FORT Mobile SDK for iOS

Merchant Integration Guide

Document Version: 3.0

May, 2018

Contact Us

integration@payfort.com
www.payfort.com

Contents

1 PAYFORT

PAYFORT is a trusted online payment gateway enabling businesses, governments, SMEs, startups and institutions with innovative payment options for both the banked and non-banked online shoppers.

We work with our customers first by understanding both their financial and revenue model; identify areas of risk exposure, and payment processes in order to formulate strategies to maximize online payment acceptance. We work under the notion that “People are different” thus we help our Merchants in offering different payment options that mirror their online shoppers behavior for both credit card and non-credit cardholders.

Our team is comprised of seasoned bankers, technology gurus, and risk management experts that have been helping hundreds of firms manage and innovate their online payment processes across the Arab World and beyond.

2 About this Document

This document describes our FORT Mobile SDK (for iOS) and includes information on how to integrate it with the Merchant's Mobile Application.

2.1 Intended Audience

This document was created for the iOS Merchants' developers who will integrate the FORT Mobile SDK with their Merchants' Applications.

3 Before Starting the Integration with FORT

These are the steps you need to know; to start building an integration with PayFort:

Step 1: Access your test account
You need to make sure that you have access to the test account, it's a full test environment allow you to simulate and process simulation transactions.

Step 2: make sure that you are using the correct integration type
Prior building the integration, you need to make sure that you are selecting and using the proper parameters in the API calls as per the required integration type.
All the mandatory parameters mentioned under every section in the API document

Step 3: Create the Transaction Request
Process the valid API request depends on transaction parameters included, you need to check the documentation and read every parameter possible values in order to reduce the errors in processing the transaction.

Step 4: Process the Transaction Response
After every payment, PayFort return the transaction response on the URL configured in your account under Technical Settings channel configuration.

For more details; check the Direct Transaction Feedback section.

You need to validate the response parameters returned on this URL by calculating the signature for the response parameters using the SHA Response Phrase configured in your account under Security Settings.

Step 5: Test and Go Live
You can use our testing cards to test your integration and simulate your test cases. PayFort team may require to test your integration before the going live to assure your application integration.

4 About the Software

4.1 Supported Platforms

IOS 8+

4.2 Localization

The FORT Mobile SDK supports both English and Arabic languages.

4.3 Screen Orientation

Portrait is the only orientation supported within the FORT Mobile SDK.

4.4 Supported Payment Methods

Through the first version of the FORT Mobile SDK, the Merchant has the ability to process a **CREDIT CARD** transactions only.

4.5 Supported Payment Options

The supported credit card payment options are **VISA**, **MASTERCARD**, **American Express (AMEX) and MADA.**

5 FORT Mobile SDK

The FORT Mobile SDK allows Merchants to securely integrate the payment functions. It also allows Merchants to easily accept In-App payments. Instead of the traditional, time-consuming, and complex way of being redirected to the mobile browser to complete the payment, In-App payments can be completed through our FORT Mobile SDK. In turn, this gives the Merchants' consumers a smooth, pleasing user-experience by using In-App payment functions through the native applications.

5.1 Download the FORT Mobile SDK

To download the FORT iOS Mobile SDK, click here.

5.2 Create FORT Mobile SDK Token

A Mobile SDK token is required to authenticate every request sent to the SDK. The token is also significant to process payment operations in the FORT through our FORT Mobile SDK.

NOTE!

- A unique token should be created for each transaction. Each token has a life-time of only one hour if no new request from the same device is sent.

- The creation and initiation of a Mobile SDK token happens on the Merchant's server side.

5.3 FORT Mobile SDK Token URLs

Test Environment URL
https://sbpaymentservices.payfort.com/FortAPI/paymentApi

Production Environment URL
https://paymentservices.payfort.com/FortAPI/paymentApi

5.4 Parameters Submission Type

REST POST request using JSON.

5.4.1 FORT Mobile SDK Token Request Parameters

Request Parameters							
Parameter Name	Type	Mandatory	Description	Length	Special Characters	Possible/ Expected Values	Example
service_command	Alpha	Yes	Command	20	–	SDK_TOKEN	

access_code	Alphanumeric	Yes	Access code.	20			zx0IPmPy5jp1vAz8Kpg7
merchant_identifier	Alphanumeric	Yes	The ID of the Merchant.	20			CycHZxVj
language	Alpha	Yes	The checkout page and messages language.	2		- en - ar	
device_id	Alphanumeric	Yes	A unique device identifier.	100	-		ffffffff-a9fa-0b44-7b27-29e70033c587
signature	Alphanumeric	Yes	A string hashed using the Secure Hash Algorithm. (More details are available in our PayFort Merchant Integration Guide).	200			7cad05f0212ed933c9a5d5dffa31661acf2c827a

NOTE!

device_id - This value to be generated from the UIDevice Class Reference, and you can generate this parameter as the following:

```
[payFort getUDID];
```

5.4.2 FORT Mobile SDK Token Response Parameters

Response Parameters						
Parameter Name	**Type**	**Mandatory**	**Description**	**Length**	**Possible/ Expected Values**	**Example**
service_command	Alpha	Yes	Command.	20	SDK_TOKEN	
access_code	Alphanumeric	Yes	Access code.	20		zx0IPmPy5jp1vAz8Kpg7

merchant_identifier	Alphanumeric	Yes	The ID of the Merchant.	20		CycHZxVj
language	Alpha	Yes	The checkout page and messages language.	2	- en - ar	
device_id	Alphanumeric	Yes	The ID of the used device for this payment.	100		ffffffff-a9fa-0b44-7b27-29e70033c587
sdk_token	Alphanumeric	Yes	An SDK token to enable using the FORT Mobile SDK.	100		Dwp78q3
signature	Alphanumeric	Yes	A string hashed using the Secure Hash Algorithm. (More details are available in our PayFort Merchant Integration Guide).	200		7cad05f0212ed933c9a5d5dffa31661acf2c827a
status	Numeric	No	A two-digit numeric value that indicates the status of the transaction.	2	(Please refer to section Statuses).	
response_code	Numeric	No	Response Code carries the value of our system's response. *The code is made up of five digits, the first 2 digits refer to the statuses, and the last 3 digits refer to the messages.	5		20064
response_message	Alphanumeric	No	Message description of the response code. It returns according to the request language.	150		Insufficient Funds

NOTE!

Every parameter the Merchant sends in the Request should be received by the Merchant in the Response - even the optional ones.

6 Integrate the FORT Mobile SDK

To process a transaction using the FORT Mobile SDK, create a Mobile SDK token (Please refer to section Create FORT Mobile SDK Token) and proceed through the following sections.

6.1 Using the FORT Mobile SDK

6.1.1 Payment Process

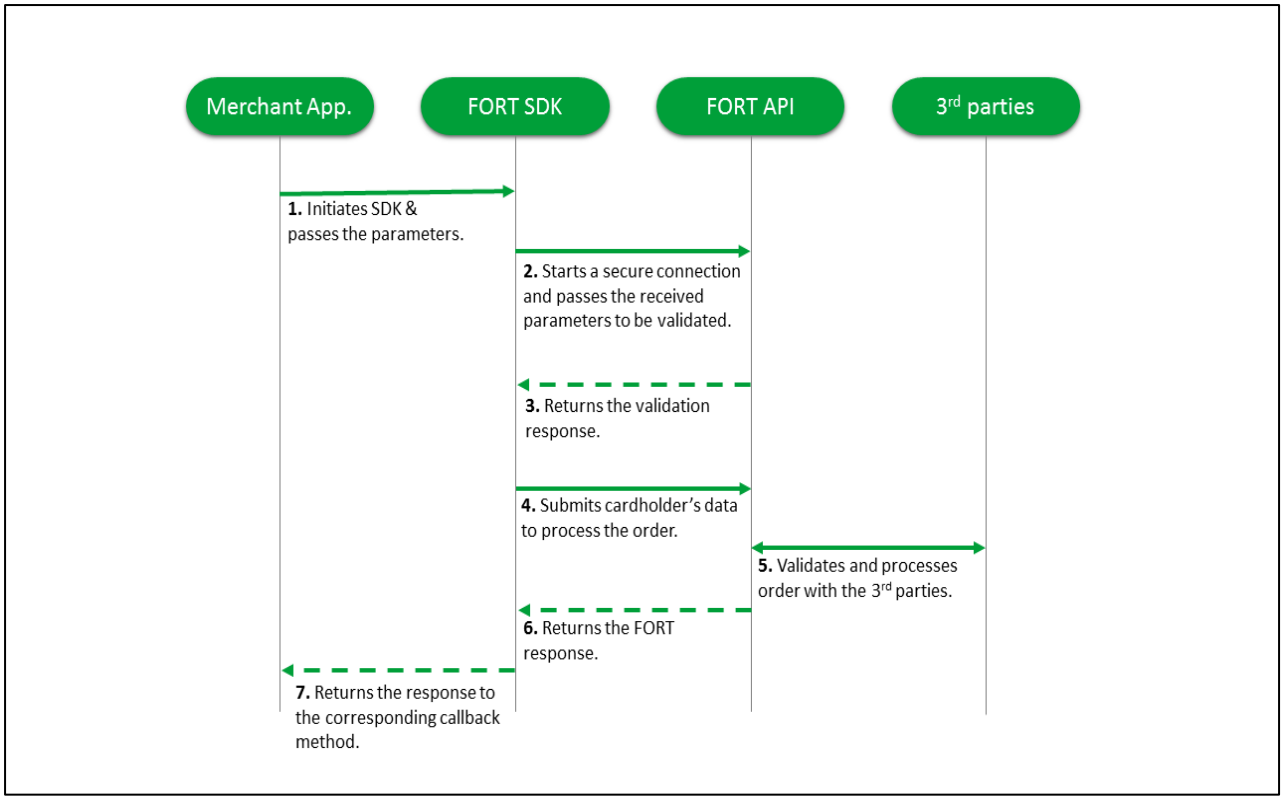


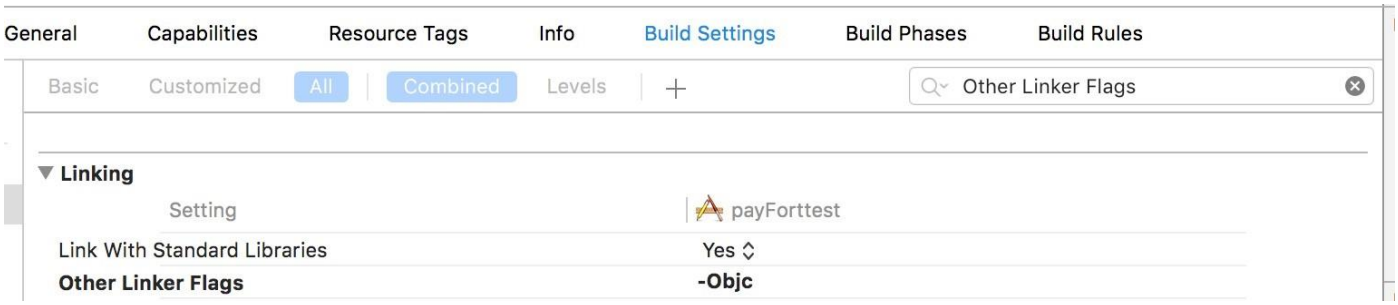
Figure 1: Payment Workflow

Workflow Description:

1. The Merchant's application initiates the FORT Mobile SDK and passes the parameters to the FORT Mobile SDK.
2. The FORT Mobile SDK starts a secure connection and passes the received parameters to the FORT API to be validated.
3. The FORT API returns the validation response.
4. The FORT Mobile SDK submits the cardholder's data to the FORT API to process the order.
5. The FORT API validates and processes the order with the third parties.
6. The FORT API returns the FORT response.
7. The FORT Mobile SDK returns the response to the corresponding callback method.

6.2 Include the SDK to your Xcode Project

- Extract the folder found in section 5.1
- Drag the PayFortSDK.framework & PayFortSDK.bundle to Frameworks in Project Navigator.
- Create a new group Frameworks if it does not exist.
    - Choose Create groups for any added folders.
    - Make Sure to select Copy files if needed.



- Set -ObjC in the Other Linker Flags in the Target → Build Settings Tab.
- For Swift Projects Don't forget to add the:

  `#import <PayFortSDK/PayFortSDK.h>` to the `Bridging-Header.h`

NOTE!
Ensure linked once in the Linked Framework and Libraries or just drag the PayFortSDK.framework to Embedded Binaries in the general tab in the project settings.

NOTE!
In Xcode, secondary-click your project's .plist file and select Open As → Source Code. Insert the following XML snippet into the body of your file just before the final, same as below:

```
</dict>element
<key>NSAppTransportSecurity</key>
<dict>
<key>NSAllowsArbitraryLoads</key><true/>
</dict>
```


NOTE!
To make the application not disconnected when go to background make sure to add this code:

- Objective C :

```
(void)applicationDidEnterBackground:(UIApplication *)application
{
__block UIBackgroundTaskIdentifier backgroundTask;
backgroundTask =
[application beginBackgroundTaskWithExpirationHandler: ^ {
[application endBackgroundTask:backgroundTask];
backgroundTask = UIBackgroundTaskInvalid; }];
}
```

- Swift :

```
func applicationDidEnterBackground(_ application: UIApplication)
{
var bgTask: UIBackgroundTaskIdentifier = 0
bgTask = application.beginBackgroundTask(expirationHandler: {
application.endBackgroundTask(bgTask)
bgTask = UIBackgroundTaskInvalid
})
}
```

6.3 Installation

1. Import the PayFort Library.

```
#import <PayFortSDK/PayFortSDK.h>
```

2. Initialize PayFortConrtoller with targeted environment, You set the target environment by setting one the two ENUM KPayFortEnviromentSandBox or KPayFortEnviromentProduction
   - Objective C

```
PayFortController *payFort = [[PayFortControlleralloc]initWithEnviroment:KPayFortEnviroment SandBox];
```

   - Swift

```
let payFort = PayFortController.init(enviroment: KPayFortEnviromentSandBox)
```

3. Set Dictionary contain all keys and values for SDK
   - Objective C

```
NSMutableDictionary *request = [[NSMutableDictionary alloc]init];
[request setValue:@"10000" forKey:@"amount"];
[request setValue:@"AUTHORIZATION" forKey:@"command"];
[request setValue:@"USD" forKey:@"currency"];
```



```
[request setValue:@ "email@domain.com" forKey:@"customer_email"];
[request setValue:@"en" forKey:@"language"];
[request setValue:@"112233682686" forKey:@"merchant_reference"];
[request setValue:`SDK TOKEN GOES HERE` forKey:@"sdk_token"];
[request setValue:@"" forKey:@"payment_option"];
[request setValue:@"gr66zzwW9" forKey:@“token_name"];
```

- Swift

```
let request = NSMutableDictionary.init()
request.setValue("1000", forKey: "amount")
request.setValue("AUTHORIZATION", forKey: "command")
request.setValue("USD", forKey: "currency")
request.setValue("email@domain.com", forKey: "customer_email")
request.setValue("en", forKey: "language")
 request.setValue("112233682686", forKey: "merchant_reference")
 request.setValue("token" , forKey: "sdk_token")
```

4. Call PayFort and response callback
   - Objective C

```
[payFort callPayFortWithRequest:request currentViewController:self
                        Success:^(NSDictionary *requestDic, NSDictionary *responeDic) {
                          NSLog(@"Success");
                          NSLog(@"responeDic=%@",responeDic);
                        }
                        Canceled:^(NSDictionary *requestDic, NSDictionary *responeDic) {
                          NSLog(@"Canceled");
                          NSLog(@"responeDic=%@",responeDic);
                        }
                        Faild:^(NSDictionary *requestDic, NSDictionary *responeDic, NSString *message) {
                          NSLog(@"Faild");
                          NSLog(@"responeDic=%@",responeDic);
                        }];
```

   - Swift

```
PayFort.callPayFort(withRequest: request, currentViewController: self,
                        success: { (requestDic, responeDic) in
                        print("success")
      },
```



```
                        canceled: { (requestDic, responeDic) in
                        print("canceled")
    },
                        faild: { (requestDic, responeDic, message) in
                        print("faild")
    })
```

6.4 SDK Response

By default the response will be dictionary to show the sent data in addition to the status, response message and response code.

The response will be ready in the registered call back handler with success, failed and cancelled. You can view the response by log the result as the followings:

- Objective

```
[payFort callPayFortWithRequest:request currentViewController:self
                        Success:^(NSDictionary *requestDic, NSDictionary *responeDic) {
                          NSLog(@"Success");
                          NSLog(@"requestDic=%@",requestDic);
                          NSLog(@"responeDic=%@",responeDic);
                        }
                          Canceled:^(NSDictionary *requestDic, NSDictionary *responeDic) {
                          NSLog(@"Canceled");
                          NSLog(@"requestDic=%@",requestDic);
                          NSLog(@"responeDic=%@",responeDic);
                        }
                         Faild:^(NSDictionary *requestDic, NSDictionary *responeDic, NSString *message) {
                          NSLog(@"Faild");
                          NSLog(@"requestDic=%@",requestDic);
                          NSLog(@"responeDic=%@",responeDic);
                          NSLog(@"message=%@",message);
                        }];
```

- Swift

```
PayFort.callPayFort(withRequest: request, currentViewController: self,
                        success: { (requestDic, responeDic) in
                          print("success")
                          print("responeDic=\(responeDic)")
                          print("responeDic=\(responeDic)")
    },
```



```
                    canceled: { (requestDic, responeDic) in
                      print("canceled")
                      print("requestDic=\(requestDic)")
                      print("responeDic=\(responeDic)")
    },
                    faild: { (requestDic, responeDic, message) in
                      print("faild")
                      print("requestDic=\(requestDic)")
                      print("responeDic=\(responeDic)")
                      print("message=\(message)")
    })
```

Also there is an option to show response view directly in elegant view that show response results either its success or failed. By activating the following option:

- Objective C

```
PayFort.IsShowResponsePage = YES;
```

- Swift

```
PayFort.IsShowResponsePage = true;
```

6.5 Hidden PayFort loading

There is an option to hide loading view when SDK initialize the connection request. By disable the following option:

- Objective C

```
PayFort.HideLoading = YES;
```

- Swift

```
PayFort.HideLoading = true;
```

6.6 Custom Payment Designing

You have the option to provide your custom UI theme for the payment view by the followings:

- Create your nibFile .xib and set the name of Arabic xib same name with English one with suffix -ar.
- Link the xib with PayFortView and bind all the IBOutlets in interface section

```
IBOutlet UILabel *titleLbl;
IBOutlet UIButton *BackBtn;
IBOutlet UILabel *PriceLbl;
IBOutlet JVFloatLabeledTextField *CardNameTxt;
IBOutlet JVFloatLabeledTextField *CardNumberTxt;
IBOutlet JVFloatLabeledTextField *CVCNumberTxt;
IBOutlet JVFloatLabeledTextField *ExpDateTxt;
```



```
IBOutlet UILabel *cardNumberErrorlbl;
IBOutlet UILabel *cVCNumberErrorlbl;
IBOutlet UILabel *expDateErrorlbl;
IBOutlet UISwitch *savedCardSwitch;
IBOutlet UIButton *paymentBtn;
IBOutlet UILabel *saveCardLbl;
IBOutlet UIImageView *imageCard;
```

- Assign new created xib file to PayFort Controller.

```
[payFort  setPayFortCustomViewNib:@"PayFortView2"];
```

NOTE!

- If you call Arabic view and the Arabic view not existed the application will crash.

- Don’t forget to set the custom view field in the identity inspector.

Customization example:

The following image is the standard design and layout of the Mobile SDK Payment page:

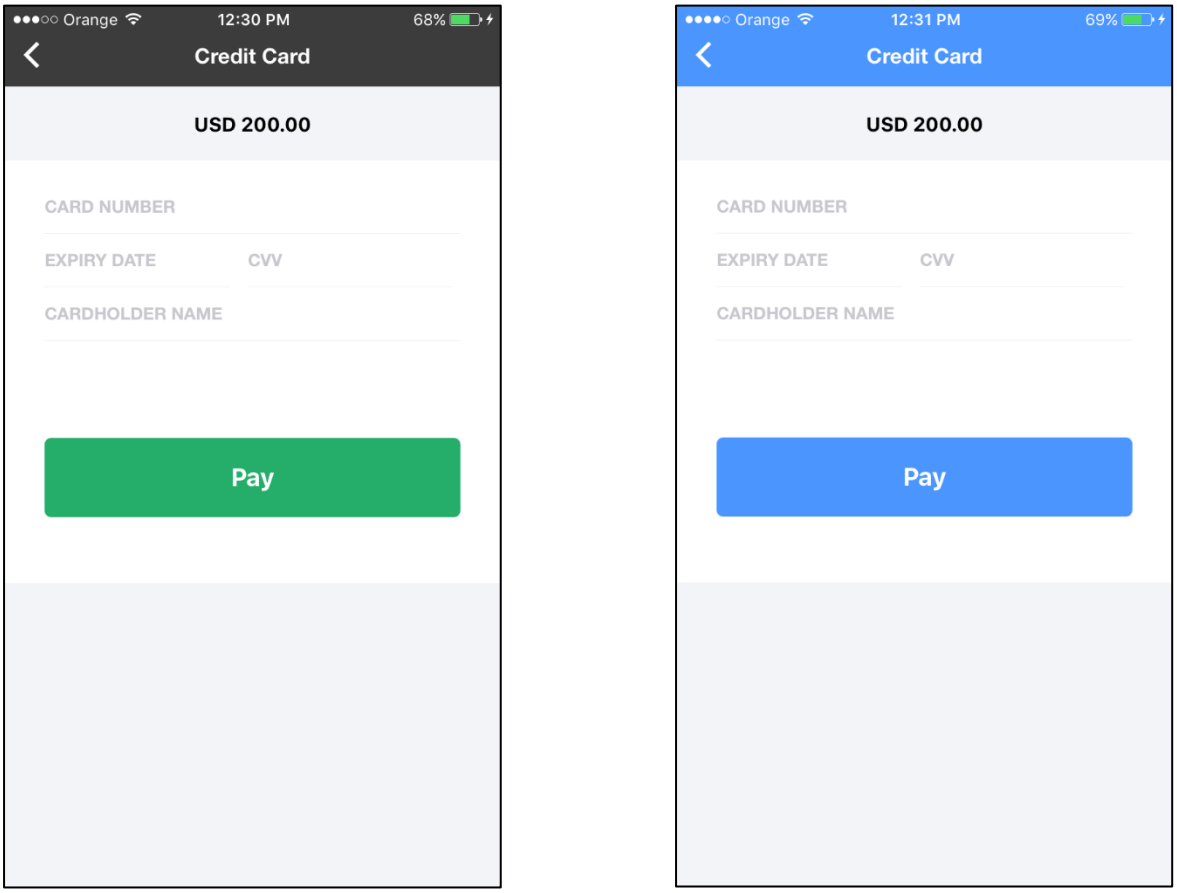


Figure 2: Standard vs. Customized Mobile SDK Payment Page Design

6.7 FORT Mobile SDK Operations

The FORT Mobile SDK allows the Merchant's application to process Authorization and Purchase operations.

6.7.1 Request Parameters

Request Parameters							
Parameter Name	**Type**	**Mandatory**	**Description**	**Length**	**Special Characters**	**Possible/ Expected Values**	**Example**
command	Alpha	Yes	Command.	20		- AUTHORIZATION - PURCHASE	
merchant_reference	Alphanumeric	Yes	The Merchant's unique order number.	40	- _ .		XYZ9239-yu898
amount	Numeric	Yes	The transaction's value. *Each currency has predefined	10			10000

			allowed decimal points that should be taken into consideration when sending the amount.				
currency	Alpha	Yes	The currency of the transaction's amount in ISO code 3.	3			AED
language	Alpha	Yes	The checkout page and messages language.	2		- en - ar	
customer_email	Alphanumeric	Yes	The customer's email.	254	_ - . @ +		customer@domain.com
sdk_token	Alphanumeric	Yes	An SDK token to enable using the FORT Mobile SDK.	100			Dwp78q3
payment_option	Alpha	No	Payment option.	10		- MASTERCARD - VISA - AMEX - MADA (for Purchase operations only)	
eci	Alpha	No	E-commerce indicator.	16		ECOMMERCE	
order_description	Alphanumeric	No	It holds the description of the order.	150	# ' / . _ - : $ **Space**		iPhone 6-S
customer_ip	Alphanumeric	No	It holds the customer's IP address. *It's Mandatory, if the fraud service is active.	45	.		192.178.1.10
customer_name	Alpha	No	The customer's name.	40	_ \		John Smith

					/ - . '		
phone_number	Alphanumeric	No	The customer's phone number.	19	+ - ( ) **Space**		009627972 19966
settlement_reference	Alphanumeric	No	The Merchant submits this value to the FORT. The value is then passed to the Acquiring bank and displayed to the merchant in the Acquirer settlement file.	34	- _ .		XYZ9239-yu898
merchant_extra	Alphanumeric	No	Extra data sent by merchant. Will be received and sent back as received. Will not be displayed in any report.	999	. ; / _ - , ' @		JohnSmith
merchant_extra1	Alphanumeric	No	Extra data sent by merchant. Will be received and sent back as received. Will not be displayed in any report.	250	. ; / _ - , ' @		JohnSmith
merchant_extra2	Alphanumeric	No	Extra data sent by merchant. Will be received and sent back as received. Will not be displayed in any report.	250	. ; / _ - , ' @		JohnSmith
merchant_extra3	Alphanumeric	No	Extra data sent by merchant. Will be received and	250	. ;		JohnSmith

			sent back as received. Will not be displayed in any report.		/ _ - , ' @		
merchant_extra4	Alphanumeric	No	Extra data sent by merchant. Will be received and sent back as received. Will not be displayed in any report.	250	. ; / _ - , ' @		JohnSmith

NOTE!

Before sending the amount value of any transaction, you have to multiply the value with the currency decimal code according to ISO code 3.

For example: If the amount value was 500 AED; according to ISO code 3, you should multiply the value with 100 (2 decimal points); so it will be sent in the request as 50000.

Another example: If the amount value was 100 JOD; according to ISO code 3, you should multiply the value with 1000 (3 decimal points); so it will be sent in the request as 100000.

6.7.2 Response Parameters

Response Parameters					
Parameter Name	**Type**	**Description**	**Length**	**Possible/ Expected Values**	**Example**
command	Alpha	Command.	20	- AUTHORIZATION - PURCHASE	
merchant_reference	Alphanumeric	The Merchant's unique order number.	40		XYZ2939-yu898
amount	Numeric	The transaction's value. *The amount parameter is returned by our system according to the predefined allowed decimal points per currency.	10		10000
currency	Alpha	The currency of the transaction's amount in ISO code 3.	3		AED

customer_email	Alphanumeric	The customer's email.	254		customer@domain.com
fort_id	Numeric	The order's unique reference returned by our system.	20		1443796866848
sdk_token	Alphanumeric	An SDK token to enable using the FORT Mobile SDK.	100		Dwp78q3
payment_option	Alpha	Payment option.	10	- MASTERCARD - VISA - AMEX - MADA (for Purchase operations only)	
eci	Alpha	E-commerce indicator.	16	ECOMMERCE	
authorization_code	Alphanumeric	The authorization code returned from the 3rd party.	100		P100000000000372136
order_description	Alphanumeric	It holds the description of the order.	150		iPhone 6-S
response_message	Alphanumeric	Message description of the response code. It returns according to the request language.	150		Insufficient Funds
response_code	Numeric	Response Code carries the value of our system's response. The code is made up of five digits.	5	(Please refer to section Messages).	
status	Numeric	A two-digit numeric value that indicates the status of the transaction.	2	(Please refer to section Statuses).	
customer_ip	Alphanumeric	It holds the customer's IP address.	45		192.178.1.10
expiry_date	Numeric	The card's expiry date.	4		1705
card_number	Numeric	The clear credit card's number.	16		400555******0001
customer_name	Alpha	The customer's name.	40		John Smith
phone_number	Alphanumeric	The customer’s phone number.	19		00962797219966
settlement_reference	Alphanumeric	The Merchant submits this value to the FORT. The value is then passed to the Acquiring bank and displayed to the merchant in the Acquirer settlement file.	34		XYZ9239-yu898
merchant_extra	Alphanumeric	Extra data sent by merchant. Will be received and sent back as received. Will not be displayed in any report.	999		JohnSmith
merchant_extra1	Alphanumeric	Extra data sent by merchant. Will be received and sent back as received. Will not be displayed in any report.	250		JohnSmith

merchant_extra2	Alphanumeric	Extra data sent by merchant. Will be received and sent back as received. Will not be displayed in any report.	250		JohnSmith
merchant_extra3	Alphanumeric	Extra data sent by merchant. Will be received and sent back as received. Will not be displayed in any report.	250		JohnSmith
merchant_extra4	Alphanumeric	Extra data sent by merchant. Will be received and sent back as received. Will not be displayed in any report.	250		JohnSmith

NOTE!

Every parameter the Merchant sends in the Request should be received by the Merchant in the Response - even the optional ones.

6.8 FORT Transaction Feedback

6.8.1 Overview

The FORT transaction Feedback system provides Merchants with two types of configurable notifications:

1. Direct Transaction Feedback, PayFort will send Merchants HTTPs notifications that inform Merchants of the transaction's final status whenever a transaction is processed.

2. Notification Transaction Feedback, PayFort will send Merchants HTTPs notifications that inform Merchants of the transaction's final status whenever a transaction status is updated.

6.8.2 Registering Transaction Feedback URLs

1. Log in to your back-office account.
2. Select the active channel under Integration Settings → Technical Settings.
3. Enter your Direct Transaction Feedback URL and Notification Transaction Feedback URL.
4. Click "Save Changes" button.

6.8.3 Transaction Feedback submission

The FORT will send Transaction Feedback data as form POST Parameters to the Merchant's Transaction Feedback URLs.

However if you want to change the submission type to JSON or XML, you can contact us on integration@payfort.com .

This configuration can be enabled by internal PayFort team only.

The specifics of the data will differ based upon the financial operation that has been processed.

Please refer to the FORT integration guide for more details.

6.8.4 Responding to FORT Transaction Feedback

Beyond whatever your Transaction Feedback URL does with the data received, it must also return a 2xx (like 200, 201, etc…) or 302 HTTP status code to tell the FORT that the notification was received. If your URL does not return 2xx or 302, the FORT will continue to retry the notification until it's properly acknowledged.

In case the FORT does not receive 200 or 302 HTTP status code it will attempt to send the notification for 10 times with 10 seconds in between.

This configuration is editable as well, if you want to change the grace period or the time interval between the retries please contact us on integration@payfort.com .

6.9 Sample Code

6.9.1 Initialize the Mobile SDK

- Objective C:

```
PayFortController *payFort = [[PayFortController
alloc]initWithEnviroment:KPayFortEnviromentSandBox];

//if you need to switch on the Payfort Response page
payFort.IsShowResponsePage = YES;

//Generate the request dictionary as follow
NSMutableDictionary *requestDictionary = [[NSMutableDictionary alloc]init];
[requestDictionary setValue:@"10000" forKey:@"amount"];
[requestDictionary setValue:@"AUTHORIZATION" forKey:@"command"];
[requestDictionary setValue:@"USD" forKey:@"currency"];
[requestDictionary setValue:@"email@domain.com" forKey:@"customer_email"];
[requestDictionary setValue:@"en" forKey:@"language"];
[requestDictionary setValue:@"112233682686" forKey:@"merchant_reference"];
[requestDictionary setValue:@"" forKey:@"payment_option"];
[requestDictionary setValue:@"gr66zzwW9" forKey:@"token_name"];

[payFort callPayFortWithRequest:requestDictionary currentViewController:self
Success:^(NSDictionary *requestDic, NSDictionary *responeDic) {

} Canceled:^(NSDictionary *requestDic, NSDictionary *responeDic) {


} Faild:^(NSDictionary *requestDic, NSDictionary *responeDic, NSString *message) {

}];
```

- Swift :

```
let payFort = PayFortController.init(enviroment: KPayFortEnviromentSandBox)

//if you need to switch on the Payfort Response page
```



```
paycontroller.isShowResponsePage = true


let request = NSMutableDictionary.init()
request.setValue("1000", forKey: "amount")
request.setValue("AUTHORIZATION", forKey: "command")
request.setValue("USD", forKey: "currency")
request.setValue("email@domain.com", forKey: "customer_email")
request.setValue("en", forKey: "language")
request.setValue("112233682686", forKey: "merchant_reference")
request.setValue("gr66zzwW9" , forKey: "token_name")
request.setValue("" , forKey: "payment_option")


payFort.callPayFort(withRequest: request, currentViewController: self,
success: { (requestDic, responeDic) in


},
canceled: { (requestDic, responeDic) in

},
faild: { (requestDic, responeDic, message) in

})
}
```

