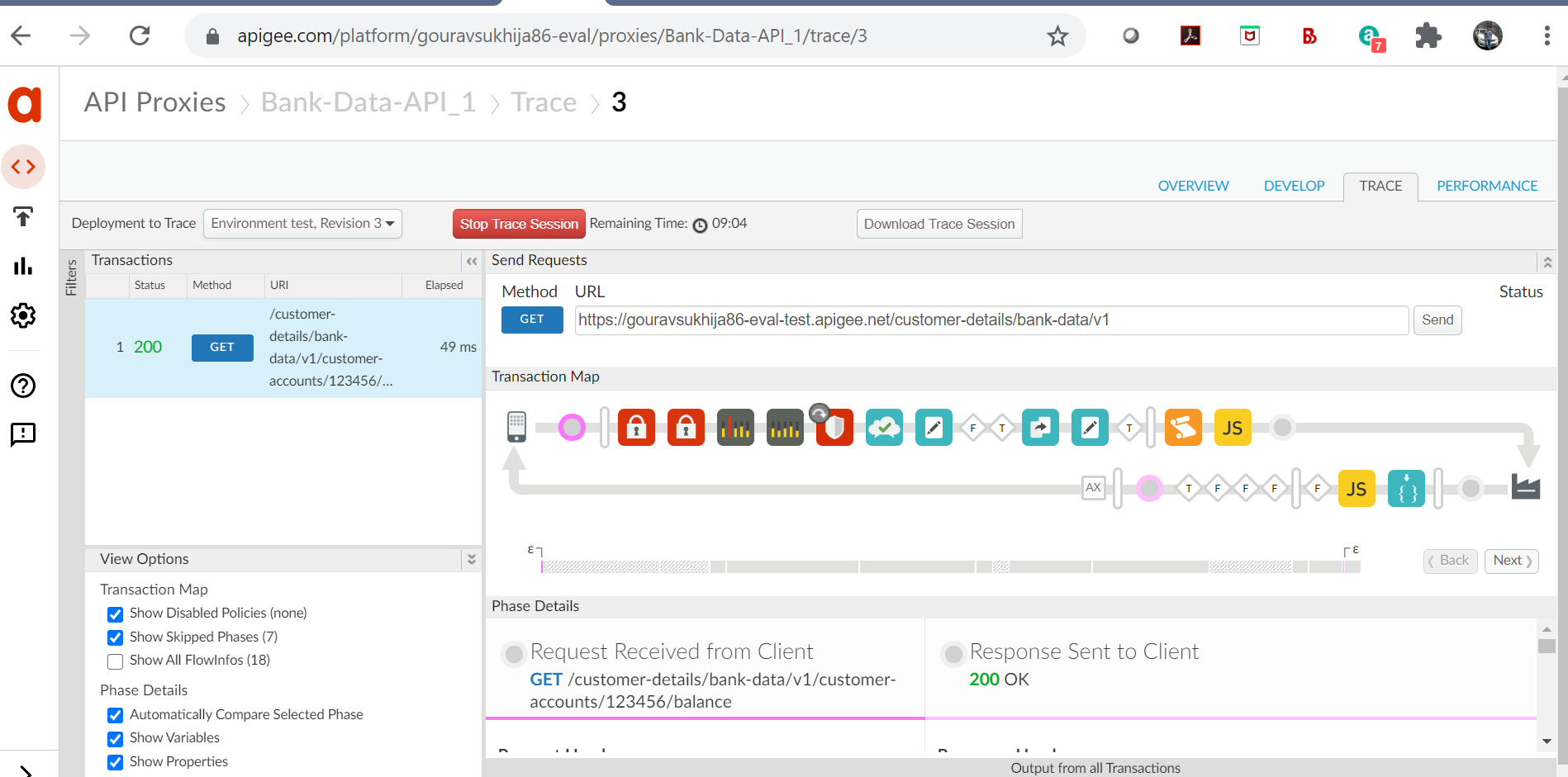




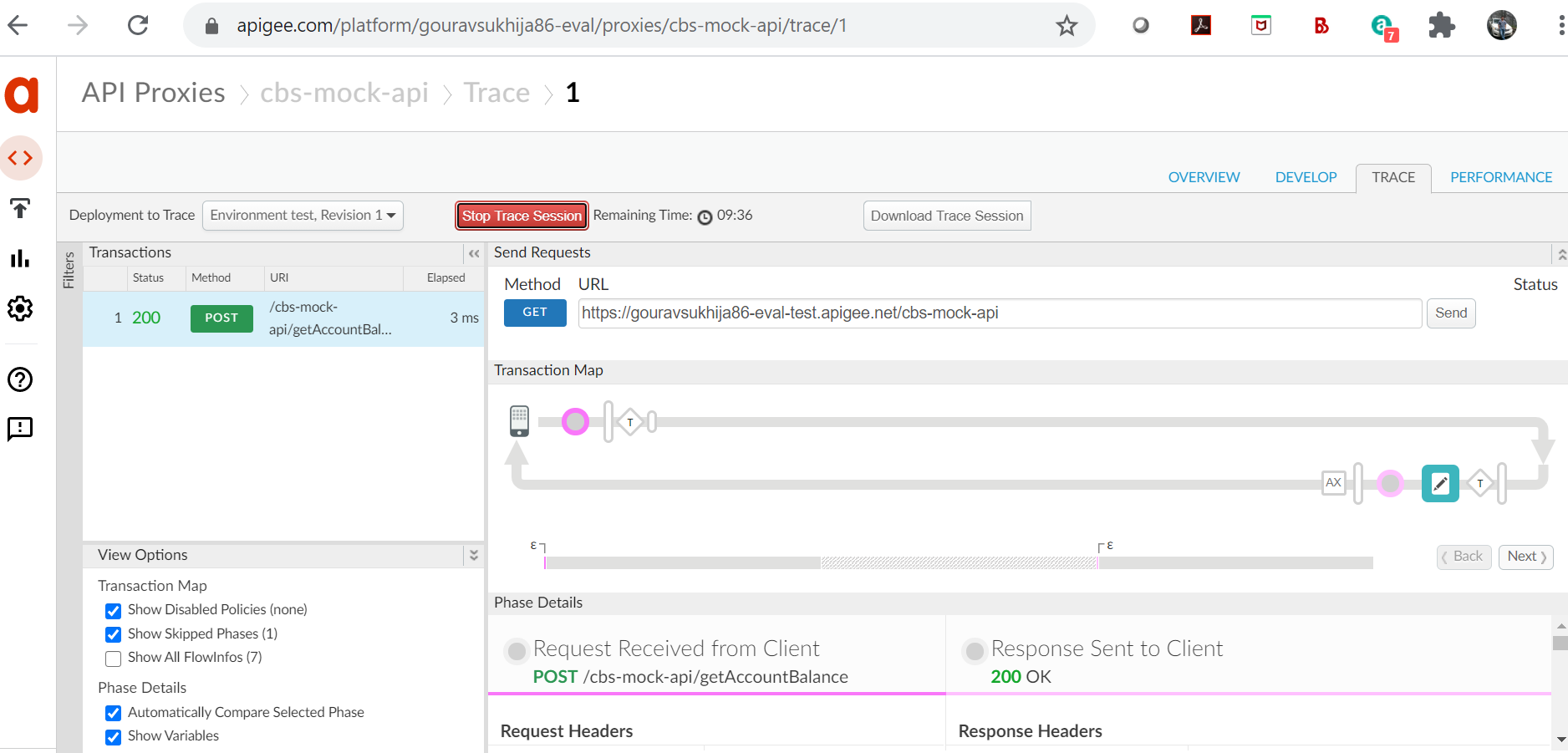


1. Grab the access the token and call Bank data API





Mock Response:



## Details Of Deliverables

1. Bank-Data-API\_1 🡪 This gets the details from core banking system



1. EnterpriseAuth\_1 🡪 This API is used to obtain the access token. It supports implicit grant type and authorization code grant\_type



1. cbs-mock-api 🡪 This api is used to mock the Soap response of core banking system



1. OAS document for BankDataAPI\_1



1. PostMan collection





## Details of Requirement and its implementation details.

Please find the comments inline

* Prerequisite: The person needs to have a free apigee edge account created.

Had the free edge account

* Overall requirement: There is a need to expose the existing APIs of a Bank in a secured way. The APIs would be consumed by third party apps, which would be registered through developer portal.

Please check the high level design for the same

* Common Feature in APIs:
  + They MUST be protected against the JSON, XML threat
* Implemented it(Json threat protection policy in BankDataAPI\_1)
  + SHOULD only handle 100 per second with an option to change it in future
* Implemented it (spike arrest protection policy in BankDataAPI\_1. Can be made configurable with KVM. But have not implemented the KVM for it)
  + MUST support a standard format of error message representation
* Implemented it
  + MUST support standard HTTP error 4XX and 5XX errors
* Implemented it
  + MUST comply with data masking
* Implemented it. Included the data-mask-configuration xml file. It can be applied using management API
  + Best practices of API development MUST be taken care
* Implemented the options call, fault handling and cors headers
  + There SHOULD be a logging functionality provided. It’s up to the developer to decide which all meaningful parameters should go to log file.
* This is only supported by private edge cloud installation
  + MUST ensure a quota counter defined per developer app.
* Implemented it using quota policy for dev-app
  + All the communication between the API consumer and API should be in JSON format.
* Implemented it
  + All the confidential and user specific data needs to be masked.
* Implemented it

* Authentication API:
  + The authenticating API should support ALL the Oauth v2 features (auth code generation, token generation with client credentials, auth code and refresh token grant types)
* Implemented the auth code generation, implicit grant type and refresh grant types
  + The code and token should be valid for a limited duration, which is configurable.
* Default expiry time is considered
  + While generating the token and auth code, the API should add a custom attribute to it.
* Custom attribute(mobile number) was added in the dev-app. Response of Access-Entity policy attached below

<?xml version="1.0" encoding="UTF-8" standalone="yes"?><App name="BankingData\_DevApp"><ApiProducts/><Credentials><Credential><Attributes/><ConsumerKey>NvGjfLWMwJJcxEKtFC1M9ABTAYwDtG5b</ConsumerKey><ConsumerSecret>eAA349i9XUnGATTr</ConsumerSecret><ExpiresAt>-1</ExpiresAt><IssuedAt>1601988174926</IssuedAt><ApiProducts><ApiProduct><Name>BankingData\_Product</Name><Status>approved</Status></ApiProduct></ApiProducts><Scopes/><Status>approved</Status></Credential></Credentials><AppFamily>default</AppFamily><AppId>0b945a53-87ad-48bd-a45c-2a81fbb19b5e</AppId><Attributes><Attribute><Name>DisplayName</Name><Value>BankingData\_DevApp</Value></Attribute><Attribute><Name>Notes</Name><Value></Value></Attribute><Attribute><Name>mobile-number</Name><Value>+918396965023</Value></Attribute></Attributes><CallbackUrl>https://dummy.com</CallbackUrl><CreatedAt>1601988174344</CreatedAt><CreatedBy>gourav.sukhija86@gmail.com</CreatedBy><DeveloperId>adfbfcd4-3841-4031-a797-582c0d4c0da7</DeveloperId><LastModifiedAt>1602052919978</LastModifiedAt><LastModifiedBy>gourav.sukhija86@gmail.com</LastModifiedBy><Scopes/><Status>approved</Status></App>

AccessEntity.ChildNodes.Access-Entity.App <?xml version="1.0" encoding="UTF-8"?><App name="BankingData\_DevApp"><ApiProducts/><Credentials><Credential><Attributes/><ConsumerKey>NvGjfLWMwJJcxEKtFC1M9ABTAYwDtG5b</ConsumerKey><ConsumerSecret>eAA349i9XUnGATTr</ConsumerSecret><ExpiresAt>-1</ExpiresAt><IssuedAt>1601988174926</IssuedAt><ApiProducts><ApiProduct><Name>BankingData\_Product</Name><Status>approved</Status></ApiProduct></ApiProducts><Scopes/><Status>approved</Status></Credential></Credentials><AppFamily>default</AppFamily><AppId>0b945a53-87ad-48bd-a45c-2a81fbb19b5e</AppId><Attributes><Attribute><Name>DisplayName</Name><Value>BankingData\_DevApp</Value></Attribute><Attribute><Name>Notes</Name><Value/></Attribute><Attribute><Name>mobile-number</Name><Value>+918396965023</Value></Attribute></Attributes><CallbackUrl>https://dummy.com</CallbackUrl><CreatedAt>1601988174344</CreatedAt><CreatedBy>gourav.sukhija86@gmail.com</CreatedBy><DeveloperId>adfbfcd4-3841-4031-a797-582c0d4c0da7</DeveloperId><LastModifiedAt>1602052919978</LastModifiedAt><LastModifiedBy>gourav.sukhija86@gmail.com</LastModifiedBy><Scopes/><Status>approved</Status></App>

* + There should be an option in the Authentication API to revoke the tokens which are associated with an app and end user.
* Not implemented
  + Consumer should be able to select the response code (200 vs 302) for authorization code generation.
* Not implemented
  + The access token should have custom attributes phone number picked up from the developer app’s custom attribute. The same parameter should be available at custom report.
* Not implemented

* Banking Data API:
  + There should be resources like users, accounts, balance2s, payments and transactions
* These resources are not yet implemented
  + The API consumer should be able to retrieve the balance, payment and transaction details for a specific account and all the balances, payments, transactions for a specific user, who has more than one account.
* Implemented one resource to get the account balance for a particular account number
  + The API consumer should be able to retrieve the details of the user.
* Not implemented yet
  + The API consumer should be able update the contact details of the user.
* Not implemented yet
  + Each of the resources needs to be accessed through access token (authorization code grant type) and the with a specific oauth scope.
* Implemented it
  + The API needs to communicate with Core Banking System, which is a SOAP based service.
* Implemented the SOAP based mock service
  + The email id associated with the access token should be available at custom report.
* Not implemented yet

* Mock API:
  + In the absence of Core Banking System, please implement a suitable mocking functionality on Apigee.
* Implemented it
* Please implement API Product and Apps involving the APIs, based on popular use cases in the industry.
* Yes Implemented it
* Please create basic BDD test cases
* Yes attached the postman collection
* A custom report should be created which would show the usage of API resources over a specific duration by a specific email id.
* Not Implemented yet
* Please create a document containing the following:
  + API Design
* Yes, attached it

# App Onboarding thoughts

* + 1. The API can be published in the dev-app after going through below review.
  + Data Review
  + Security Review
  + Product Owner Review
    1. Once the api-proxy is published. Third parties can consume it.
    2. Third party can be on-boarded via dev-protal and can locate the API proxies.
    3. External users can go under an approval flow from dev-portal.