Ministry of Housing and Urban Affairs
Government of India

National Urban Digital Mission

Building cities that work for people

Payment Gateway Integration

Payment Gateway Integration

NIUA Payment Gateway service acts as a liaison between NIUA apps and external payment gateways facilitating payments, reconciliation of payments and lookup of transactions' status'.

Extensions

- Axis, Phonepe,PayU and Paytm payment gateways are implemented.
- Additional gateways can be added by implementing the Gateway interface. No changes required to the core packages.

Integration Benefits

- Record of every transaction against a bill.
- Record of payment for multiple bill details for a single consumer code at once.

Configuration Details

Steps to fill data

The payment gateway is a vendor oriented service that is integrated with different modules in order to facilitate the transactions. Below mentioned are the steps which are followed:

1. The client has to finalize a payment gateway vendor (for example PAYU, Paytm, HDFC, AXIS atc.) depending upon the requirements.
2. Need to understanding of all the headers as required by payment gateway integration document.
3. In case of any doubt, please reach out to the person who has shared this template with you to discuss and clear your doubts. After which the details/ documents mentioned in the template would be provided by the vendor.
4. These details are to be received separately for both production as well as UAT.
5. Get the IP address for UAT and Production environments **whitelisted** from the vendor.
6. Verify the data once again by going through the checklist and making sure that each and every point mentioned in the checklist is covered.

Payment Gateway Integration

Following are the details required from the new payment gateway vendor in order to configure the payment gateway:

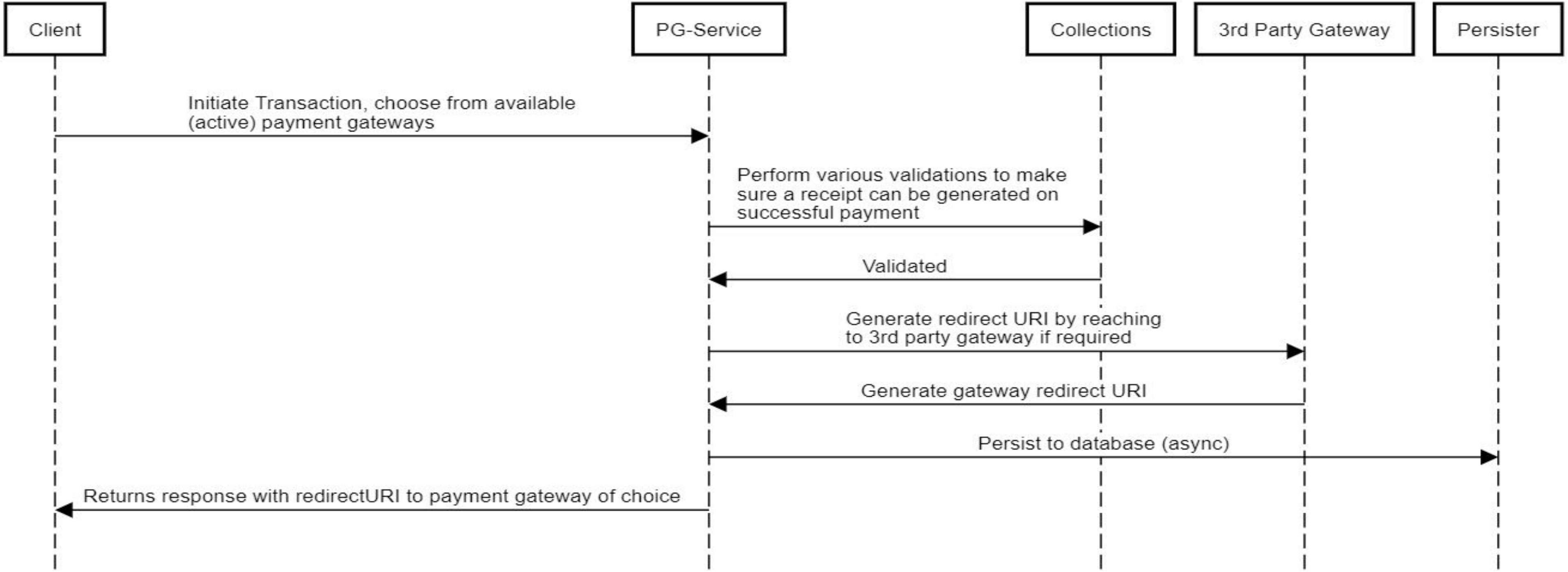
- Integration Kit provided by vendor for UAT and Production environment setup.
- Reconciliation logic has to be implement for new integration
- While finalizing a payment gateway vendor make sure that the vendor should support transactions into multiple bank accounts based on the key(which would be tenantid)

Reconciliation

- Reconciliation is carried out by two jobs scheduled via a Quartz clustered scheduler.
- Early Reconciliation job is set to run every 15 minutes [configurable via app properties] and is aimed at reconciling transactions that were created 15 - 30 minutes ago and are in PENDING state.
- Daily Reconciliation job is set to run once per day and is aimed at reconciling all transactions that are in PENDING state, except for ones that were created 30 minutes ago.

Payment Flow

PG-Service - Initiate Transaction



Client

PG-Service

Collections

3rd Party Gateway

Persister

Initiate Transaction, choose from available (active) payment gateways

Perform various validations to make sure a receipt can be generated on successful payment

Validated

Generate redirect URI by reaching to 3rd party gateway if required

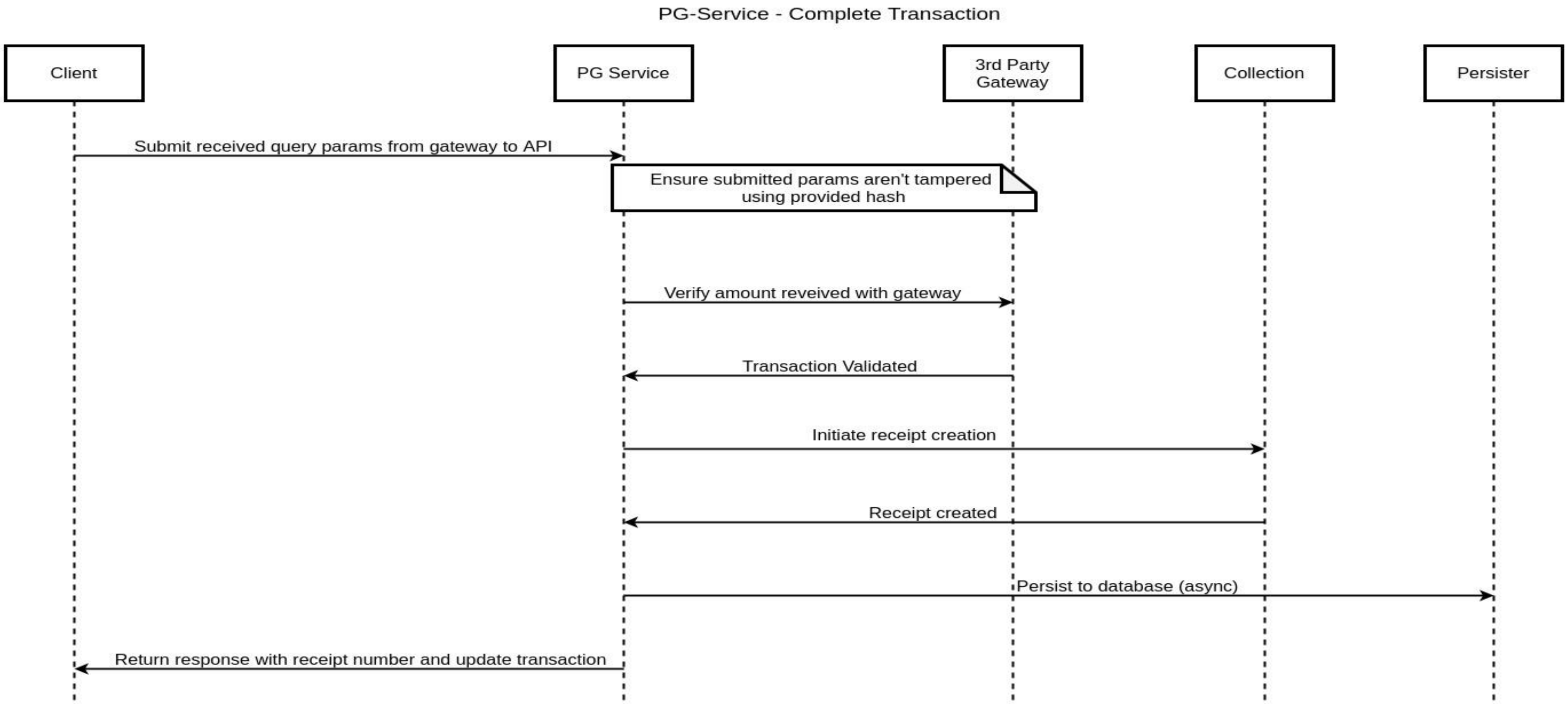
Generate gateway redirect URI

Persist to database (async)

Returns response with redirectURI to payment gateway of choice

Ministry of Housing and Urban Affairs
Government of India

Payment Flow


PG-Service - Complete Transaction
Client
PG Service
3rd Party Gateway
Collection
Persister
Submit received query params from gateway to API
Ensure submitted params aren't tampered using provided hash
Verify amount reveived with gateway
Transaction Validated
Initiate receipt creation
Receipt created
Persist to database (async)
Return response with receipt number and update transaction

Configuration Details

Project Structure

Packages

- config - All configuration related to the App, including main quartz scheduler configs
- service - Consists of main service classes for app functioning.
- service/gateways/{gatewayName} - Third-party payment gateways, each sub-package to encapsulate the entire code needed for the gateway.
- service/jobs/* - Contains jobs and respective configs for the jobs which are to be scheduled on the Quartz scheduler
- web/controllers - Controllers for the app.

Dependencies

- Postgres database to store transaction data and enable quartz clustered scheduling.
- Collection Service to validate request and to create receipts.
- ID Gen Module to generate unique transaction ID's.
- Persister module for persistence.
- Merchant specific properties, such as merchant id and secret needs to be configured.

Integration Details

Steps to Integration

1. To integrate, host of pg-service should be overwritten in helm chart
2. /pg-service/transaction/v1/_create should be added in the module to initiates a new payment transaction, on successful validation
3. /pg-service/transaction/v1/_update should be added as the update endpoint to updates an existing payment transaction. This endpoint is issued only by payment gateways to update the status of payments. It verifies the authenticity of the request with the payment gateway and forward all query params received from a payment gateway
4. /pg-service/transaction/v1/_search should be added as the search endpoint for retrieving the current status of a payment in our system.

Thank You

“UPYOGal Transformation is more about humans than UPYOGal”