#18/2A, GRS Towers, Second Floor, Above Spencer's
Sarjapura Road, Bangalore – 560 102, Karnataka, India

Strictly Confidential

Payment Gateway Integration Document

INTRODUCTION

Payment Gateway Interface is part of eGov Collection system. It provides a interface for citizen to facilitate online payment for the different services (Propert Tax, Water Charges, Swerage Charges, Trade License, Building Plan Approval etc.) provided by the ULB.

SCOPE

Single interface to make online payments for the different services provided by ULB and to reconcile the transactions in pending state.

SYSTEM DESIGN

- All requests for the online payment directed by the citizen go through this module.
- It provides a unified interface to the external Payment Gateways like Bill Desk, ICICI, AXIS, INDUSIND, ATOM etc.
- The system tracks success/failures of all the requests that go through.
- Collection system creates the requests as per the specifications provided by Payment Gateway.
- Upon completion of transaction at the payment gateway, response should be sent back to eGov Collection system. For both successful/failure transactions:

- Payment Gateway should send a 302 Redirect response back to the browser. The redirect URL should be the RU (Return URL) which was sent by ERP in the initial request (i.e., http://dev4.governation.com/collection/citizen/onlineReceipt!acceptMessageFromPaymentGateway.action).

For successful transactions, eGov Collection system call back will then return an acknowledgement screen as its response, where citizen can click on a link and generate the receipt.

For failed transactions, a message along with the ERROR CODE and ERROR MESSAGE(reason of failure) is persisted in eGov Collection system.

- Transaction with status as PENDING for reconciliation with Payment Gateway, can be reconciled through a scheduler provided by eGov Collection system (OR)
- eGov Collection system also provides a user interface to manually update the PENDING

transactions as SUCCESS/FAILURE/TO BE REFUNDED, based on the status report provided by the Payment Gateway.

- Reports (transactions list) - date wise, payment gateway wise and status wise provided by the system.

Integration Process

Payment gateway that is to be integrated must be defined in the masters configuration. This facilitates system administrator to display payment gateway in the UX, to define the payment gateway service URL, callback URL and to enable/disable the payment gateway as and when required. Masters configuration includes defining service category and service details.

For example, if system needs to be integrated with "Federal Payment Gateway", following activites need to be performed:

1. Define Service Category : This master defines category for payment gateway and enables the system administrator to enable/disable the payment gateway.

Insert into egcl_servicecategory (id, name, code, isactive, version, createdby, createddate, lastmodifiedby, lastmodifieddate) values (nextval('seq_egcl_servicecategory'), 'Federal Payment Gateway', 'FPG', true, 0, (select id from eg_user where username='egovernments'), now(), (select id from eg_user where username='egovernments'), now());

name : Name of the payment Gateway
code : Code of the payment gateway, unique arbitratry code can be allocated.
Isactive : To enable/disable the payment gateway from appearing in the UX.

2. Define Service Details : This master defines the service URL of the payment gateway and callback URL to be sent in each payment request.

Insert into egcl_servicedetails (id, name, serviceurl, isenabled, callbackurl, servicetype, code, fund, fundsource, functionary, vouchercreation, scheme, subscheme, servicecategory, isvoucherapproved, vouchercutoffdate, created_by, created_date, modified_by, modified_date, ordernumber) values (nextval('seq_egcl_servicedetails'), 'Federal Payment Gateway', 'https://federalpayment.com/payrequest',

true, 'https://xyx.com/collection/citizen/onlineReceipt!acceptMessageFromPaymentGateway.action', 'P', 'FPG', (select id from fund where code='01'), null, null, false, null, null, (select id from egcl_servicecategory where code='FPG'), false, now(), 1, now(), 1, now(), null);

serviceurl – URL provided by payment gateway provider to invoke the payment gateway UI.
Callbackurl – Callback URL to be send in each payment request to the payment gateway. Replace "xyz.com" in the sample script with the actual domain name where the ERP is hosted, remaining URL should remain intact.

3. Prepare Payment Request and Parse Payment Reponse :

Write an adaptor based on the specifications provided by payment gateway as each payment gateway has different approach. Development team can achieve this by implementing PaymentGatewayAdaptor interface provided by eGov ERP.

PaymentGatewayAdaptor provides 2 APIs that needs to be overridden to prepare the payment request and to parse the payment response as per the specifications provieded by payment gateway.

```
public interface PaymentGatewayAdaptor {
 public PaymentRequest createPaymentRequest(ServiceDetails paymentGatewayService, ReceiptHeader receipt);

public PaymentResponse parsePaymentResponse(String response);

}
```

Refer Javadoc or PaymentGatewayAdaptor, PaymentRequest, PaymentResponse for detailed understanding.

4. Register Payment Gateway Adaptor :

Adaptor created above (point no.3) must be registered in order to enable system to fetch the adaptor on the fly. Adaptor bean must be defined in the configuration file "*applicationContext-collection-services.xml*".

Bean Id must follow the naming convention:
<ServiceDetailsCode>*PaymentGatewayAdaptor*

Thus the bean for the example payment gateway must be defined as :

```
<bean id="FPGPaymentGatewayAdaptor" class="org.egov.collection.FederalAdaptor" />
```

5. Offline Reconciliation :

System provides facility to reconcile pending transactions as well. Pending transactions are the ones that are not reconciled with the payment gateway due to various reasons like response from payment gateway lost due to internet connectivity issue. To reconcile such payments, implementation team can create a job and service and register the schedular with preffered time in the class CollectionSchedulerConfiguration.