

## Tokenization

### **What is tokenization?**

It is a feature provided by the payment processors to save the credit card data in a secure vault and provide a randomly generated string (token) that can be used to recognize the card for future operations in scenarios where customer interaction is limited to none.

### **Where tokenization is useful?**

For the scenarios where customer uses your application/website on a frequent basis and a hassle free and smooth customer experience is required without the customer entering the credit card details every time there's a payment operation. Typical examples include food delivery applications, e-commerce websites/apps and subscription services.

### **Tokenization with Sadad**

To use the feature of tokenization with Sadad is easy, convenient and secure. This can be implemented in 5 easy steps.

1. Request Sadad administrator/ browse merchant panel add on section to get tokenization feature enabled.
2. After tokenization is enabled, if using direct payment, the parameter will be passed automatically if customer selects credit card as payment method and clicks save this card for future. If web checkout is used, and the customer selects credit card and ticks the box "save card", at the end of the transaction (successful only) the token will be generated and will be sent as a parameter in the callback response and merchant needs to save this token in his database securely.
3. For future payments, on a checkout screen payment method selection, merchant calls the API with token to get the masked card number and card type to display the saved card to the customer.
4. If a customer has multiple cards saved with the merchant, the customer will select one card. The merchant needs to develop the interface for the same.

5. The merchant will pass the customer's selected card token in a server to server request and if the token is valid, the transaction will be processed and merchant will get transaction success response along with transaction id.

## Base URLs

1. Sandbox mode: <https://api-sandbox.sadad.qa/api>
2. Live mode: <https://api-s.sadad.qa/api>

## Get the card details from the saved token

### 1. Get access token

**Path:** /userbusinesses/login

**Method:** POST

**Body:**

```
{
  "sadadId": "XXXXXXX",
  "secretKey": "XXXXXXXXXXXXXXXXX",
  "domain": "example.com"
}
```

**Response:** {

```
  "accessToken":
  "lBl6mCODvU0nCQtrYo6jTHe42UchbLLokUImgfcymNFNDJfm3pLBWeTYr
  xwJDLgW"
}
```

**Error:**

```
{"error":{"statusCode":400,"name":"Error","message":"Login Failed!"}}
```

**Parameters:**

sadadId – Sadad merchant ID. You can find this by login to merchant panel.

secretKey – Secret key under your merchant account > API section.

Domain - Any of your registered domain under API section.

## 2. Get the card details

**Path:** /usercards/getDetails?filter[token][0]=<Bank\_token>

(Use the base URL based on the mode in which the token was generated)

For Multiple tokens belong to the same customer.

usercards/getDetails?filter[token][0]=<Bank\_token1>&  
filter[token][1]=<Bank\_token2>

**Method:** GET

**Headers:**

Authorization: <Access token from step 1>

**Response:**

```
[{"cardnumber":"XXXXXXXXXXXX0008","cardType":"MASTERCARD","externaltoken":"Bank token"}]
```

**Error:**

```
{"error":{"statusCode":404,"name":"Error","message":"No data found!"}}
```

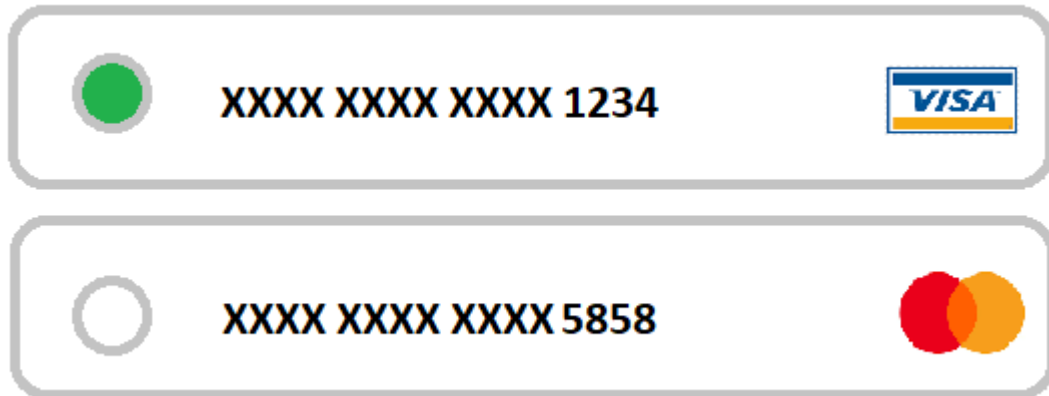
Or

```
[  
  {  
    "cardnumber": "XXXXXXXXXXXX0008",  
    "cardType": "MASTERCARD",  
    "externaltoken": "Bank token1"  
  },  
  {  
    "error": "Token is not valid or blocked",  
    "externaltoken": "Bank token2"  
  }  
]
```

## Create a payment transaction using the token

To create a transaction, the merchant will perform the following steps:

1. Get the card details for the logged in customer using the token saved for his credit card and display that in website/app interface.



2. If multiple tokens are saved for a customer, allow to select his desired credit card.
3. The customer selects the card and submits the order. Then merchant will do the following API call:

**URL:** <https://sadamqa.com/jslib/callapi.php>

**Method:** POST

**Body (form data):**

```
[merchant_id] => 6564499
[ORDER_ID] => 9006
[WEBSITE] => http://sadam.de
[TXN_AMOUNT] => 50.00
[CUST_ID] => example@example.com
[EMAIL] => example@example.com
[MOBILE_NO] => 999999999
[credit_card_token] =>
ovDZVQkDJLDPXnMmScOiA58ncauDK00Go7MD7aWAGD7sFSh6EBu3V0
[txnDate] => 2021-07-11 13:02:14
```



[checksumhash] =>  
T62Cs3z/Ox3LXNKymTXOwkEpDDnZHJW+rWqz7pwKjIlymodVw4udIbfozdfmXopIMk7oAt117Fz  
HN9FSyWd62uZiVba1XMj1XVfMeGgAcXA=

### CURL example:

```
curl -d '
merchant_id=6564499&ORDER_ID=9006&WEBSITE=http://sadamq.com&TXN_AMOUNT=50.00&CUS
T_ID=example@example.com&EMAIL=example@example.com&MOBILE_NO=999999999&credit_card
_token=ovDZVQkDJLDPXnMmSc0iA58ncauDK00Go7MD7aWAGD7sFSh6EBu3V0&txnDate=2021-07-11
12:45:16
&checksumhash=nys/D601sfLiSNMeFzAGDIUX5aS/6eqN2nAvWLgLIV2rTCT3EevonasINDRZGzLWuQ5
gFDBKJHZN0301NSOHCU0WddDhGiLBYVe70jILRo' -X POST https://
sadamq.com/jslib/callapi.php
```

To generate the checksumhash for the form data, a merchant can use Sadad's checksum generation API.

### Response:

```
{"status":"success","transaction_number":"SD2618692930920","TXNAMOUNT":50,"ORDERID":"SD2618692930920"}
```

### Error:

```
{"status":"failed","error_code":"ERR-05","error_message":"Token is blocked because of too many wrong attempts. Please contact administrator to unblock."}
```