

UPC Secure Page

API Integration

Version 2.0

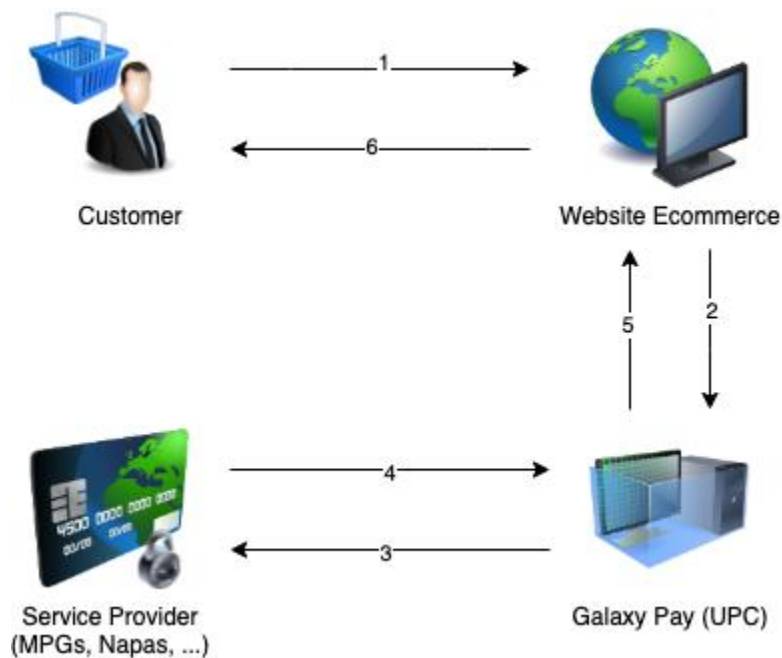
Version	Date	Author	Note
1.0	09-03-2022	Huỳnh Hữu Lộc	
2.0	10-03-2022	Huỳnh Hữu Lộc	Add Integration Flow and Demo section

Contents

1	Overview	2
1.1	Integration Flow.....	2
1.2	Demo.....	2
1.3	Message Structure	3
1.4	Integration Keys	3
1.5	Response Code Table	3
1.6	Encryption & Decryption Algorithm.....	4
1.7	Signature Algorithm	5
1.8	Card BIN Table.....	6
1.9	Card Test Information	8
1.9.1	International (via MPGs)	8
1.9.2	Viet Nam Local Banks (via Napas).....	8
2	API Specfication	9
2.1	Definition Table.....	9
2.2	Secure Page API.....	9
2.2.1	Request Message	9
2.2.2	Response Message	11
2.3	Cancel URL PostBack	12
2.3.1	Response Message	12
2.4	Result & IPN PostBack.....	13
2.4.1	Response Message	13

1 Overview

1.1 Integration Flow



1.2 Demo

Demo		Link
Live Demo		https://uat-merchant.galaxypay.vn:8000
Source Demo	.NET	https://gitlab.com/gpay-sample/upc/upc-dotnet
	React & Node JS	https://gitlab.com/gpay-sample/upc/gpay_upc_demo

1.3 Message Structure

- Service: REST API
- HTTP Method: POST
- Content Type: application/json

The request structure:

#	Component	Level	Param	Description
1	Request Header	L1	merchantKey	Key for authentication request. Provided by UPC .
3		L1	Accept	Accept encoding: application/json
4		L1	Content-Type	Request content type: application/json
5	Request Body	L1		Content of request

1.4 Integration Keys

Name	Value	Note
Endpoint	https://uat-payment.hdbank.com.vn:6969/transaction	Use for UAT only
SHA 256 Salt	GO@2021	Use for Signature
AES Key (HEX)	9d5b2dfde02abb9a49fc2bba7c26862ac44314f732ac97532f5055d877c49d68	Use for Decrypt data
Merchant Key	UPC will provided detail when integration	Use in Request Header

1.5 Response Code Table

Response Code	Description
00	Success
400	Bad Request
401	Invalid Merchant Key
499	Cancel by User

1.6 Encryption & Decryption Algorithm

- Use **AES 256** to decrypt response data that receive from UPC.

Key	Hex String (256)
Encoding	UTF8
Padding Mode	PKCS7
Cipher Mode	CBC
IV	00000000000000000000000000000000

Example:

Hex Key	9d5b2dfde02abb9a49fc2bba7c26862ac44314f732ac97532f5055d877c49d68
Encrypted Data	2ec9a9d930ae9874a903fcd984e8b56777b3dd6cb063bb4d4182b631c6c84f17d3491297013846a8331aa4a8c42a683c37b4d58ac4afdd16fc4c694dfc5546aa616961501af147bfe638762504553bd6a68798c4d70308ae955694a8e49c3fde1140939f240cd34f132c4270131ec8b6768356713f1c5bcff8fa3f8e950712839f38e11a4a4a574cf9d8c7a286e3743e297e1d3088cd6d87ac1ad59d16e816f195cbb03c1994ba6e610633a90d53fb6c47321627ccc176ad977182e6cac1377e1b43c2ea6447a4d80bccf529df82b2c2eb05c8c5416885af7cf797c3d7fd1d978383799cc8260256b304feeb0905ce89fb6b4ac9daa8cc875799d881f50283e10ef14afa8971b161af59eff9b3dab2338e86478e3b1344ee8739c1a7524d4a7c709574a72b50e1ea1d4be606faa272dd9e970100e0f2b9d47ffa14d69b621eee38205e64550edd8f711f23c732ca92b87af838b318ec2c4ee07163d60643e68afa8bcce544a29536556b3a638904ad1e1fdd11445001daa4a147a4006f583070bfa6f7ca330c59fe8036a3b318945347fa1b2a1ffff89b3a78607624eedf5e9a72af807bac5e63f2e727a6e466e4965c8f1dcd503b24d512729eb24a0985cf80aebfa310fc25ceee8e5f92f33222775b24cbf37b4a4e3b285ce0cebccc18140
Decrypt Data	{"responseCode":"00","responseMessage":"SUCCESS","pay_timestamp":1646208061,"pay_amount":"12001","language":"en","orderId":"e371f9dc-9139-48c4-9188-e273b553b597","billNumber":"206153040447","orderAmount":"12001","orderDescription":"this is order description","orderCurrency":"VND","cardType":"atm","cardScheme":"970437","token":"","order_timestamp":1646208020866,"client_ip_addr":"1.1.1.1","uphParameters":{"ip":"14.241.237.122","agent":"Safari 15.3","device":"Mac"}}

1.7 Signature Algorithm

- Flatten json objects into a single-depth Object.
- Sort the json keys with alphabetical order to create the sign data by concat the json string values.
- Apply hash algorithm **SHA256** with **Salt**.

Signature = SHA256(SignData + Salt)

Example:

Salt	GO@2021
JSON Data	<pre>{ "billNumber": 205249008064, "orderAmount": 100000, "orderCurrency": "VND", "orderDescription": "DEMO TRANSACTION", "cardType": "atm", "bank": "970437", "otp": "on", "request": "purchase", "token": "", "language": "vi", "client_ip_addr": "1.1.1.1", "order_timestamp": 1646797594, "uphParameters": { "ip": "1.1.1.1", "agent": "Chrome 96.0", "device": "Windows" } }</pre>
Sign Data	970437205249008064atm1.1.1.1vi100000VNDDEMO TRANSACTION1646797594onpurchaseChrome 96.0Windows1.1.1.1
Signature	dd3c7de9428c7c51a3797b4a9494455849ac0b5376c1ce01308fc996d95654fa

1.8 Card BIN Table

Viet Nam Local Banks	BIN
Ngân hàng TMCP Á Châu	970416
Ngân hàng TMCP Ngoại thương Việt Nam	970436
Ngân hàng Wooribank	970457
Ngân hàng TMCP Bưu Điện Liên Việt	970449
Ngân hàng TNHH Indovina	970434
Ngân hàng TMCP Nam Á	970428
Ngân hàng Liên Doanh Việt Nga	970421
Ngân hàng TMCP Đại Chúng Việt Nam	970412
Ngân hàng TMCP Sài Gòn Thương Tín	970403
Ngân hàng Nông Nghiệp và Phát triển Nông Thôn Việt Nam	970405
Ngân hàng TMCP Sài Gòn Công Thương	970400
Ngân hàng TMCP Xăng Dầu Petrolimex	970430
Ngân hàng TMCP Kiên Long	970452
Ngân hàng TMCP Bản Việt	970454
Ngân hàng TMCP Việt Nam Thương Tín	970433
Ngân hàng TNHH MTV Shinhan Việt Nam	970424
Ngân hàng TMCP Quốc Tế	970441
Ngân hàng TMCP Đông Á	970406
Ngân hàng TMCP Sài Gòn - Hà Nội	970443

Ngân hàng TMCP Đầu tư và Phát triển Việt Nam	970418
Ngân hàng TMCP Quốc Dân	970419
Ngân hàng TMCP Việt Á	970427
Ngân hàng TMCP Hàng Hải Việt Nam	970426
Ngân hàng TMCP An Bình	970425
Ngân hàng TMCP Phương Đông	970448
Ngân hàng TMCP Sài Gòn	970429
Ngân hàng TMCP Đông Nam Á	970440
Ngân hàng TMCP Xuất nhập khẩu Việt Nam	970431
Ngân hàng TMCP Công Thương Việt Nam	970415
Ngân hàng TNHH MTV Public Việt Nam	970439
Ngân hàng TMCP Việt Nam Thịnh Vượng	970432
Ngân hàng TMCP Quân Đội	970422
Ngân hàng TMCP Bảo Việt	970438
Ngân hàng TMCP Đại Dương	970414
Ngân hàng TNHH MTV Hongleong Việt Nam	970442
Ngân hàng TMCP Kỹ thương Việt Nam	970407
Ngân hàng TM TNHH MTV Dầu Khí Toàn Cầu	970408
Ngân hàng TMCP Phát Triển Thành Phố Hồ Chí Minh	970437
Ngân hàng TMCP Tiên Phong	970423

1.9 Card Test Information

1.9.1 International (via MPGs)

Network	Card Number	Expiry Date	CVV
Mastercard	5123450000000008	01 / 39	100
	2223000000000007	01 / 39	100
	5111111111111118	01 / 39	100
	2223000000000023	01 / 39	100
Visa	4508750015741019	01 / 39	100
	4012000033330026	01 / 39	100

1.9.2 Viet Nam Local Banks (via Napas)

Card Number	Card Holder Name	Issue Date	OTP
9704000000000018	Nguyen Van A	03/07	otp
9704020000000016	Nguyen Van A	03/07	otp

2 API Specification

2.1 Definition Table

Definition	Remark
Mandatory (M)	Required when call API
Optional (O)	DOT NOT Required when call API
Condition (C)	Base on Condition of another field when call API, that field can be Required or Optional

2.2 Secure Page API

Use to create a payment request to UPC Secure Page.

2.2.1 Request Message

Level	Field Name	Data Type	Required	Description
1	billNumber	String	M	Merchant Bill number or Transaction ID
1	orderAmount	Number	M	Transaction Amount
1	orderCurrency	String	M	VND or USD
1	orderDescription	String	M	Transaction Description
1	order_timestamp	String	M	Transaction time stamp (Unix time stamp)
1	cardType	String	M	atm or international atm: Napas Local Bank international: Visa and Master Card
1	bank	String	C	Send Local BIN (More detail in Card BIN Table) when cardType = atm 970437 : HD Bank 970403 : Sacombank ... Send VISA or MASTER When cardType = international
1	otp	String	O	Set default value: on
1	request	String	M	Set default value: purchase

1	client_ip_addr	String	O	Client IP address
1	language	String	M	vn or us vn: Vietnamese us: English
1	signature	String	M	View Signature Algorithm section for more detail

Example

```
{
  "billNumber": 205249008064,
  "orderAmount": 100000,
  "orderCurrency": "VND",
  "orderDescription": "DEMO TRANSACTION",
  "cardType": "atm",
  "bank": "970437",
  "otp": "on",
  "request": "purchase",
  "token": "",
  "language": "vi",
  "client_ip_addr": "1.1.1.1",
  "order_timestamp": 1646801369,
  "uphParameters": {
    "ip": "1.1.1.1",
    "agent": "Chrome 96.0",
    "device": "Windows"
  },
  "signature": "3142823eca766f6ab46011d20c3726c1f9496abc0c74c4cbd9509d9f3e10e76a"
}
```

2.2.2 Response Message

Level	Field Name	Data Type	Required	Description
1	orderId	String	M	UPC transaction ID
1	responseCode	String	M	See Response Code Table for detail
1	endpoint	String	M	Secure page URL (Use for Redirect to Secure page)
1	signature	String	M	Response signature

Example

```
{
  "orderId": "3fd93bd0-a64d-435b-91ee-27b051457304",
  "responseCode": "00",
  "endpoint": "https://uat-payment.hdbank.com.vn:6969/transaction/3fd93bd0-a64d-435b-91ee-27b051457304",
  "signature": "21cf195e51aaf6b5dcd2e9db4439b2053be90e2adafea3adbc8746f9dc9b15b7"
}
```

2.3 Cancel URL PostBack

Use for Merchant to redirect customer to cancel payment page.

2.3.1 Response Message

Level	Field Name	Data Type	Required	Description
1	orderId	String	M	UPC Transaction ID
1	data	String	M	Response data encrypted by AES 256
Example				
<pre>{ "orderId": "2a94fd42-0efd-428c-a872-54d5cf8f19b0", "data": "2ec9a9d930ae9874a903fcd984e8b5671a239e325e9cf4a163ab4b5a5fea8348555563d69f83c74f5974 d922d2d2a174ac103ff22f4f38be02f9979789ffcf1431f7532d77e94116bf5af3f3c2109d601f443f7f486c ba783b5a491fcee063122dfde90a43d4bc2d83631db072843ed6e79d9e48aed9c57dc2eb84b43b9468e 673bfaa8c3e3ce6cf7b77ba6901ad7c9c" }</pre>				
<p>Decrypted Data with AES 256</p> <pre>{ "responseCode": "499", "responseMessage": "Canceled by user", "uphParameters": { "ip": "1.1.1.1", "agent": "Chrome 96.0", "device": "Windows" } }</pre>				

2.4 Result & IPN PostBack

- Result URL: Use for Merchant to show result for Customer
- IPN URL: Use for Merchant to receive payment result and process business workflow.

2.4.1 Response Message

Level	Field Name	Data Type	Required	Description
1	orderId	String	M	UPC Transaction ID
1	data	String	M	Response data encrypted by AES 256
1	signature	String	M	Response signature
Example				
<pre>{ "orderId": "2a94fd42-0efd-428c-a872-54d5cf8f19b0", "data": "2ec9a9d930ae9874a903fcd984e8b5675f194086fc505e017500eef5989e80a60d6455964b2ab118150 23694b18fff9536b8ae2ad85d407ca198040d394fd4fd30dbb0b7f1394e8eb4ad88cd0cd5d6883f77313 dc3284f3fb064047e793ed3764500052e845a5fde18465de036766345b9b231824554c75fe0f6dd9a9ee a1986b557ce60108874596e82b412096b33977cdc01e6821d0fff3e6e85f119f915556e10867363b7f37 a19054364b2ae05a314fd72f7be21a9313e76cdeb94c78feafe89859ce94377cde8f808be261cf93a8e180 7fce347ad5f722062f8d3318db15d410a758c8b4b91c4e23e6b625a293deadf46254e3e31e7f7e0d942e eff8bc1ef01a45f045671b5679b066e9476ea54ce388a923129ed886ce01e25119fdc15721574017da06 ed58a5fff0b5d0370a833a75df51c89eebd6d168688a7624d015f080ce33cde08c50aa1b8f773686be71 b6f24ac80a130825cebe2bbc6ca3d427e4c5ac41d2e7cd9ee32832fee080bbc0979e564a7e803eed342d fbd16de81939dd9d581c83ad1d32efb557b58694c09dba36a9f14634a2d77015d21eff2d6e452927d33f ee41bdac80866da1d71b33ee083440a5df5ecdc33ea23f3e554181e2c560404d8afbbf879a4655df8d1e 72f7", "signature": "425e1ae7daa783fef1afc8b25af417c8b7a6f07cc8e5b24de912a1f37f1a63cc" }</pre>				
<p>Decrypted Data with AES 256</p> <pre>{ "responseCode": "117", "responseMessage": "Merchant is not found", "pay_timestamp": 1646802548, "pay_amount": 10969, "language": "vi", "orderId": "2a94fd42-0efd-428c-a872-54d5cf8f19b0", "billNumber": "18804", "orderAmount": 10969, "orderDescription": "test paymnet", "orderCurrency": "VND", "cardType": "atm",</pre>				

```
"cardScheme": "970437",  
"token": "",  
"order_timestamp": 1646802413,  
"client_ip_addr": "1.1.1.1",  
"uphParameters": {  
  "ip": "1.1.1.1",  
  "agent": "Chrome 96.0",  
  "device": "Windows"  
}  
}
```

GALAXY PAY