DubaiPay Development and Integration Guide

Development Guide

DSG_eServices_DubaiPay _00001

Version Number 5.0

Document Control

Document History

Date	Version	Author(s)	Description
29/01/2014	1.0	Waqas Ur Rehman	First draft
12/03/2014	2.0	Waqas Ur Rehman	Added ePay5 Validations
22/03/2014	3.0	Waqas Ur Rehman	Update Security Mechanism
01/04/2014	4.0	Waqas Ur Rehman	Added SSL Certificate
01/01/2019	5.0	Manar Al Maazmi	Updated the layout

Distribution List

Name	Title	Entity
Mira Sultan Obaid	Director, SSED	SDG
Waqas Rehman	Manager, ePay Section	SDG

Approval List

Date	Name	Title	Signature
	Mira Sultan	Director, SSED	
	Waqas Rehman	Manager, ePay Section	

Table of Contents

Glossary

Term	Description
DSG	Dubai Smart Government
DOF	Department of Finance
DubaiPay	Electronic Payment Gateway of Dubai Smart Government
IT	Information Technology
URL	Uniform Resource Locator
SP	Service Provider, Government departments accepting credit card payments for the services they provide
JSP	Java Server Pages
MD5	Message-Digest algorithm 5
PG	Payment Gateway
XML	Extensible Markup Language
3DS	3-Domain Secure Payment
MiGS	MasterCard Internet Gateway Service
API	Application Programming Interface
PCI-DSS	Payment Card Industry Data Security Standard
AMEX	American Express
J2EE	Java 2 Platform Enterprise Edition
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol over Secure Socket Layer

Secure Sockets Layer (SSL)	It’s a protocol designed for secure transfer of Information over the Internet. Information sent through an SSL-secured form is encrypted so that the information is not tempered with while the transfer is taking place.
Authorization	It is a process to charge the customer credit card. It will block the amount and 6 digits authorization code will be generated.
Capture	It is a process to complete the payment. It will mark the transaction amount from customer credit card to be moved to merchant bank account.
Settlement	It run once at the cutoff time agreed with the payment Processor. Settlement process moves the money for all captured transactions to the merchant.
Card Issuer	The bank or company which issues a payment card to the customer
Card Verification Code (CVV)	Credit Card verification code is 3 or 4digit number, which found on the back of the card. Many credit cards have a CVN printed, not embossed, on the card. Each card association has its own name for this value: Visa calls it the Card Verification Value (CVV2), MasterCard calls it the Card Validation Code (CVC2), and American Express and Discover call it the Card Identification Digits (CID
Cardholder	A person to whom a payment card has been issued.
Card Not Present Scenario	A transaction where the merchant does not have physical access to the card (e.g. through telephone, mail order or Internet transactions). All transactions where a credit card is not physically swiped through a terminal, including internet transactions, phone transactions, or credit-card numbers keyed into a terminal/virtual terminal, fall into this category.
Card Present Scenario	A transaction where the card is presented physically to the merchant. Examples are POS transactions, online transactions where Secure Code is presented etc.
Moto	A MOTO transaction, also known as a Card Not Present transaction is a transaction for which the credit card is not physically swiped through a terminal. This type of transaction includes telephone, mail order, and internet. Unlike a card-present transaction, in which the Issuing Bank is liable, for a MOTO transaction, the Acquiring Bank is liable

Payment Gateway	A payment gateway facilitates the secure transfer of transactions from a merchant to a third-party payment processor, associated with the merchant's acquiring bank
Service Provider	An entity interfacing with DubaiPay to provide payment facility to their customers

1. Chapter One: This Development Guide

1.1 About This Guide

The DubaiPay Development & Integration Guide is a guide to present and explain the necessary development activities required by Service Providers to integrate their Online Services and applications with DSG’s ePayment Gateway to provide Online Payments facility to their users.

Here we cover everything you need to know to make the best use of this guide.

1.2 Who should read this Guide

This guide is intended for Developers, Software Specialist and Software Application vendors who are looking to integrate with Dubai Smart Government ePayment Gateway.

It is also intended for Development Managers, IT Managers and IT Consultants to help them understand the Centralized ePayment system provided by DSG, and how it can be integrated with their Department's Online Services to provide online payments facility to the users.

1.3 Pre-Requisites

The following are the pre-requisites for Service Provider’s to integrate with DSG’s DubaiPay application.

Before proceeding with this development guide you should have a good knowledge of Java or .Net programming language.

You should be familiar with Web Applications, a basic understating to the HTTP "Hyper Text Transfer Protocol", WebServices Technologies, SOAP, WSDL, WS-* standards and hashing algorithm is also essential.

1.4 How this Guide is organized

1.1.1 Chapter One: This Development Guide

This chapter explains this development guide, how it should be used and whom it is addressed to. This chapter shows how this Development Guide is divided and what you should expect once finished reading it.

1.1.2 Chapter Two: DubaiPay Service Overview

This Chapter gives a brief overview on the ePayment Gateway Initiative launched by Dubai Smart Government. It also describes various payment options service provider can provide to the users.

1.1.3 Chapter Three: Integration Approaches

This Chapter explains the integration approaches service provider can use integrate with DubaiPay.

1.1.4 Chapter Four: Online Integration

This Chapter explains the online integration accompanied with Development Guide, in the first section it shows how the applications works from functional perspective. The second part of this Chapter details the integration.

1.1.5 Chapter Five: Web Service Integration

This Chapter explains the Web Service integration accompanied with Development Guide, in the first section it shows how the applications works from functional perspective. The second part of this Chapter details the integration.

1.1.6 Chapter Six: Authorize Integration

This Chapter explains the Authorize integration accompanied with Development Guide, in the first section it shows how the applications works from functional perspective. The second part of this Chapter details the integration.

1.1.7 Chapter Seven: Transaction Inquiry

This Chapter explains the transaction inquiry accompanied with Development Guide, in the first section it shows how the inquiry works from functional perspective. The second part of this Chapter details the integration.

1.1.8 Chapter Eight: Reconciliation

This Chapter explains the reconciliation process accompanied with Development Guide for automatic and manual reconciliation.

1.1.9 Chapter Nine: Sample

This section has the sample codes for the service providers.

1.1.10 Further Information

In Case you need further assistance related to this development guide you can always contact us at PaymentSupport@dsg.gov.ae

1.1.11 Feedback

To send comments, errors, suggestions, and questions about this Guide to the DSG team, please send the feedback at PaymentSupport@dsg.gov.ae

2. Chapter Two: DubaiPay Service Overview

Dubai Smart Government (DSG) offers a payment solution (DubaiPay) to government and non-government “Service Providers”, allowing them to offer their customers online payment for their services such as water and electricity while abstracting the e-payment process for them.

DubaiPay is a centralized, integrated and secure payment gateway. It enables service providers to provide customers with On-Line payment capabilities (DubaiPay) through different payment methods such as credit cards (Visa, Master, JCB and Amex), direct debit banks (CBD, ADCB, ADIB, UNB, and DIB) and eDirham cards.

3. Chapter Three: Integration Approaches

DubaiPay offers many features and options that can be tailored to specific business needs of the Service Providers. They can easily connect to the DubaiPay, which provides the complex infrastructure and security necessary to ensure fast, reliable and secure transmission of transaction data.

Service provider can provide payments options from the following channels:

1. Online Payment from their Portals
2. Mobile Payment
3. POS (Point of Sale)
4. Telephone

DubaiPay provides following integration approaches of making electronic payments, depending on service providers business:

1. Online Integration Profile.
2. Web Service API integration Profile.
3. Authorization integration Profile

3.1 Online Integration Profile

Online integration profile is a more secure way considered for online payments where Service Provider will redirect the user to DubaiPay. User will select the payment options (Credit Card, Direct Debit, eDirham G2). After selecting the payment option, user will be redirect to the selected payment processors. User provides his credit card related information to credit card payment processor Gateway directly and is redirected to back to the Service Provider website after verification and other checks by Payment Gateway.

This model is secure as credit card information is not transmitted by the Service Provider; according to PCI standards, any non PCI compliant party cannot transmit process and store sensitive credit card information.

3.2 Web Service API Integration Profile

Web Service API profile approach refers to the traditional way for making online payments, Service Provider collects all the credit card related information from the user and sends it to the Payment Gateway for processing. This model is also adapted for payments where redirection is not an option like Smart Phones, POS terminals, MOTO transactions and other devices.

This approach requires transmission of sensitive credit card option and service provider is subjected to follow PCI standards.

3.3 Authorization integration Profile

In some cases, service provider has to block the amount only and after manual verification provide service to the customer. DubaiPay allows service providers to block the amount using online integration profile.

This approach is also secure as credit card related information are provided to the Payment Processor and service providers are not transmitting sensitive credit card data and using Web Service API integration profile to capture / reverse the transaction.

4. Chapter Four: Online Integration Profile

In this Chapter we will explain how the payment flow works based on online integration profile, it is essential to understand the workflow of the payment before we can explain the integration details in the second part of this Chapter.

4.1 How it works?

In Online integration, user will visit Service Providers website, login and selects the service to make the payment. This approach is more secure as credit card related information is provided to the Payment Processor.

A high level transaction flow is depicted in the below diagram.

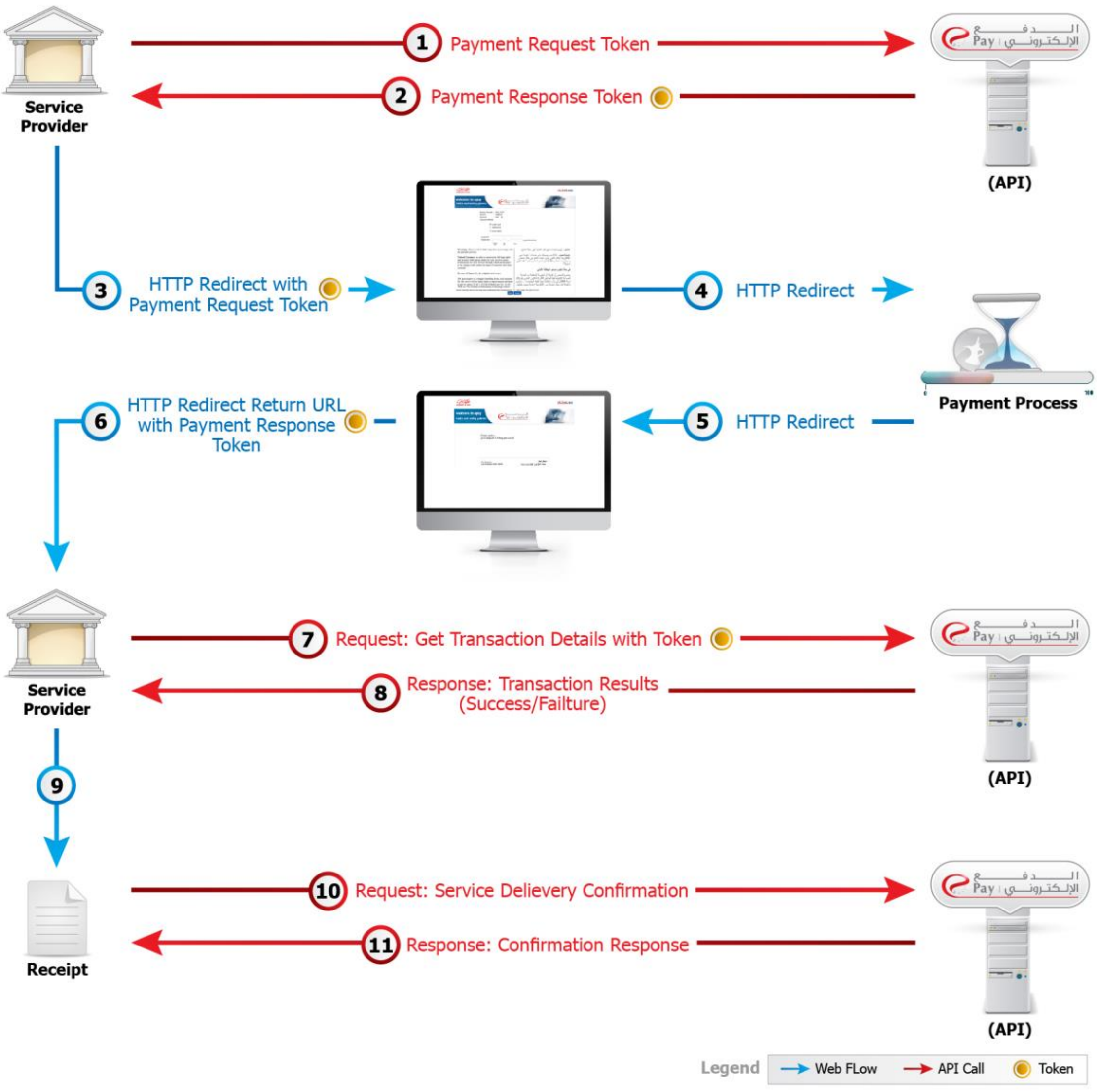

Service Provider
1 Payment Request Token
2 Payment Response Token
Pay
(API)
3 HTTP Redirect with Payment Request Token
4 HTTP Redirect
Payment Process
6 HTTP Redirect Return URL with Payment Response Token
5 HTTP Redirect
Service Provider
7 Request: Get Transaction Details with Token
8 Response: Transaction Results (Success/Failture)
(API)
9
Receipt
10 Request: Service Delievery Confirmation
11 Response: Confirmation Response
(API)
Legend
Web FLow
API Call
Token

Figure 1 – Online Integration flow

Step 1: Service Provider has to generate a payment request token with DubaiPay by sending a payment request XML with transaction, authenticated user, actual service and beneficiary details using an API.

Step 2: DubaiPay will validate the payment request, registers the transaction, generate a payment request token and send it back to the service provider.

Step 3: Service provider will redirect the customer to DubaiPay with the payment request token. DubaiPay will verify the payment request and display payment options (Credit Card/ eDirhamG2/ Direct Debit). It will prefill the email address and mobile number of the beneficiary.

Step 4: Once user chooses one of the payment option (Credit Card/ eDirhamG2/ Direct Debit/One Click Pay), User will be redirected to the Payment Gateway's/Banks

Step 4a: For ***Credit Card payment*** option, DubaiPay will redirect the user to configured credit payment processor (Comtrust or CyberSource). User has to provide credit card information. Credit Card processor will authorize the transaction and redirect the payment response back to DubaiPay.

Step 4b: For ***Direct Debit payment*** option, DubaiPay will redirect the user to the internet banking of the selected bank. User has to login and confirm the payment. After confirmation, user will be redirected back to DubaiPay.

Step 4c: For ***eDirhamG2***, user is redirected to eDirhamG2. User has to enter their edirham card number and pin. eDirham payment gatewaty will verify, process the transaction and redirect the payment response back to DubaiPay

Step 5: Payment Processor will redirect the user to DubaiPay with payment response details. DubaiPay will update the transaction status and send email notification to the user.

Step 6: DubaiPay will redirect the customer back to Service provider with the payment response token.

Step 7: Service Provider has to request payment details from DubaiPay using payment response token by invoking an API.

Step 8: DubaiPay will authenticate, verifies the token and provide the payment response back to the Service Provider.

Step 9: Service Provider has to verify payment response and provide the service to the customer.

Step 10: Service Provider has to confirm the delivery of the service to DubaiPay.

Step 11: DubaiPay will update service delivery flag and send confirmation response back to the service provider.

4.2 Integration Details

Service Provider has to do the followings:

1. Generate Payment Request Token
2. Redirection to DubaiPay using Payment Request Token
3. Receive Payment Response token
4. Get Payment Response details
5. Confirm Service Delivery

4.2.1 Generate Payment Request Token

The first step in the payment process is to generate the payment request token. Service Provider has to send the payment request XML with transaction, authenticated user, actual service and beneficiary details.

Payment Request is divided into sections as mentioned below.

S.No	Section Name	Description
1	**transactionInfo**	This section contains information of the transaction
2	**userInfo**	This section contains logged-in user information.
3	**serviceInfos**	Information about service, user is trying to pay for. It includes information about beneficiaries.

Below is the sample Payment Request for generating token.

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<epay:generateTransactionTokenRequest xmlns:epay="http://dsg.dubai.gov.ae/ws/epay"
        xmlns:com="http://dsg.dubai.gov.ae/ws/epay">
   <epay:transactionInfo>
      <epay:spCode>DP</epay:spCode>
      <epay:servCode>TFS</epay:servCode>
      <epay:sptrn>12312312344</epay:sptrn>
      <epay:amount currency="AED">100.00</epay:amount>
      <epay:timestamp>2014-05-30T09:30:10+04:00</epay:timestamp>
      <epay:description>RTA-Salik Payment</epay:description>
      <epay:type>sale</epay:type>
      <epay:versionCode>2.1</epay:versionCode>
      <epay:paymentChannel>100</epay:paymentChannel>
   </epay:transactionInfo>   <!--Optional:-->
   <epay:userInfo>
      <!--Optional:-->
      <epay:isAuthenticated>?</epay:isAuthenticated>
      <!--Optional:-->
      <epay:userId>?</epay:userId>
      <!--Optional:-->
      <epay:userName>?</epay:userName>
      <!--Optional:-->
      <epay:fullNameEn>?</epay:fullNameEn>
      <!--Optional:-->
      <epay:fullNameAr>?</epay:fullNameAr>
      <!--Optional:-->
      <epay:mobileNo>?</epay:mobileNo>
      <!--Optional:-->
      <epay:email>?</epay:email>
      <!--Optional:-->
      <epay:nationalityCode>?</epay:nationalityCode>
      <!--Optional:-->
      <epay:emiratesId>?</epay:emiratesId>
      <!--Optional:-->
      <epay:emirateCode>?</epay:emirateCode>
      <!--Optional:-->
      <epay:poBox>?</epay:poBox>
   </epay:userInfo>
   <!--Optional:-->
   <epay:serviceInfos>
      <!--1 or more repetitions:-->
      <epay:service>
         <epay:serviceNameEn>?</epay:serviceNameEn>
         <epay:serviceNameAr>?</epay:serviceNameAr>
         <epay:serviceId>?</epay:serviceId>
         <epay:gessServiceId>?</epay:gessServiceId>
         <epay:beneficiaryInfos>
            <!--1 or more repetitions:-->
            <epay:beneficiaryInfo>
               <!--Optional:-->
               <epay:accountId>?</epay:accountId>
               <!--Optional:-->
               <epay:txnAmount currency="?">?</epay:txnAmount>
               <!--Optional:-->
               <epay:fullNameEn>?</epay:fullNameEn>
               <!--Optional:-->
               <epay:fullNameAr>?</epay:fullNameAr>
               <!--Optional:-->
               <epay:mobileNo>?</epay:mobileNo>
               <!--Optional:-->
               <epay:email>?</epay:email>
               <!--Optional:-->
               <epay:emiratesId>?</epay:emiratesId>
               <!--Optional:-->
               <epay:type>?</epay:type>
               <!--Optional:-->
               <epay:companyInfo>
                  <!--Optional:-->
                  <epay:companyNameEn>?</epay:companyNameEn>
                  <!--Optional:-->
                  <epay:companyNameAr>?</epay:companyNameAr>
                  <!--Optional:-->
                  <epay:tradeLicenseNumber>?</epay:tradeLicenseNumber>
                  <!--Optional:-->
                  <epay:licenseIssuingAuthority>?</epay:licenseIssuingAuthority>
               </epay:companyInfo>
               <!--Optional:-->
               <epay:additionalParams>
                  <!--1 or more repetitions:-->
                  <com:entry>
                     <com:key>?</com:key>
                     <com:value>?</com:value>
                  </com:entry>
               </epay:additionalParams>
            </epay:beneficiaryInfo>
         </epay:beneficiaryInfos>
         <epay:additionalParams>
            <!--1 or more repetitions:-->
            <com:entry>
               <com:key>?</com:key>
               <com:value>?</com:value>
            </com:entry>
         </epay:additionalParams>
      </epay:service>
   </epay:serviceInfos>
</epay:generateTransactionTokenRequest>
```

Figure 2 – Payment Request XML

Below is the description of XML tags and XSD

Tag/Attribute	Mandatory	Description	Validation	Sample value
generateTransactionTokenRequest	Yes	Parent element		-
transactionInfo	Yes	This element will have transaction details		-
spCode	Yes	Service provider code from DSG.	Maximum 25 characters	DP
servCode	Yes	Service Code from DSG	Maximum 25 characters	TFS
sptrn	Yes	Service Provider Transaction Number Max(25) characters	Maximum 25 characters	1234567890
amount	Yes	Amount of the transaction with 2 decimal places	Any none zero decimal value with maximum 2 decimal places	100.00
currency	Yes	3 letter currency code. See section 10.8	must be valid currency code (i.e. ISO_4217)	AED

timestamp	Yes	Transaction Date in below format 2014-05-30T09:30:10+04:00	Required and default date format. ePay is also verifying the validity of the time. Time should be in Sync with ntp servers. DSG is using "ntp1.emirates.net.ae" as a source	
description	Yes	Transaction Description	Maximum 1000 characters	RTA-Salik Payment
type	Yes	Transaction Type (sale / authorize) sale: authorize and capture amount with one request authorize: Only Block the money.	sale / authorize only	sale
versionCode	Yes	Version code for the payment request from DSG	It should be 2.1	2.1
paymentChannel	Yes	Payment Channel code 100 for online, for complete list check **section 10.1**	Must be a valid value based on section 10.1 For Online payment, it should be 100	100
userInfo	Yes	This element will have logged-in user information.		-
isAuthenticated	Yes	Yes for logged-in user and anonymous for payments without any login	True or False	Yes/No
userId	No	Unique Identifier for the logged-in user from the service provider	Maximum 25 characters	1234
Username	No	Unique User name. Customer Verification process is relying on username parameter.	Maximum 100 characters	wrehman
fullNameEn	No	Full Name in English of the user	Maximum 100 characters	Aleem Ul Haq
fullNameAr	No	Full Name in Arabic of the user	Maximum 100 characters	

mobileNo	No	Mobile Number of the user. It should be 971551234567 format without any dashes	Valid mobile number \\+?([0-9]{2})?([0-9]){7,15} Maximum 25 length	971501234567
Email	No	Email address of the user	Maximum 254 characters and when specified must be a valid email address (i.e. http://en.wikipedia.org/wiki/Email_address)	abc@hotmail.com
nationalityCode	No	ISO Country code as mentioned in **section 10.6**	valid ISO 3 country code	UAE
emiratesId	No	Emirates ID of the logged-in User without any dashes	Must be 15 digits	123456789012345
emirateCode	No	Emirate code (AUH, DXB, SHJ etc) as mentioned in **section 10.7**	It should be as per the in **section 10.7**	AUH
poBox	No	Pobox of the logged-in user	Maximum 50 characters	90300
serviceInfos	Yes	Element to have list of actual services information		-
Service	Yes	Element to have actual service information		-
serviceId	No	Unique Service Id of the Service	Maximum 25 characters	
gessServiceId	No	DSG eServices Statistics System (GeSS) Service Id from DSG	Maximum 25 characters	
serviceNameEn	Yes	Actual Service Name (customer is trying to Pay) in English	Maximum 250 characters	DEWA Bill Payment
serviceNameAr	No	Actual Service Name in Arabic	Maximum 250 characters	
beneficiaryInfos	No	This section will have list of beneficiaries information		-
beneficiaryInfo	No	Beneficiary information		-

accountId	No	Account Id, unique identifier of the account for example: Contract Account Number for DEWA Salik Account Number for Salik Payments	Maximum 25 characters	1234567
txnAmount	No	Beneficiary Account Transaction amount. In-case of one service payment. It will be same as in transaction info section <epay:amount>	Any none zero decimal value with maximum 2 decimal places	100.00
fullNameEn	No	Name of the Beneficiary in English	Maximum 100 characters	-
fullNameAr	No	Name of the Beneficiary in Arabic	Maximum 100 characters	-
Type	No	Beneficiary Type (Individual, Corporate, Government)		Corporate
mobileNo	No	Mobile Number of the Beneficiary. It should be 971551234567 format without any dashes	Valid mobile number \\+?([0-9]{2})?([0-9]){7,15} Maximum 25 length	971551234567
Email	No	Email Address of the Beneficiary	Maximum 254 characters and when specified must be a valid email address (i.e. http://en.wikipedia.org/wiki/Email_address)	
emiratesId	No	Emirates Id of the Beneficiary	Must be 15 digits	
companyInfo	No	Company Information		-
companyNameEn	No	Name of the Company in English	Maximum 200 characters	
companyNameAr	No	Name of the Company in Arabic	Maximum 200 characters	-
tradeLicenseNumber	No	Trade License Number	Maximum 100 characters	
licenseIssuingAuthority	No	License Issuing Authority (DED, Free Zone etc)	Maximum 100 characters	

additionalParams	No	This section will have additional parameters which can be sent by the government department to ePayment Gateway. In-case of no additional parameter it will be empty	Maximum 4000 characters for all additional parameters	
Entry	No	Additional Parameter		
Key	No	Name of the Parameter		
Value	No	Value of the Parameter		

- To avoid duplicate transaction, service provider has to ensure that SPTRN transaction number is unique.

  Service Provider has to sign and encrypt each SOAP request .It will be used by DubaiPay to verify the authenticity of the payment request
- After sending payment request to DSG, service provider has to mark transaction as “**Pending**” to avoid duplicate payment.

- **Schemas and WSDL**

common_types.xsd	payment_service.wsdl	epay_schema.xsd

Service Provider can generate payment request token using ePayment API. It is protected with ***WS-Security Sign and encryption policy***. Service Provider has to consult DSG team for PKI certificate generation.

Payment Request Token via SOAP:

Service provider can use “***generateTransactionToken***” operation in the web service to generate payment request token. Below are the input parameters:

S.No	Web Service Parameter Name	Type	Description
1	**transactionInfo**	Object	Transactions information
2	**userInfo**	Object	Logged in user information
3	**serviceInfos**	Object	Service and Beneficiary information

Staging URL: https://epayment.qa.dubai.ae/ePayHub/WSDL/PaymentAPIService.wsdl
Production URL: https://epayment.dubai.ae/ePayHub/WSDL/PaymentAPIService.wsdl

This web service is protected with ***WS-Security Sign and Encryption policy***.

Response from Payment Request Token API:
Below is the response of Payment request token API.

```
<epay:generateTransactionTokenResponse  valid="true"
xmlns:ns2="http://dsg.dubai.gov.ae/schema/epay"><uri>https://epayment.qa.dubai.ae/ePay
Hub/Authentication/SPServlet?token=C6D93BA757B3AC6BB93A71C713297E38FDAB15BFB106BFF5613
E2FA29FDE470187DBF6DE00FAE79379D78C0F6634655C84FF99C666B9747B2CB3C576B99753E3ED9B82936
9EA9610C46C46AFF485B308EB14C5B13D1379E095B75B8CD30B68C0</uri>
</epay:generateTransactionTokenResponse>
```

Service Provider has to ensure that response is valid and redirect the user to the URI mentioned in the response. In-case of invalid response, Service provider has to display an error message to the customer and contact DSG team for further investigation.

4.2.2 Redirection to ePay using Payment Request Token

After generating the payment request, service provider has to redirect the user to the URI received in the response.

4.2.3 Receive Payment Response Token

Once user will complete the payment process, ePay will send a payment response token to the service provider as HTTP post. Service Provider has to implement a page to receive the response. This URI is shared with DSG team during the rollout.

S.No	Parameter Name	Description
1	**TOKEN**	Payment Response Token

4.2.4 Get Payment Response Details

After receiving the response, Service Provider has to get payment response details using ePayment API.

Payment Response Details via SOAP Web Service:

Service provider can use “**getReponseTokenDetails**” operation in the web service to get the details of the payment from ePay. The following are the parameters:

S.No	Web Service Parameter Name	Type	Description
1	**responseToken**	String	Response Token Received from ePay
2	**spCode**	String	Service Provider Code from DSG
3	**servCode**	String	Service Code from DSG

Staging URL: https://epayment.qa.dubai.ae/ePayHub/WSDL/PaymentAPIService.wsdl
Production URL: https://epayment.dubai.ae/ePayHub/WSDL/PaymentAPIService.wsdl

This web service is protected with ***WS-Security Sign and encryption policy***. Service Provider has to consult DSG team for PKI certificate generation.

Response from Get Payment Response Token Details API:

Below is the response of successful payment

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<epay:responseTokenDetailsResponse valid="true">
  <epay:spCode>DP</epay:spCode>
  <epay:servCode>TFS</epay:servCode>
  <epay:sptrn>123465</epay:sptrn>
  <epay:degTrn>00000012234</epay:degTrn>
  <epay:txnTimestamp>2001-12-31T12:00:00</epay:txnTimestamp>
  <epay:paymentMethod>sale</epay:paymentMethod>
  <epay:message>
    <comm:code>0</ comm:code>
    <comm:text>Successful Payment</comm:text>
  </epay:message>
</epay:responseTokenDetailsResponse>
```

Figure 3 – Payment Response Details

Service Provider has to verify that response is valid by checking the valid flag.

Below is the list of XML tags in the response.

Tag	Mandatory	Description	Sample value
responseTokenDetailsResponse	Yes	Parent element for Payment Response with an attribute that response is valid	-
spCode	No	Service provider code from DSG	DP
servCode	No	Service Code from DSG	TFS
swptrn	No	Service Provider Transaction Number	1234567890
degTrn	No	ePayment Gateway unique payment reference number	00001112345
transDate	No	Transaction Date in “2002-05-30T09:30:10+04:00” format	
paymentMehod	No	Payment options (Credit Card, Direct Debit, eDirhamG2 etc	Credit Card
Text	Yes	Message of the transaction	Successful Transaction
Code	Yes	Message Code 0 means successful transaction any other response codes are considered as failure transaction	0

- After receiving payment confirmation response from DSG, service provider has to mark transaction as “**Success / Failure**” based on the Message code.

  Note: “0” message code means successful transactions any other response code should be consider as failure transactions

- Service provider will provide the online transaction receipt to the customer. Upon Successful payment the service for which the customer has made the payment need to be delivered as per the terms and conditions agreed before initiating the payment.

4.2.5 Service Delivery Confirmation

After providing the service to the user Service Provider has to confirm the service delivery. Below are the details of confirming the service delivery to ePay.

Service Delivery Confirmation via SOAP Web Service:

Service provider can use “**confirmServiceDelivery**” operation in the web service to confirm the delivery of the service. Below are the parameters:

S.No	Web Service Parameter Name	Type	Description
1	**spCode**	String	Service Provider Code from DSG
2	**servCode**	String	Service Code from DSG
3	**sptrn**	String	Service Provider Transaction number
4	**code**	String	Message code received in the **getReponseTokenDetails. It should be "0" for successful payment**
5	**text**	String	Message code received in the **getReponseTokenDetails.**

Staging URL: **https://epayment.qa.dubai.ae/ePayHub/WSDL/PaymentAPIService.wsdl**
Production URL: **https://epayment.dubai.ae/ePayHub/WSDL/PaymentAPIService.wsdl**

This web service is protected with ***WS-Security Sign and encryption policy***. Service Provider has to consult DSG team for PKI certificate generation.

Response from Service Delivery Confirmation API:

Below is the response of successful confirmation response from ePay.

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<epay:serviceDeliveryConfirmationResponse>
      <epay:message>
            <com:code>0</com:code>
            <com:text>Service Delivered</com:text>
      </epay:message>
</epay:serviceDeliveryConfirmationResponse>
```

Figure 4 – Service Delivery Confimation Response

Note: This operation is required for successful payment only.

DSG production environment is only accessible over GIN network. In-case department is not using GIN network, a request needs to be send to DSG team for network connectivity.

5. Chapter Five: Web Service Integration Profile

In this Chapter we will explain how the payment flow works based on web service integration profile, it is essential to understand the workflow of the payment before we can explain the integration details in the second part of this Chapter.

5.1 How it works?

In this integration approach, Service Provider collects all the credit card related information from the user and sends it to the Payment Gateway for processing. This model is also adapted for payments where redirection is not an option like Smart Phones, POS terminals, MOTO transactions and other devices.

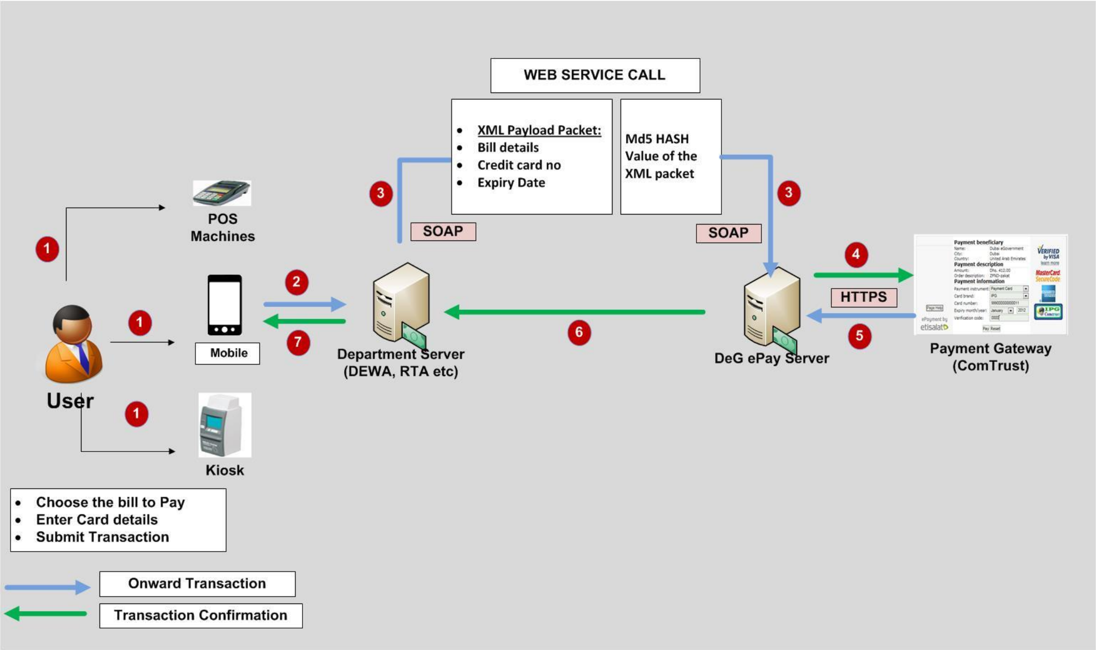


Figure 5 – POS Payment Flow

Step 1: Customer uses mobile app, kiosk or POS machine to pay government transactions.

Step 2: Customer chooses the bill he wants to pay enters the card details and submits the transaction.

Step 3: Service Provider create an xml request that contains the transaction details, logged-in user, service, beneficiary and credit card details. A hash value of the xml request is also generated.

Step 3.1: ePay will ensures the authenticity of the payment request by verifying the secure hash parameter.

Step 4: After verification, transaction is sent to the payment processor (Comtrust) for further processing.

Step 5: Payment is processed by payment processor and response is shared with ePay.

Step 6: Response is routed back to department servers.

Step 7: Service Provider has to validate the authenticity of the payment response, based on the status of the payment provide the service to the customer.

5.2 Integration Details

Due to PCI-DSS compliance, this payment options is not supported by DSG.

6. Chapter Six: Authorize Integration Profile

In this Chapter we will explain how the payment flow works based on authorize integration profile, it is essential to understand the workflow of the payment before we can explain the integration details in the second part of this Chapter.

6.1 How it works?

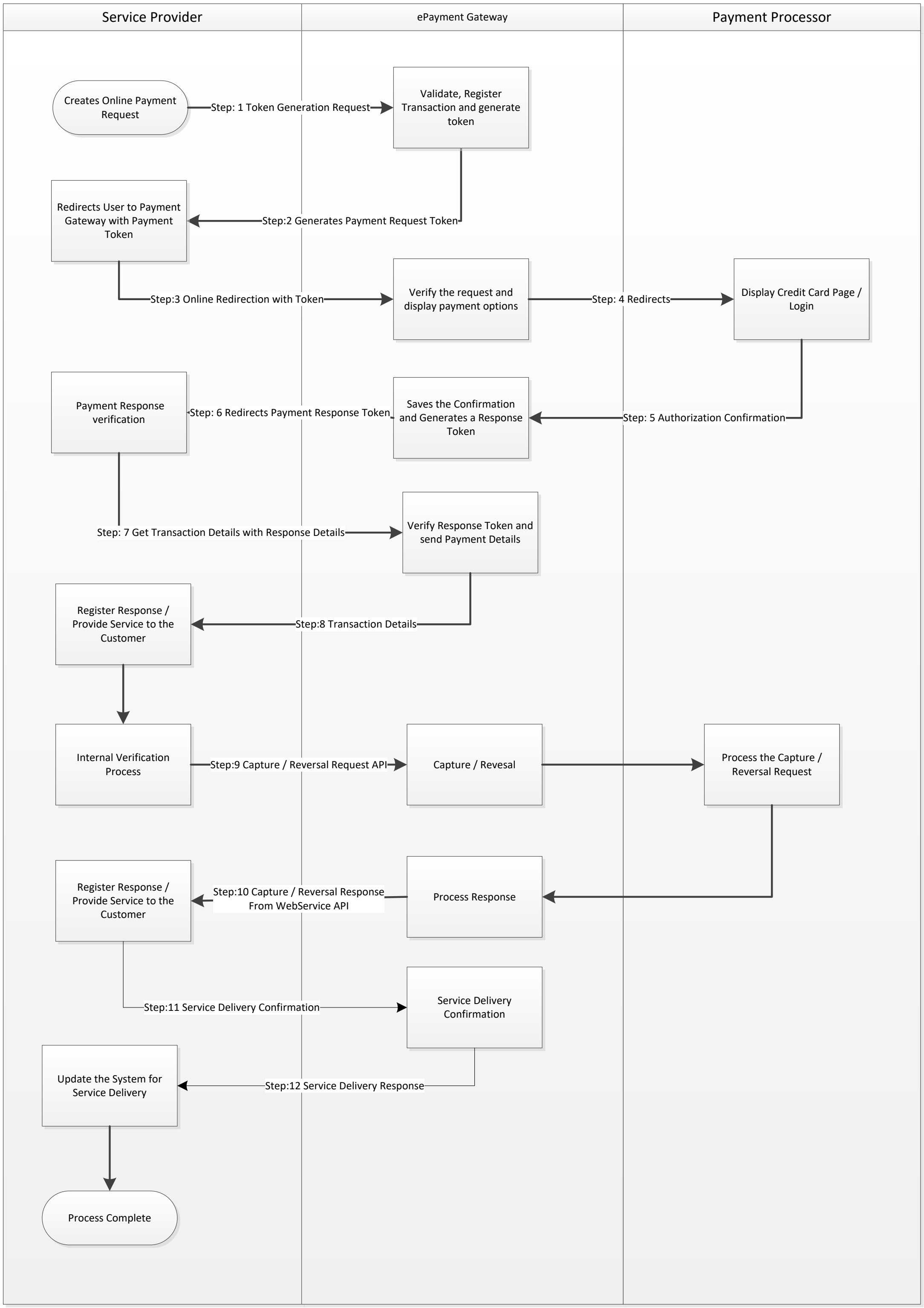
In some cases, service provider has to block the amount and provider the service after manual verification. DSG payment gateway allows the service providers to block the amount using online integration profile. This approach is also secure as credit card related information are provided to the Payment Processor and service providers are not transmitting sensitive credit card data using Web Service API integration.

Below are the steps for above approach.

1. Department will follow the same online integration steps as mentioned in **Section 4**. They have to pass a parameter "**TxnType as Authorize**" with the payment request. This will authorize the amount only.
2. Service Provider will receive the confirmation from DSG payment gateway and amount will be blocked for maximum 15 days. The maximum block period for a transaction varies for each card issuer.
3. Service Provider can have their own internal process / review before taking a decision to provide a service to the customer.
4. Service Providers sends a Capture / Reversal request using Web Service API.
5. ePay will verify the Capture / Reversal request.
6. ePay will send these requests to Payment Processor.
7. ePay will send the response back to the Service Provider.
8. Department will send the service delivery confirmation request to ePay.
9. ePay will update the service delivery flag and send confirmation response back to the service provider.

Note: This option is only available for Credit Card Payment method

Below is the process of this integration.



Service Provider
ePayment Gateway
Payment Processor
Creates Online Payment Request
Step: 1 Token Generation Request
Validate, Register Transaction and generate token
Redirects User to Payment Gateway with Payment Token
Step:2 Generates Payment Request Token
Step:3 Online Redirection with Token
Verify the request and display payment options
Step: 4 Redirects
Display Credit Card Page / Login
Payment Response verification
Step: 6 Redirects Payment Response Token
Saves the Confirmation and Generates a Response Token
Step: 5 Authorization Confirmation
Step: 7 Get Transaction Details with Response Details
Verify Response Token and send Payment Details
Register Response / Provide Service to the Customer
Step:8 Transaction Details
Internal Verification Process
Step:9 Capture / Reversal Request API
Capture / Revesal
Process the Capture / Reversal Request
Register Response / Provide Service to the Customer
Step:10 Capture / Reversal Response From WebService API
Process Response
Step:11 Service Delivery Confirmation
Service Delivery Confirmation
Update the System for Service Delivery
Step:12 Service Delivery Response
Process Complete

Figure 6 –Authorization Integration flow

6.2 Integration Details

Due to PCI-DSS compliance, this payment options is not supported by DSG.

7. Chapter Seven: Transaction Inquiry

In this Chapter we will explain transaction inquiry from service providers, it is essential to understand the scenarios of the transaction inquiry before we can explain the integration details in the second part of this Chapter.

7.1 How it works?

Service provider has to mark transaction as pending after redirecting to the payment gateway. It is to avoid duplicate payment of fines / bill. Once ePayment Gateway receives an inquiry request, it will check the status of the transaction as below:

Case 1: **Successful / Failure**	If a transaction is successful / failure from the bank, ePay will send Success / Failure response back to the service provider and they have to mark transaction these transaction as successful or failure. And in-case of successful transaction they have to provide service to the customer.
Case 2: **In-Progress**	If transaction is in progress for less than **30** minutes. ePay will send an in-progress response “80013 and 80014” to the service provider and they have do another inquiry after 30 minutes.
Case 3: **SP Terminated**	If a transaction is in-progress for more than **30** minutes. ePay will terminate the transaction and send to the service provider “SPTERMINATED” response code “11”. Service provider should consider this transaction as failure.

The message code and message returned in this operation are mentioned in the table below

S No	Message	Message Code
1	SUCCESS	0
2	CANCELLED	10
3	SP TERMINATED	11
4	FAILURE	Code from PG
5	Pending Capture	80013
6	Transaction in Progress	80014

7.2 Integration Details

Service providers have to implement a timer routine to get the status of all pending transactions. DSG ePayment gateway provides following web service to query the status of the transaction.

7.2.1 Transaction Status via Web service

Service provider can use Transaction Inquiry web service to get the status of a transaction from ePay.

Operation (getTransactionStatus):

The input parameters for this operation are

S.No	Parameter Name	Type	Description
1	**SPCODE**	String	Service Provider code from DSG
2	**SERVCODE**	String	Service Code from DSG
3	**SPTRN**	String	Service Provider transaction number

Staging: https://epayment.qa.dubai.ae/ePayHub/epaynmservicewar/epaynmwebservice?WSDL
Production: https://epayment.dubai.ae/ePayHub/epaynmservicewar/epaynmwebservice?WSDL

DSG will be providing an SDK for java which can be used to invoke transaction inquiry web service. Service Provider can also implement the web service client based on the WSDL file.

Sample Code in Java:

Below is the sample code

```
public class CallWS
{
public static void main(String[] args)
{
String spCode = "spCode";
String servCode = "servCode";
EPayNMWebServiceHandler epayNMServiceHandler = new EPayNMWebServiceHandler();
epayNMServiceHandler.setPropertyFile(spCode+"."+servCode);
Hashtable resultMap = epayNMServiceHandler.getTransactionStatus(String spCode,
String servCode, String spTrn);
}
<%
      Service Provider’s business logic goes here.
%>
}}
```

Figure 7 –Transaction Inquiry via SOAP

Response from the Web Service:

Output from the Web Service call for Success Transaction will have a hash table with the following values

S No	Description	Field Name	Sample Value
1	Service Provider Code	SPCODE	DP
2	Service Code	SERVCODE	PF
3	Service Provider Transaction No.	SPTRN	999999999(**Max : 25 characters**)
4	DEG Transaction No	DEGTRN (This value will not be returned if the transaction status at DEG is DEG Landed(1))	999999999(**Max : 25 characters**)
5	Transaction Date	TRANSDATE	dd/mm/yyyy hh:mi:ss
6	Payment Method Name	PYMTMETHOD	Credit Card
7	Message	MESSAGE	SUCCESS
8	Message Code	MESSAGECODE	0

Output from the Web Service call for invalid transaction search criteria will have a hash table with the following values

S No	Description	Field Name	Sample Value
1	Message Code	MESSAGECODE	70011
2	Message	MESSAGE	Invalid Transaction Search Criteria

8. Chapter Eight: Reconciliation

Reconciliation is a process to verify the transactions status from Service Providers with DSG ePayment gateway. After verification, DOF will settle the money to the government department account.

ePayment gateway provides following options for the reconciliation:

8.1 Automatic reconciliation

Reconciliation can be performed automatically where Service provider implements a webservice interface which provides the transaction details to DSG ePayment Gateway automatically.

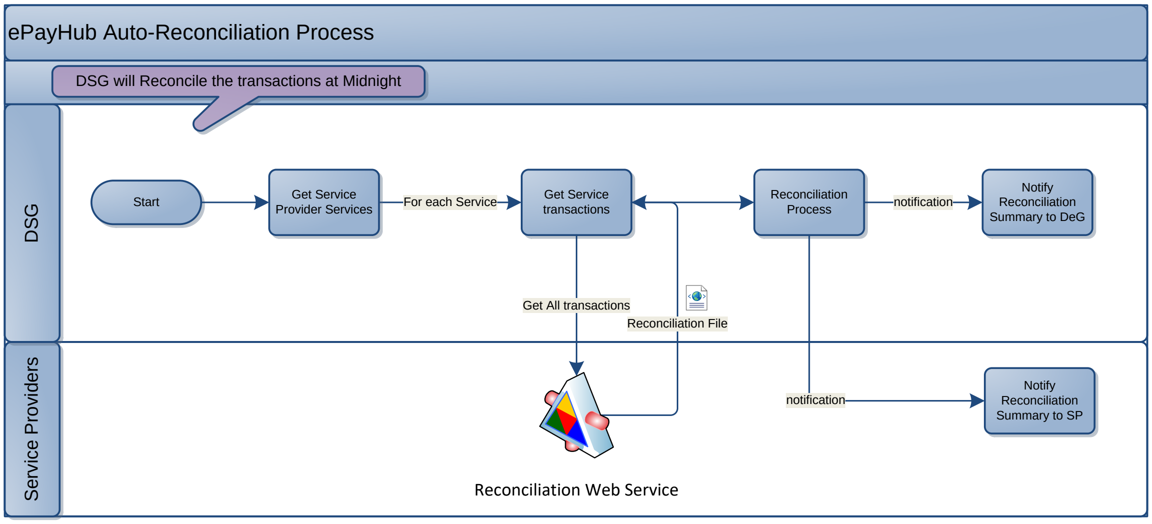

ePayHub Auto-Reconciliation Process
DSG will Reconcile the transactions at Midnight
DSG
Service Providers
Start
Get Service Provider Services
For each Service
Get Service transactions
Reconciliation Process
notification
Notify Reconciliation Summary to DeG
Get All transactions
Reconciliation File
notification
Notify Reconciliation Summary to SP
Reconciliation Web Service

Figure 8–Automatic reconciliation Flow

DSG is providing readymade component for .Net and Java to the government departments. Government department can also implement the web service based on the WSDL file.

Below is the WSDL file and XSD. DSG will be providing separate interface guide for auto reconciliation and readymade packages.

WSDL	XSD
SPReconciliationService.WSDL	SPReconciliationService.xsd

8.2 Manual Reconciliation

In some cases, if web service interface is not possible, Service Provider can provide a screen for DOF to generate the reconciliation file manually. Finance Users will use this URL and download the data and upload to the DSG system to do the Reconciliation.

The Service Providers are expected to provide a web interface to with the following filters.

- Transaction From Date
- Transaction To Date
- SP Transaction Number
- DSG Transaction Number
- PG Transaction Number
- Transaction Status (All, Success, Failure)

Along with the transaction details, the Service Provider Code and the Service Code should be available in the Service Provider Details section of the XML.

Sample XML File :

```
<Reconciliation>
   <ServiceProviderDetails>
      <SPCODE>DOHMS</SPCODE>
      <SERVCODE>DOHM</SERVCODE>
   </ServiceProviderDetails>
   <TransactionDetails>
      <Transaction>
         <SPTRN>999999999</SPTRN>
         <TransDate>27/11/2004 23:30:45</TransDate>
         <Amount>999.99</Amount>
         <DEGTRN>999999999</DEGTRN>
         <Status>0</Status>
         <PaymentMethod>CreditCard</PaymentMethod>
      </Transaction>
      <Transaction>
         <SPTRN>999999999</SPTRN>
         <TransDate>27/11/2004 23:30:45</TransDate>
         <Amount>999.99</Amount>
         <DEGTRN>999999999</DEGTRN>
         <Status>1</Status>
         <PaymentMethod>CreditCard</PaymentMethod>
      </Transaction>
   </TransactionDetails>
</Reconciliation>
</Reconciliation >
```

Figure 9–Reconciliation XML

XML Schema Definition:

```
<?xml version=”1.0” encoding=”UTF-8”?>

<xs:schema xmlns:xs=”http://www.w3.org/2001/XMLSchema” elementFormDefault=”qualified”>


        <xs:element name=”Reconciliation”>
                <xs:complexType>
                        <xs:sequence>
                                <xs:element ref=”ServiceProviderDetails” maxOccurs=”1”/>
                                <xs:element ref=”TransactionDetails” maxOccurs=”1”/>
                        </xs:sequence>
                </xs:complexType>
        </xs:element>

        <xs:element name=”ServiceProviderDetails”>
                <xs:complexType>
                        <xs:sequence>
                                <xs:element ref=”SPCODE”/>
                                <xs:element ref=”SERVCODE”/>
                        </xs:sequence>
                </xs:complexType>
        </xs:element>


        <xs:element name=”SERVCODE” type=”xs:string”/>
        <xs:element name=”SPCODE” type=”xs:string”/>


        <xs:element name=”TransactionDetails”>
                <xs:complexType>
                        <xs:sequence>
                                <xs:element ref=”Transaction” minOccurs=”0”
maxOccurs=”unbounded”/>
                        </xs:sequence>
                </xs:complexType>
        </xs:element>


        <xs:element name=”Transaction”>
                <xs:complexType>
                        <xs:sequence>
                                <xs:element ref=”SPTRN”/>
                                <xs:element ref=”TransDate”/>
                                <xs:element ref=”Amount”/>
                                <xs:element ref=”DEGTRN”/>
                                <xs:element ref=”Status”/>
                                <xs:element ref=”PaymentMethod”/>
                        </xs:sequence>
                </xs:complexType>
```

```
       </xs:element>

       <xs:element name=”SPTRN” type=”xs:string”/>

       <xs:element name=”TransDate” type=”xs:string”/>

       <xs:element name=”Amount” type=”xs:string”/>

       <xs:element name=”DEGTRN” type=”xs:string”/>

       <xs:element name=”Status” type=”xs:integer”/>

       <xs:element name=”PaymentMethod”>

              <xs:simpleType>
                     <xs:restriction base=”xs:string”>
                            <xs:enumeration value=”CreditCard”/>
                            <xs:enumeration value=”BizDirect”/>
                     </xs:restriction>
              </xs:simpleType>

       </xs:element>

</xs:schema>
```

Figure 10–Reconciliation XSD

9. Chapter Nine: Samples Codes

DSG will be sharing sample application in .Net / Java / SOAP UI to assist government departments in the rollout.

10. Appendix

This section will have the list of Payment Channels, Payment Modes, Transaction Statuses, Common Error Codes and Messages, etc.

10.1 List of Payment Channels

Below is the list of payment channels.

S No	Payment Channel Code	Payment Channel Name	Entry point allowed
1	100	Online	Web Redirection
2	101	KIOSK	Web Service Integration
3	102	IVR	Web Service Integration
4	103	POS	Web Service Integration
5	104	MOBILE	Web Service Integration

10.2 List of Payment Methods

Below is the list of payment methods.

S No	Payment Method Code	Payment Method Name
1	10000	Credit Card
2	10010	Direct Debit
3	10008	eDirhamG2
4	10012	One Click Pay

10.3 List of Operation Types

Below is the list of operation types.

S No	Operation Type	Description
1	AUTHCAPTURE	This operation is used to authorize and deduct the money from customer account with one request.
2	AUTHORIZE	This operation is used to authorize transaction amount. Customer amount will be on-hold. Service Provider can send the capture / reversal request. This operation is used by Service providers where manual verification is required before providing the service to the customer.

3	CAPTURE	This operation is used to complete the transaction. It can only be used after the authorization only
4	REVERSAL	This operation is used to reverse the authorization. It can be used after the authorization only
5	FINALIZATION	This operation is used to confirm the authorization. It can be used after online authentication.

10.4 Common Error Messages / Error Codes

Below is the list of error codes

S No	Error Code	Source of Error	Error Message
1	0 – 9999	COMTRUST	The error message received from the COMTRUST will be sent to the Service Providers as it is.
2	10000 – 99999	DSG	Error generated at DSG’s end
3	10001	DSG	Payment Mode is temporarily blocked
4	10002	DSG	User has already been authorized
5	11111	DSG	There’s a critical problem at our end. Please try again later
6	88888	DSG	Database Down. Try again later
7	99999	DSG	Database Problem. Contact Portal Administrator
8	70000 -79999	DSG	Generated by the java client component (wsclient.jar).Service Provider to check inputs. Or try again at a later time.
9	70002	Java Component	Please Check your inputs
10	70003	Java Component	Please check the url you are trying to access
11	70004	Java Component	Unable to connect to DSG. Please try again later
12	70006	Java Component	Problem with your configuration

13	70007	Java Component	Service Provider Transaction Number cannot be empty
14	70008	Java Component	DSG Transaction Number cannot be empty
15	70009	Java Component	Service Provider Transaction Number cannot be empty
16	70010	Java Component	DSG Transaction Number cannot be empty
17	70011	Java Component	Invalid Transaction Search Criteria
18	70012	Java Component	Please Enter the Transaction Search Criteria
19	10016	DSG	Transaction is timed out. Please try to Pay again
20	TXN0001	DSG	Request could not be completed – TXN0001
21	TXN0002	DSG	Checksum Failed – TXN0002
22	TXN0003	DSG	Checksum Failed – TXN0003
23	TXN0004	DSG	Validation Failed – TXN0004
24	TXN0005	DSG	Validation Failed – TXN0005
25	TXN0006	DSG	Secure Hashcode Empty-TXN0006
26	80011	Java Component	Timeout
27	80013	DSG	Transaction in Progress
28	80014	DSG	Transaction in Progress

10.5 Emirates Codes

Below is the list of Emirate Codes

Emirates Code	Emirates Name
AUH	Abu Dhabi
DXB	Dubai
SHJ	Sharjah
AJM	Ajman
UAQ	Umm Al Quwain
RAK	Ras Al Khaimah
FUJ	Fujairah

10.6 Country Code

Below is the list of country code

Country or Area name	ISO Code
Afghanistan	AFG
Åland Islands	ALA
Albania	ALB
Algeria	DZA
American Samoa	ASM
Andorra	AND
Angola	AGO
Anguilla	AIA
Antigua and Barbuda	ATG
Argentina	ARG
Armenia	ARM
Aruba	ABW
Australia	AUS
Austria	AUT
Azerbaijan	AZE
Bahamas	BHS
Bahrain	BHR
Bangladesh	BGD
Barbados	BRB
Belarus	BLR
Belgium	BEL
Belize	BLZ
Benin	BEN
Bermuda	BMU
Bhutan	BTN
Bolivia	BOL
Bosnia and Herzegovina	BIH
Botswana	BWA
Brazil	BRA
British Virgin Islands	VGB
Brunei Darussalam	BRN
Bulgaria	BGR

Burkina Faso	BFA
Burundi	BDI
Cambodia	KHM
Cameroon	CMR
Canada	CAN
Cape Verde	CPV
Cayman Islands	CYM
Central African Republic	CAF
Chad	TCD
Chile	CHL
China	CHN
Hong Kong Special Administrative Region of China	HKG
Macao Special Administrative Region of China	MAC
Colombia	COL
Comoros	COM
Congo	COG
Cook Islands	COK
Costa Rica	CRI
Côte d'Ivoire	CIV
Croatia	HRV
Cuba	CUB
Cyprus	CYP
Czech Republic	CZE
Democratic People's Republic of Korea	PRK
Democratic Republic of the Congo	COD
Denmark	DNK
Djibouti	DJI
Dominica	DMA
Dominican Republic	DOM
Ecuador	ECU
Egypt	EGY
El Salvador	SLV
Equatorial Guinea	GNQ
Eritrea	ERI
Estonia	EST
Ethiopia	ETH
Faeroe Islands	FRO
Falkland Islands (Malvinas)	FLK
Fiji	FJI

Finland	FIN
France	FRA
French Guiana	GUF
French Polynesia	PYF
Gabon	GAB
Gambia	GMB
Georgia	GEO
Germany	DEU
Ghana	GHA
Gibraltar	GIB
Greece	GRC
Greenland	GRL
Grenada	GRD
Guadeloupe	GLP
Guam	GUM
Guatemala	GTM
Guernsey	GGY
Guinea	GIN
Guinea-Bissau	GNB
Guyana	GUY
Haiti	HTI
Holy See	VAT
Honduras	HND
Hungary	HUN
Iceland	ISL
India	IND
Indonesia	IDN
Iran, Islamic Republic of	IRN
Iraq	IRQ
Ireland	IRL
Isle of Man	IMN
Israel	ISR
Italy	ITA
Jamaica	JAM
Japan	JPN
Jersey	JEY
Jordan	JOR
Kazakhstan	KAZ
Kenya	KEN

Kiribati	KIR
Kuwait	KWT
Kyrgyzstan	KGZ
Lao People's Democratic Republic	LAO
Latvia	LVA
Lebanon	LBN
Lesotho	LSO
Liberia	LBR
Libyan Arab Jamahiriya	LBY
Liechtenstein	LIE
Lithuania	LTU
Luxembourg	LUX
Madagascar	MDG
Malawi	MWI
Malaysia	MYS
Maldives	MDV
Mali	MLI
Malta	MLT
Marshall Islands	MHL
Martinique	MTQ
Mauritania	MRT
Mauritius	MUS
Mayotte	MYT
Mexico	MEX
Micronesia, Federated States of	FSM
Moldova	MDA
Monaco	MCO
Mongolia	MNG
Montenegro	MNE
Montserrat	MSR
Morocco	MAR
Mozambique	MOZ
Myanmar	MMR
Namibia	NAM
Nauru	NRU
Nepal	NPL
Netherlands	NLD
Netherlands Antilles	ANT
New Caledonia	NCL

New Zealand	NZL
Nicaragua	NIC
Niger	NER
Nigeria	NGA
Niue	NIU
Norfolk Island	NFK
Northern Mariana Islands	MNP
Norway	NOR
Occupied Palestinian Territory	PSE
Oman	OMN
Pakistan	PAK
Palau	PLW
Panama	PAN
Papua New Guinea	PNG
Paraguay	PRY
Peru	PER
Philippines	PHL
Pitcairn	PCN
Poland	POL
Portugal	PRT
Puerto Rico	PRI
Qatar	QAT
Republic of Korea	KOR
R_union	REU
Romania	ROU
Russian Federation	RUS
Rwanda	RWA
Saint-Barthélemy	BLM
Saint Helena	SHN
Saint Kitts and Nevis	KNA
Saint Lucia	LCA
Saint-Martin (French part)	MAF
Saint Pierre and Miquelon	SPM
Saint Vincent and the Grenadines	VCT
Samoa	WSM
San Marino	SMR
Sao Tome and Principe	STP
Saudi Arabia	SAU
Senegal	SEN

Serbia	SRB
Seychelles	SYC
Sierra Leone	SLE
Singapore	SGP
Slovakia	SVK
Slovenia	SVN
Solomon Islands	SLB
Somalia	SOM
South Africa	ZAF
Spain	ESP
Sri Lanka	LKA
Sudan	SDN
Suriname	SUR
Svalbard and Jan Mayen Islands	SJM
Swaziland	SWZ
Sweden	SWE
Switzerland	CHE
Syrian Arab Republic	SYR
Tajikistan	TJK
Thailand	THA
The former Yugoslav Republic of Macedonia	MKD
Timor-Leste	TLS
Togo	TGO
Tokelau	TKL
Tonga	TON
Trinidad and Tobago	TTO
Tunisia	TUN
Turkey	TUR
Turkmenistan	TKM
Turks and Caicos Islands	TCA
Tuvalu	TUV
Uganda	UGA
Ukraine	UKR
United Arab Emirates	ARE
United Kingdom of Great Britain and Northern Ireland	GBR
United Republic of Tanzania	TZA
United States of America	USA
United States Virgin Islands	VIR
Uruguay	URY

Uzbekistan	UZB
Vanuatu	VUT
Venezuela (Bolivarian Republic of)	VEN
Viet Nam	VNM
Wallis and Futuna Islands	WLF
Western Sahara	ESH
Yemen	YEM
Zambia	ZMB
Zimbabwe	ZWE

10.7 Currency Codes

Below is the list of Currency Codes code

Currency Code	Currency
AED	United Arab Emirates dirham
AFN	Afghani
ALL	Lek
AMD	Armenian Dram
ANG	Netherlands Antillian Guilder
AOA	Kwanza
ARS	Argentine Peso
AUD	Australian Dollar
AWG	Aruban Guilder
AZN	Azerbaijanian Manat
BAM	Convertible Marks
BBD	Barbados Dollar
BDT	Bangladeshi Taka
BGN	Bulgarian Lev
BHD	Bahraini Dinar
BIF	Burundian Franc
BMD	Bermudian Dollar (customarily known as Bermuda Dollar)
BND	Brunei Dollar
BOB	Boliviano
BOV	Bolivian Mvdol (Funds code)
BRL	Brazilian Real
BSD	Bahamian Dollar
BTN	Ngultrum
BWP	Pula
BYR	Belarussian Ruble
BZD	Belize Dollar
CAD	Canadian Dollar
CDF	Franc Congolais
CHE	WIR Euro (complementary currency)
CHF	Swiss Franc
CHW	WIR Franc (complementary currency)
CLF	Unidades de formento (Funds code)
CLP	Chilean Peso

CNY	Yuan Renminbi
COP	Colombian Peso
COU	Unidad de Valor Real
CRC	Costa Rican Colon
CUP	Cuban Peso
CVE	Cape Verde Escudo
CYP	Cyprus Pound
CZK	Czech Koruna
DJF	Djibouti Franc
DKK	Danish Krone
DOP	Dominican Peso
DZD	Algerian Dinar
EEK	Kroon
EGP	Egyptian Pound
ERN	Nakfa
ETB	Ethiopian Birr
EUR	Euro
FJD	Fiji Dollar
FKP	Falkland Islands Pound
GBP	Pound Sterling
GEL	Lari
GHS	Cedi
GIP	Gibraltar pound
GMD	Dalasi
GNF	Guinea Franc
GTQ	Quetzal
GYD	Guyana Dollar
HKD	Hong Kong Dollar
HNL	Lempira
HRK	Croatian Kuna
HTG	Haiti Gourde
HUF	Forint
IDR	Rupiah
ILS	New Israeli Shekel
INR	Indian Rupee
IQD	Iraqi Dinar
IRR	Iranian Rial
ISK	Iceland Krona
JMD	Jamaican Dollar
JOD	Jordanian Dinar
JPY	Japanese yen
KES	Kenyan Shilling

KGS	Som
KHR	Riel
KMF	Comoro Franc
KPW	North Korean Won
KRW	South Korean Won
KWD	Kuwaiti Dinar
KYD	Cayman Islands Dollar
KZT	Tenge
LAK	Kip
LBP	Lebanese Pound
LKR	Sri Lanka Rupee
LRD	Liberian Dollar
LSL	Loti
LTL	Lithuanian Litas
LVL	Latvian Lats
LYD	Libyan Dinar
MAD	Moroccan Dirham
MDL	Moldovan Leu
MGA	Malagasy Ariary
MKD	Denar
MMK	Kyat
MNT	Tugrik
MOP	Pataca
MRO	Ouguiya
MTL	Maltese Lira
MUR	Mauritius Rupee
MVR	Rufiyaa
MWK	Kwacha
MXN	Mexican Peso
MXV	Mexican Unidad de Inversion (UDI) (Funds code)
MYR	Malaysian Ringgit
MZN	Metical
NAD	Namibian Dollar
NGN	Naira
NIO	Cordoba Oro
NOK	Norwegian Krone
NPR	Nepalese Rupee
NZD	New Zealand Dollar
OMR	Rial Omani
PAB	Balboa
PEN	Nuevo Sol
PGK	Kina

PHP	Philippine Peso
PKR	Pakistan Rupee
PLN	Zloty
PYG	Guarani
QAR	Qatari Rial
RON	Romanian New Leu
RSD	Serbian Dinar
RUB	Russian Ruble
RWF	Rwanda Franc
SAR	Saudi Riyal
SBD	Solomon Islands Dollar
SCR	Seychelles Rupee
SDG	Sudanese Pound
SEK	Swedish Krona
SGD	Singapore Dollar
SHP	Saint Helena Pound
SKK	Slovak Koruna
SLL	Leone
SOS	Somali Shilling
SRD	Surinam Dollar
STD	Dobra
SYP	Syrian Pound
SZL	Lilangeni
THB	Baht
TJS	Somoni
TMM	Manat
TND	Tunisian Dinar
TOP	Pa'anga
TRY	New Turkish Lira
TTD	Trinidad and Tobago Dollar
TWD	New Taiwan Dollar
TZS	Tanzanian Shilling
UAH	Hryvnia
UGX	Uganda Shilling
USD	US Dollar
USN	
USS	
UYU	Peso Uruguayo
UZS	Uzbekistan Som
VEB	Venezuelan bolívar
VND	Vietnamese đồng
VUV	Vatu

WST	Samoan Tala
XAF	CFA Franc BEAC
XAG	Silver (one Troy ounce)
XAU	Gold (one Troy ounce)
XBA	European Composite Unit (EURCO) (Bonds market unit)
XBB	European Monetary Unit (E.M.U.-6) (Bonds market unit)
XBC	European Unit of Account 9 (E.U.A.-9) (Bonds market unit)
XBD	European Unit of Account 17 (E.U.A.-17) (Bonds market unit)
XCD	East Caribbean Dollar
XDR	Special Drawing Rights
XFO	Gold franc (special settlement currency)
XFU	UIC franc (special settlement currency)
XOF	CFA Franc BCEAO
XPD	Palladium (one Troy ounce)
XPF	CFP franc
XPT	Platinum (one Troy ounce)
XTS	Code reserved for testing purposes
YER	Yemeni Rial
ZAR	South African Rand
ZMK	Kwacha
ZWD	Zimbabwe Dollar

11. Rollout Certification

This section will describe the rollout process of ePay5 integration with all the required steps needed from both SP and DSG:

11.1 Pre-Rollout Procedure

The following steps should be conducted by DSG/SP to verify that all requirements and prerequisites are completed/implemented to process the rollout:

- SP sent official email/letter requesting to integrate with ePayment Gateway
- DSG sent the Development guide and all required integration documents and components to SP
- DSG and SP signed Electronic Payment Agreement
- SP filled and shared the Project Plan with DSG
- SP filled and submitted the “**NewSP_ApplicationForm**”, “**Service_Subscription_Form_ePay**” and “**New_ePay_User_Form**” forms to DSG
- DSG received the forms
- DSG configures SP, service and users on ePay QA environment based on the shared forms
- DSG Shares details related to integration with SP
- SP provides SSL certificate to DSG. For SSL certificate, DSG is providing the below options to SP:
    - **Option 1:** To use a self-signed certificate (**Only for QA**)
    - **Option 2:** To use a trusted certificate authority such as Comtrust / Digicert etc.
    - **Option 3:** DSG has an internal CA and can issue certificates for the SP to be used for ePayment Gateway. SP has to follow the same issuing process (CSR file) with DSG. There will be no fees for this option.
- DSG team provision the service on QA using the SSL certificate shared by SP.
- SP configured ePay5 server certificate on QA environment
- SP implemented all integration requirements according to the integration documents/development guide

11.2 Rollout Procedure

The following steps should be conducted by DSG/SP to verify that SP is ready to be moved to ePay5 production environment:

- SP to list all of their services requires payment facility.
- SP to categories the services to be authenticated / anonymous.
- SP to fill ePay5 Data mapping sheet (user, service and beneficiary information) for each service to DSG
- SP to fill the development completion checklist and share with DSG.
- SP implemented Manual Reconciliation System or Automatic Reconciliation API and provided the details to DSG
- SP runs test(s) to verify the integration according to the test cases document and share the results with DSG.
- DSG verifies the test cases results submitted by SP
- DSG verifies that SP is providing user, service and beneficiary information to DSG
- DSG verifies that SP is confirming service delivery to ePayment gateway
- DSG verifies the Reconciliation
- DSG send a formal email to SP confirming that SP is certified and ready to be moved to ePay5 production environment

11.3 Post-Rollout Procedure

The following steps should be conducted by DSG/ SP to configure SP on ePay5 production environment:

- SP filled and submitted the “**NewSP_ApplicationForm**”, “**Service_Subscription_Form_ePay**” and “**New_ePay_User_Form**” forms to DSG
- DSG received the forms
- DSG created SP on ePay production environment based on the shared forms
- DSG created the service on ePay production environment based on the shared forms
- DSG created Admin user on ePay production environment based on the shared forms
- DSG shared SP, service and Admin users details to SP
- DSG provide IP addresses for production environment to SP to be allowed from their side and request SP to provide IP addresses to be allowed from DSG side (Departments not in GIN)
- SP allowed IP addresses provided by DSG (Departments not in GIN)
- DSG allowed IP addresses provided by SP (Departments not in GIN)
- SP provided the SSL certificate to DSG. For SSL certificate, DSG is providing the below options to SP:
    - **Option 1:** To use a trusted certificate authority such as Comtrust / Digicert etc.
    - **Option 2:** DSG has an internal CA and can issue certificates for the SP to be used for ePayment Gateway. SP has to follow the same issuing process (CSR file) with DSG. There will be no fees for this option.
- DSG provisioned the service on production using the provided SSL certificate and confirm to SP
- SP configured ePay5 production server certificate on production environment
- DSG monitored SP transactions after the integration
- DSG configured the Credit Card verification process after monitoring successful transactions
- DSG sends a formal email to SP confirming that SP has completed the migration of ePay5 on production environment and request SP to sign the “**Final Project Acceptance Form**”
- SP shared the signed and stamped “**Final Project Acceptance Form**”

12. SSL certificate

ePay5 payment API is using SSL certificate for authentication and authorization. DSG is providing following options to the departments:

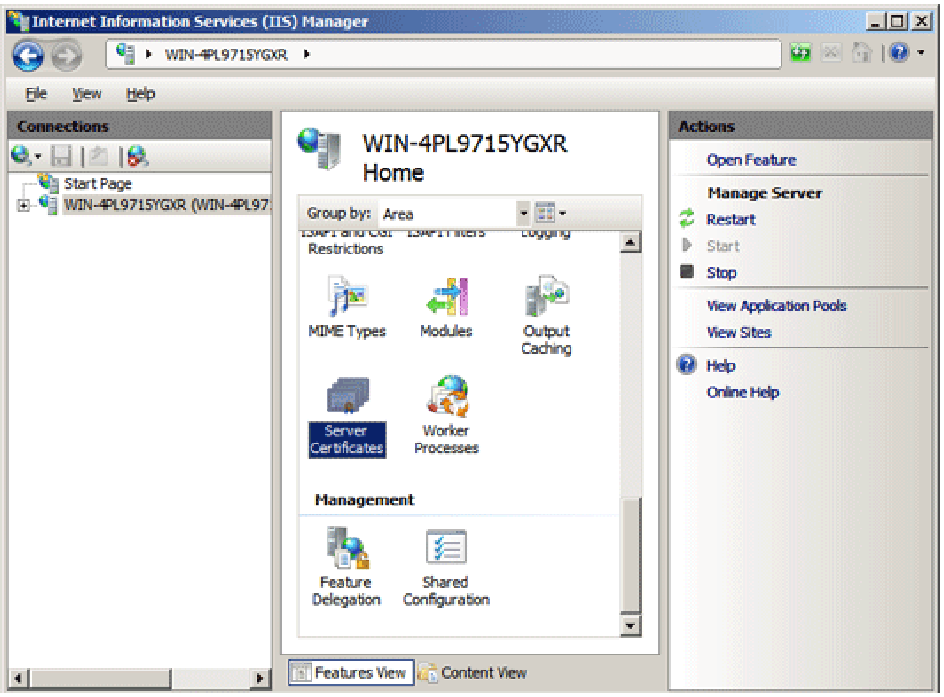
- Option 1: To use a self-signed certificate (Only for QA)
- Option 2: To use a trusted certificate authority such as Comtrust / Digicert etc.
- Option 3: DSG has an internal CA and can issue certificates for the government department to be used for ePayment Gateway. Department has to follow the same issuing process (CSR file) with DSG. There will be no fees for this option.

Department needs to consult their network and security team for the SSL certification option. In-case of option-3, department needs to follow the below steps

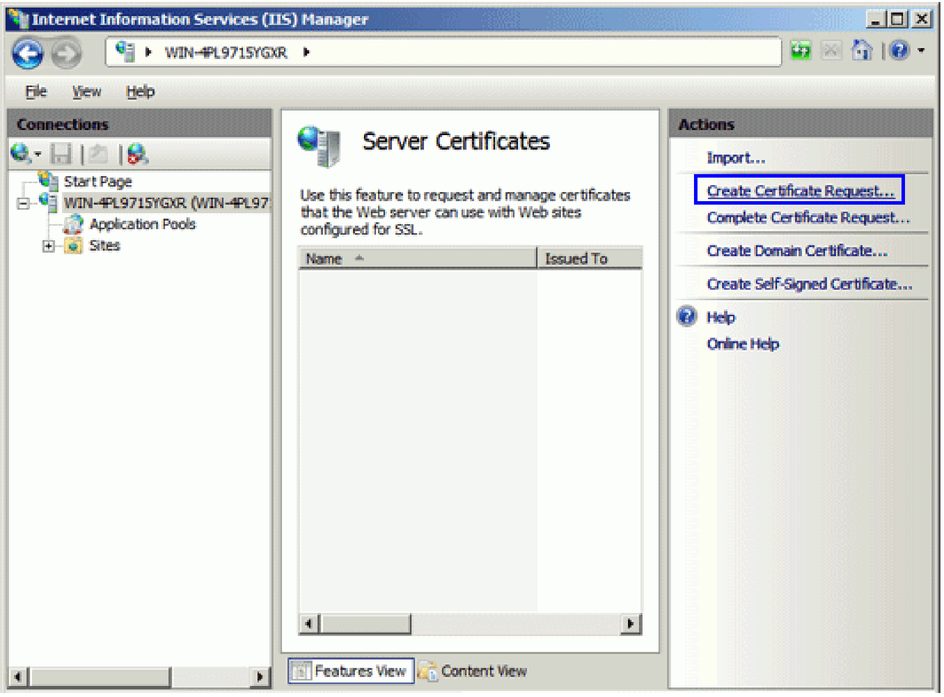
12.1 Certification Generation using IIS

Below are the steps for generating SSL certificate using IIS:

1. First the department needs to generate a CSR file. Below steps will generate a CSR file and private certificate:
   a. Click Start, then Control Panel, then Administrative Tools, then Internet Information Services (IIS) Manager
   b. Click on the server name
   c. Double-click on "Server Certificates" button in the "Security" section



d. Click on "Create Certificate Request" from the "Actions" menu


Internet Information Services (IIS) Manager
WIN-4PL9715YGXR
File
View
Help
Connections
Start Page
WIN-4PL9715YGXR (WIN-4PL97
Application Pools
Sites
Server Certificates
Name
Issued To
Features View
Content View
Actions
Import...
Create Certificate Request...
Complete Certificate Request...
Create Domain Certificate...
Create Self-Signed Certificate...
Help
Online Help

Use this feature to request and manage certificates that the Web server can use with Web sites configured for SSL.

e. In the "Distinguished Name Properties" window, enter the information as follows:

i. **Common Name** - The name through which the certificate will be accessed (usually the fully-qualified domain name, e.g., www.domain.com or mail.domain.com).

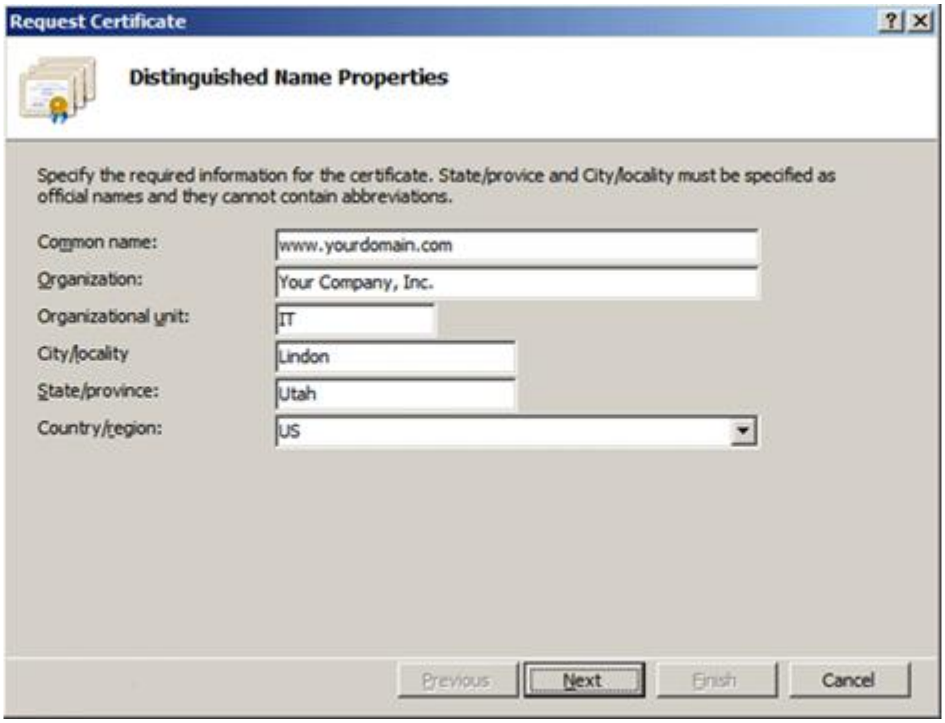
ii. **Organization** - The legally registered name of your organization/company.

iii. **Organizational unit** - The name of your department within the organization (frequently this entry will be listed as "IT," "Web Security," or is simply left blank).

iv. **City/locality** - The city in which your organization is located.

v. **State/province** - The state in which your organization is located.

vi. **Country/region** - If needed, you can find your two-digit country code in our list.



Distinguished Name Properties

Specify the required information for the certificate. State/provice and City/locality must be specified as official names and they cannot contain abbreviations.

Common name:	www.yourdomain.com
Organization:	Your Company, Inc.
Organizational unit:	IT
City/locality	Lindon
State/province:	Utah
Country/region:	US

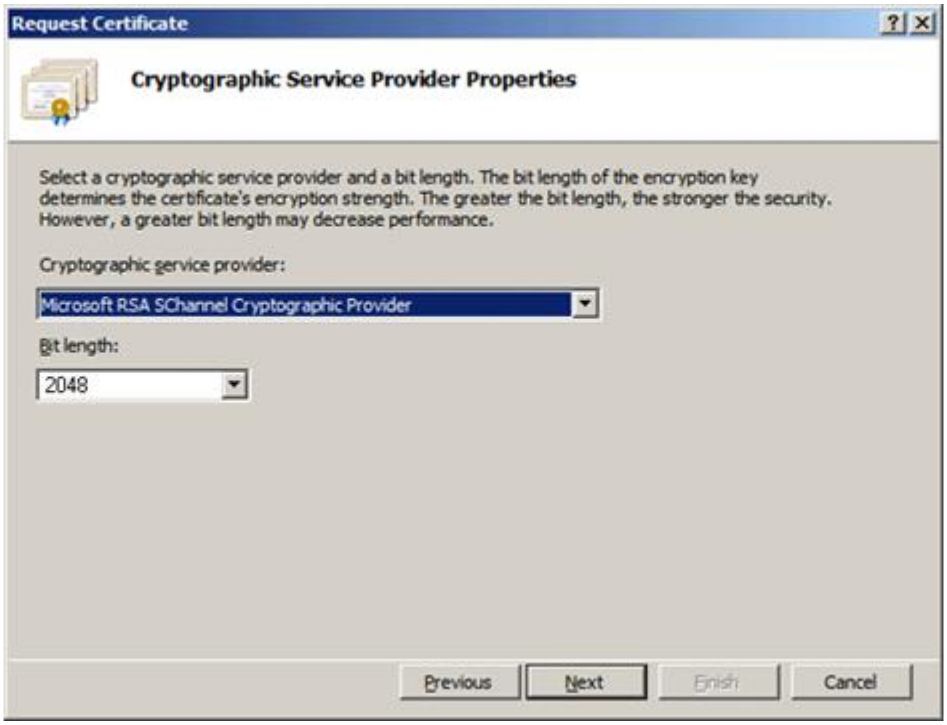
f. Click on Next button

g. In the "Cryptographic Service Provider Properties" window, enter the following information:

   i. Select “**Microsoft RSA SChannel Cryptographic Provider**” from Cryptographic service provider drop-down list, unless you have a specific cryptographic provider.

   ii. Select “**2048 (or higher)**” from Bit length drop-down list

h. Click on Next button


Request Certificate

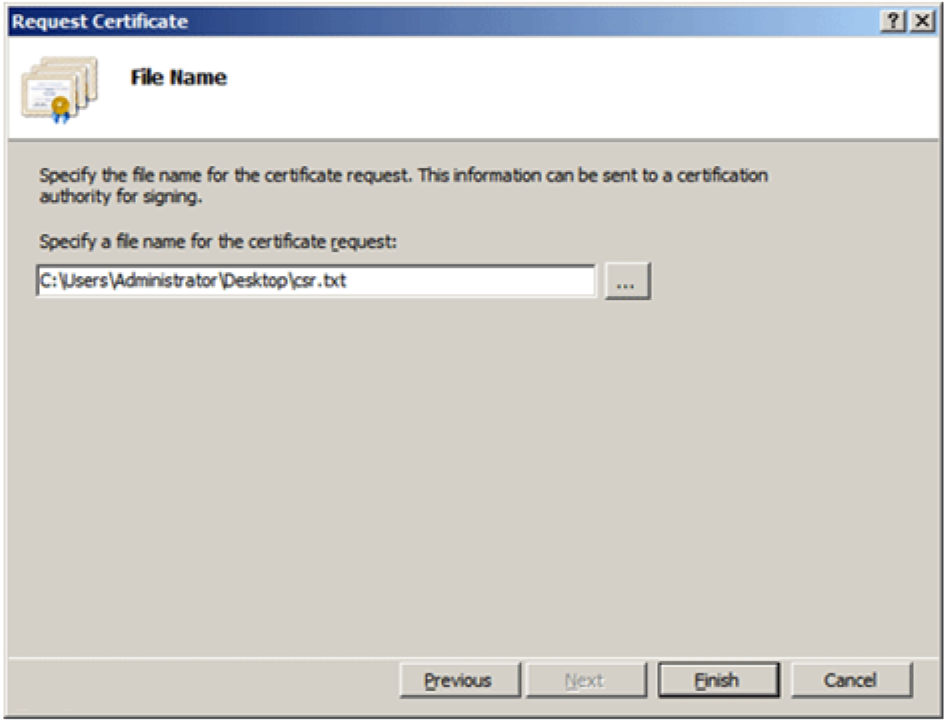
Cryptographic Service Provider Properties

Select a cryptographic service provider and a bit length. The bit length of the encryption key determines the certificate's encryption strength. The greater the bit length, the stronger the security. However, a greater bit length may decrease performance.

Cryptographic service provider: Microsoft RSA SChannel Cryptographic Provider

Bit length: 2048

i. Enter a filename for your CSR file


Request Certificate

File Name

Specify the file name for the certificate request. This information can be sent to a certification authority for signing.

Specify a file name for the certificate request: C:\Users\Administrator\Desktop\csr.txt

j. Click on “Finish” Button

2. CSR file has to be shared with DSG to issue the certificate.

3. DSG team will share the public certificate issued by DSG internal Certification Authority (CA).

12.2 Certification Installation using IIS

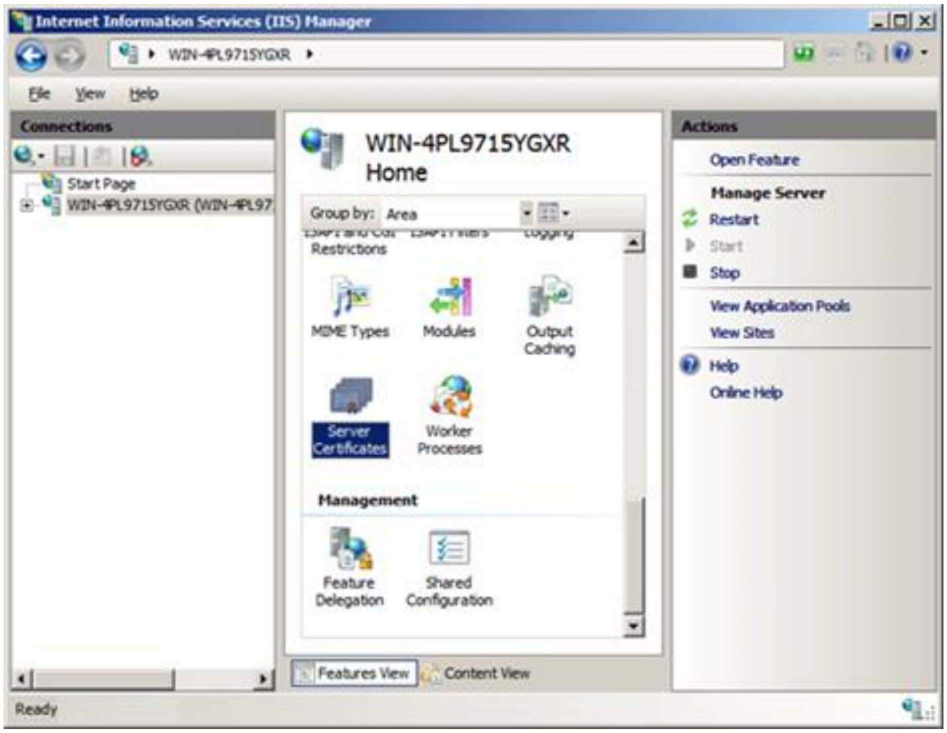
Below are the steps for Installing SSL certificate using IIS:

1. Import the certification authority reply to complete the certificate generation process. Below are the steps required to complete the process:

   a. Open the ZIP file that contains your SSL Certificate and save the SSL Certificate file (your_domain_name.cer) to the desktop of the web server that you are securing

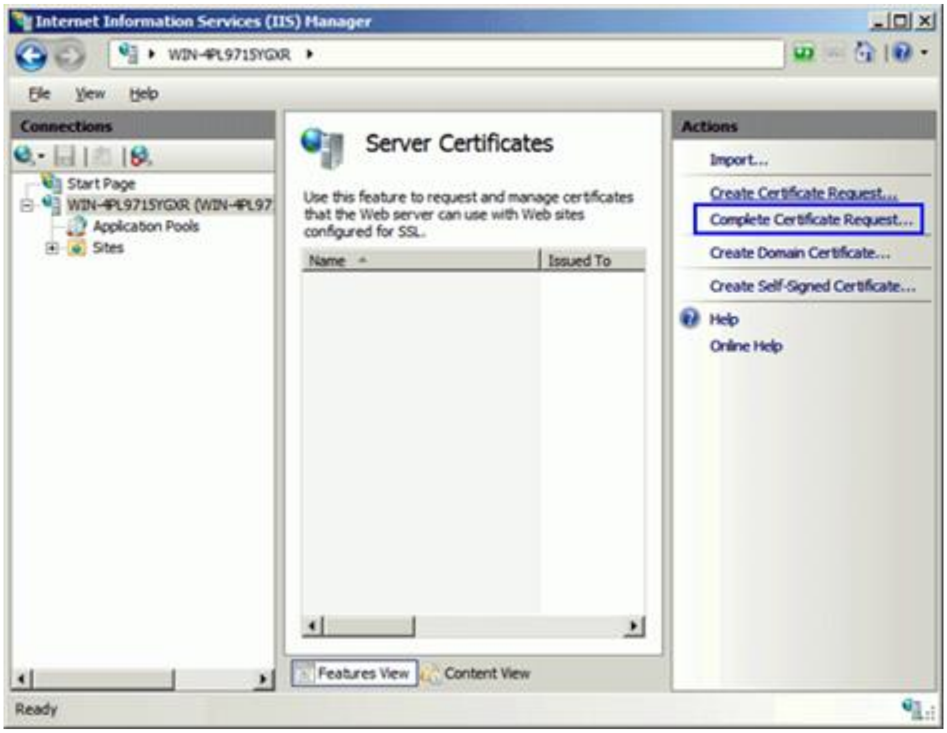
   b. Click Start, then Control Panel, then Administrative Tools, then Internet Information Services (IIS) Manager

   c. Select your server’s Hostname under " Connections"



   d. Double-click on "Server Certificates" button in the "IIS" section

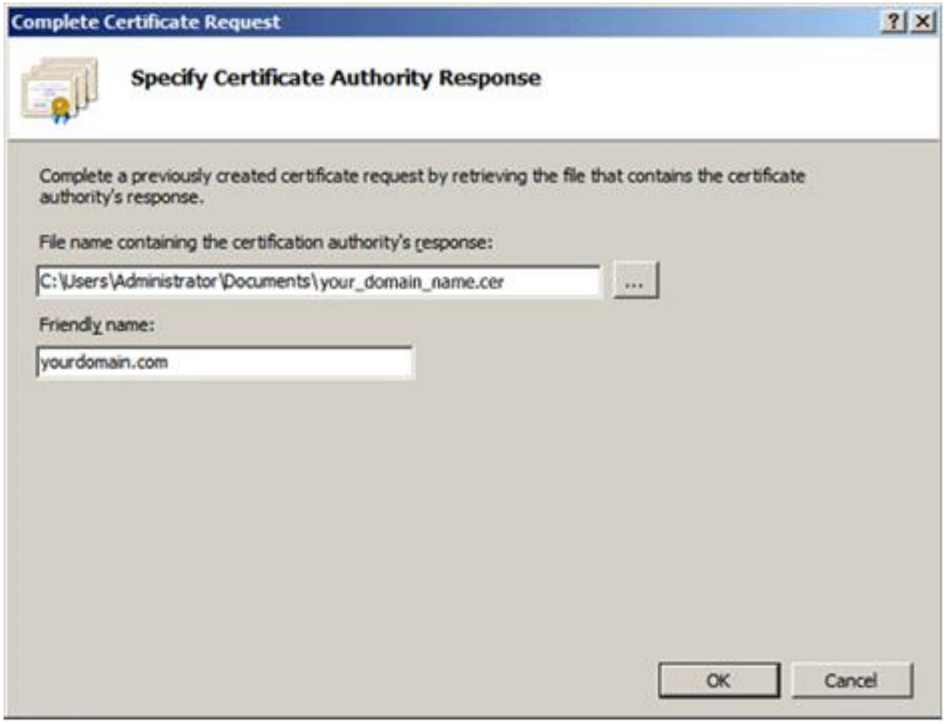
   e. Click on " Complete Certificate Request" from the "Actions" menu


Internet Information Services (IIS) Manager
WIN-4PL9715YGXR
File
View
Help
Connections
Start Page
WIN-4PL9715YGXR (WIN-4PL97
Application Pools
Sites
Server Certificates
Use this feature to request and manage certificates that the Web server can use with Web sites configured for SSL.
Name
Issued To
Features View
Content View
Actions
Import...
Create Certificate Request...
Complete Certificate Request...
Create Domain Certificate...
Create Self-Signed Certificate...
Help
Online Help
Ready

f. In the " Complete Certificate Request" window, Click on "…" button in File name containing the certification authority's response to browse the certificate file that DSG sent

g. Enter friendly name to identify the certificate in Friendly name field

h. Click on "OK" button to install the SSL Certificate to the server



Complete Certificate Request

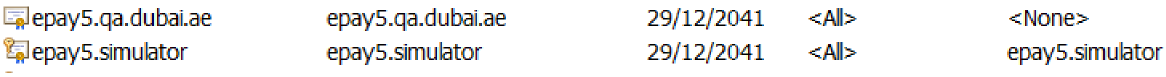
Specify Certificate Authority Response

Complete a previously created certificate request by retrieving the file that contains the certificate authority's response.

File name containing the certification authority's response: C:\Users\Administrator\Documents\your_domain_name.cer

Friendly name: yourdomain.com

2. You need to verify that a small lock will appear with the SSL certificate. This will show that both private key and public key are present with the certificate. Kindly check the below snapshot.



epay5.qa.dubai.ae	epay5.qa.dubai.ae	29/12/2041	<All>	<None>
epay5.simulator	epay5.simulator	29/12/2041	<All>	epay5.simulator

3. IIS user should be having rights to access the certificate.

12.3 Certification Generation using KeyTool

keytool provides a single step to create department private key save it in a key store, below are the set of arguments required for the current step.

Below are the steps for generating SSL certificate using Keytool:

1. Create a New Keystore

    a. You will be using the keytool command to create your new key-CSR pairing. Enter the following:

    ```
    keytool -genkey -alias server -keyalg RSA -keysize 2048 -keystore
    yourdomain.jks
    ```

    Note: 'Yourdomain' is the name of the domain you are securing. However, if you are ordering a Wildcard Certificate, do not include * in the beginning of the filename as this is not a valid filename character.

    b. You will be prompted for the DN information

    Note: When it asks for first and last name, this is not YOUR first and last name, but rather your domain name and extension (i.e., www.yourdomain.com). If you are ordering a Wildcard Certificate, this must begin with *. (Example: *.digicert.com)

    c. Confirm that the information is correct by entering 'y' or 'yes' when prompted.

    d. Next, you will be asked for your password to confirm.

    Note: Make sure to remember the password you choose.

2. Generate Your CSR with Your New keystore

    a. Use keytool to create the Certificate Signing Request. Enter the following:

    ```
    keytool -certreq -alias server -keyalg RSA -file yourdomain.csr -
    keystore yourdomain.jks
    ```

    Note: 'Yourdomain' is the name of the domain you are securing. However, if you are ordering a Wildcard Certificate, do not include * in the beginning of the filename as this is not a valid filename character.

    e. Enter the keystore password. Then the SSL Certificate CSR file is created.

    f. CSR file has to be shared with DSG to issue the certificate.

g. DSG team will share the public certificate issued by DSG internal Certification Authority (CA).

12.4 Certification Installation using KeyTool

Below are the steps for Installing SSL certificate to your Java Keystore using Keytool:

1. Save your SSL Certificate bundle file (your_certificate_name.cer) received from DSG

   Note: The certificate must be installed to the same keystore that was used to generate your CSR. You will get an error if you try to install it to a different keystore.

2. Type the following command to install the certificate file:

```
keytool -import -trustcacerts -alias server -file
your_certificate_name.cer -keystore your_site_name.jks
```

   Note: If the certificate is installed correctly, you will receive a message stating, **"Certificate reply was installed in keystore"**. If it asks, if you want to trust the certificate. Choose **y** or **yes**.Your keystore file (your_site_name.jks) is now ready to use on your server. Just configure your server to use it.

13. Production Verification

It is recommended to verify the production readiness before the go-live. This will help the department to verify the followings:

1. Network connectivity is working fine.
2. SSL certificates are configured properly.
3. ePay5 production configurations:

Production Configuration Test (.Net Simulator):

1. Configure the SSL certificates (Department and ePay5 production server certificate on the server).
2. Deploy the sample simulator in production site for testing the production configuration.
3. Configure the simulator for production environment.
   - ➢ Configure the production certificate in the web.config file.
     <!--server certificate has to provide by DSG-->

     <add key="ServerCertificate" value="epay5.dubai.ae"/>

     <!--below is the client certificate-->

     <add key="ClientCertificate" value="**Department Certificate Name** "/>

   - ➢ Configure the production ePay5 API URL in web.config file
     <endpoint address="https://epayment.dubai.ae/ePayHub/processRequestAPI"
     binding="customBinding" bindingConfiguration="PaymentAPIServicePortBinding"
     contract="echo.PaymentAPI" name="PaymentAPIServicePort">

   - ➢ Configure the production ePay5 API URL in web.config file
     <!--server certificate Alias for Prod-->

     <dns value="epay5.dubai.ae" />

4. Try to access the simulator, change the SPCODE, SERCODE for DTCM and click pay.
   SPCODE: **<Department SPCODE>**

   SERVCODE: **<Department SERVCODE>**

Test Result: System should be able to generate the token and redirect the user to ePay production. User should see the payment option page.

14. Go live Check List

Before go-live following has to be completed.

a. Integration certified in staging environment.
b. Reconciliation process is certified in staging environment.
c. DSG will send Move to production form to be signed by authorized person from Service Provider.
d. Agreement is signed with DSG.
e. DSG will configure and provide production details to the service provider.
f. Service Provider will provide the go-live date to DSG.