

Direct Payment (Sadad)

This feature is for the advanced merchants and developers who wants to customize the checkout UI or wants to collect the card details on their interface and send it to Sadad. There are two ways to implement the direct payment method:

A. Using JS SDK and checkout form

1. On your checkout form, place the code below:

```
<script>
    var sadadGetChecksum = function(){

        $.ajax({
            type: "POST",
            url: 'checksumgenerate', //replace with
your URL
            data: $('#orderForm').serialize(), // Order
data you want to save to your website. Don't send card data.
            success: function (response) {

                afterChecksumSubmit(response);

            },
            error: function (err) {
                afterChecksumSubmit(err.statusText);
            }
        });
    };
</script>
<div id="sadad_cc_container" data-i-color="#531232" data-
cbfunc="sadadGetChecksum"></div>
<script src="https://sadadqa.com/jslib/sadad1.js"></script>
```

2. Replace checksumgenerate with your server side script url that will generate the checksumhash for the order data. Refer to the checksum_form.html file mentioned in the zip package on what the server side script should return as the result. The form id must be "sadadFinalForm".

3. Generate the checksumhash using `sadad.php` file as a reference. Replace the data with your merchant account and order specific values.
4. If your server is not in the PHP, you can use the checksum APIs to generate the checksumhash. The document to follow is `Checksum APIs_final.pdf`

Customization options:

- You can specify your own function instead of `sadadGetChecksum` as `"data-cbfunc"` attribute. Make sure it returns the form like `checksum_form.html`
- You can enter any valid color code for `data-i-color` attribute.
- You can define your custom css to customize button style, radio buttons style, input styles fonts etc.
- If you want to display Arabic labels and error messages you can use `"data-sd-lang"` attribute. Valid values for this attribute are `"ARB"` and `"ENG"`. Default is `"ENG"`

B. Using direct APIs

1. The merchant will design and develop his own interface and will collect the payment method and card details (in case of credit card) in his application UI.
2. The merchant will generate the checksum with the form data he's sending (without payment data like payment method, credit card details etc.). The card details should never be sent to merchant's server and should not be stored anywhere within the merchant's server/application.
3. Merchant will append the payment data including payment method and credit card (if payment method is credit card) and call the API with following details:

URL: <https://sadadqa.com/jslib/callapi.php>

Method: POST

Content-Type: application/x-www-form-urlencoded4

Data:

merchant_id=1234567&

ORDER_ID=9006&

WEBSITE=sadad.qa&
 TXN_AMOUNT=50.00&
[CUST_ID=example@example.com&](#)
[EMAIL=example@example.com&](#)
 MOBILE_NO=999999999&
 CALLBACK_URL=https://sadad.qa/callback.php&
 txnDate=2021-08-30+11:43:34&
 productdetail[0][order_id]=9006&
 productdetail[0][itemname]=Sample Product&
 productdetail[0][amount]=50&
 productdetail[0][quantity]=1&
 productdetail[0][type]=line_item&
 checksumhash=RsvMNLuSZd7AfgWKMd3egbOWaEPze1qE8diVKI05O3s/uuPIqXudDASXwtry3cUkYmW4EKNAaSQx9DWG5AWpchVd25onwDAf7a5s9xp8EnI=&
 card_details[payment_method]=2

Parameters that should not be included in the checksumhash generation:

Parameter Name	Description	Possible Values
card_details[payment_method]	The payment method, credit or debit	1 = Credit card 2 = Debit card
card_details[card-number]	Credit card number. Only numeric values. Should not include space. Not needed if payment method is debit card.	16 digit card number. Example: 4111111111111111
card_details[card-holders-name]	The credit card holder name. Alphanumeric and spaces allowed. Not needed if payment method is debit card.	Example: Sadad
card_details[expiry-month]	Card Expiry month prefix with zero if less than 10. Not needed	01,02,03,04,05,06, 07,08,09,10,11,12

	if payment method is debit card.	
card_details[expiry-year]	Card Expiry year. Last 2 digits only. Not needed if payment method is debit card.	For 2021, value must be 21. For 2022, value must be 22.
card_details[cvc]	CVV number. 3 digits for Visa and Mastercard. 4 Digits for American Express. Not needed if payment method is debit card.	Numeric value only.
card_details[cardType]	The card scheme type. Not needed if payment method is debit card.	Visa, Mastercard, Amex, JCB, Discover are possible values.
card_details[saveCard]	To save the card for future use. Token payments. Optional if not using the tokenization service or payment method is debit card.	1
card_details[user_ipAddress]	The IP address of the user.	Required parameter and a valid IP address of the customer must be passed.

Response:

```
{
  "status" : "success",
  "msg" : "<html>Dynamic form to be submitted.</html>"
}
```

Error Response:

```
{  
  "status" : "failed",  
  "error_message" : "The error message."  
}
```

4. If the response status is success, the msg field will contain a form to be submitted. The form's name property would be echoForm for the debit card payment method and it can be submitted via simple Javascript function like below:
`document.echoForm.submit();`

Change in the credit card processing for 3DS w.e.f. 5 Oct, 2021:

If the payment method is credit card, the msg field will contain an iframe with script. The developer must place the full content in the webpage/webview and the rest of the part will be handled accordingly.

The developer should bind the iFrame load event and monitor the URL to close/hide the iframe when it hits the callback URL.

5. If the response status is failed, there will be a field present "error_message" and the merchant can show that error to his customer.

Callback URL Processing:

- After the payment is completed by the user (success/failed), Sadad system will hit the callback URL with the following data as HTTP POST.

```
'website_ref_no' => "",  
'transaction_status' => "3",  
'transaction_number' => 'SD123456789322',  
'MID' => "1234567",  
'RESPCODE' => "1",  
'RESPMSG' => 'Txn Success',  
'ORDERID' => '123456',  
'STATUS' => 'TXN_SUCCESS',  
'TXNAMOUNT' => "5",  
'checksumhash' =>  
'z1DxSMDM0KFdDNj8MFn6v3INNbxdt9krll5FWPdIy7ZVrF06YuJmARUeD  
i43uIIHOjZkwAf+MJhFmUDA+LANKb0uWjxhdpf7sgFg5g0WsjM='
```

- The checksumhash must be verified using the checksumverify API to ensure the callback URL is called by Sadad system only.
- The field description for callback URL fields is given on <https://developer.sadad.qa> under web checkout section.