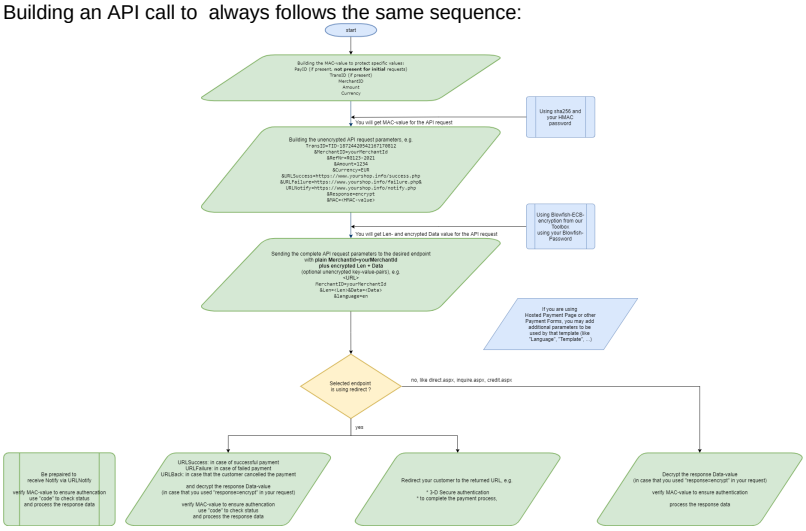
Create an API call and use Blowfish encryption

How to create an API call

Building an API call to always follows the same sequence:



Libraries for HMAC and Blowfish

Here are some libraries supporting you with HMAC calculation and Blowfish-Encryption: Implementation Sources

Merchant credentials

You will receive your merchant credentials after signing up the contract.

The merchant credentials consist of:

- MerchantId: Your MerchantId on
- HMAC-password: A password to calculate the MAC-value to protect specific values in the request (e.g. amount, currency) or response (e.g. status, code)
- Blowfish-password: A password to encrypt your request to and its response.

You may receive a multiple set of MerchantId's and password's as a MerchantId is used to unite a set of configurations (e.g. paymethods, currencies, services).

Example

> We would like to create a payment for 12,34 EUR with english language for the hosted payment page with additional template parameters.
>
> 3-D Secure 2.x shall be used in case that the customer selects credit cards (e.g. Mastercard, VISA, American Express), but also other paymethods like PayPal, Direct Debit, Sofort, ... can be selected.

Therefore we need:

1. MAC calculation to secure amount and currency
2. put API parameters together - unencrypted
3. encrypt all API parameters by this we will get "Data" + "Len" for the API request
4. add plain parameters to customize Hosted Payment Page using (HPP) with a template, e.g. language="en" for using the HPP with preselected english language
5. send the API request to the desired endpoint.

MAC calculation

The MAC is calculated always like this: `HmacSHA256("PayId*TransID*MerchantID*Amount*Currency", "YourHmacPassword")` where:

Key	Value	Comments
PayId	Referenced PayId	May be empty, e.g. for creating an initial payment process or risk management request; is used with subsequent requests like capture/refund.
TransId	Your transactionId to reference / identify your request	Your own reference to identify each request / payment process.
MerchantId	Your MerchantId	Your MerchantId assigned to you by identifiying this request.
Amount	Amount in smallest unit of currency, e.g. 123=1,23 (EUR)	Amount of this request; may be empty if not used, e.g. for status inquiries.
Currency	Currency of payment process in ISO 4217, e.g. EUR, USD, GBP	Currency of this request; may be empty if not used, e.g. for status inquiries.
YourHmac Passwort	Your HMAC-password assigned to you by	Your HMAC-password assigned to a specific MID; if you have different MIDs you will have different HMAC passwords, too.

Notes:

- in case that a value is not present just leave it empty, e.g.:
  - with amount/currency, without PayId to initiate a new payment - like in this example: `HmacSHA256("*TID-445373212216711455 8*yourMerchantId*1234*EUR", "mySecret")`
  - with amount/currency, without PayId, without TransId: `HmacSHA256("**yourMerchantId*1234*EUR", "mySecret")`
  - with PayId, without amount/currency: `HmacSHA256("fe3f002e19814eea8aa733ec4fdacafe*TID-445373212216711455 8*yourMerchantId**", "mySecret")`
- you will find more details for HMAC-calculation
  - for requests: HMAC Authentication (Request)
  - for responses/notify: HMAC Authentication (Notify)

You can find an application to verify for MAC-calculation here: Simple Test Tools

Raw parameters before encryption

The raw parameters define basic settings for this payment call, e.g. your MerchantId, amount, currency, your reference and URLs for success, failure and notify:

Key-Value	Comments
MerchantID=yourMerchantId	Your MerchantId to identify your request at
MsgVer=2.0	Indicate that 3-D Secure 2.x shall be used; Specially for 3-D Secure 2.x it is useful to provide additional data (like billing- and shipping-address) to improve frictionless processing (i.e.: payment is authenticated without challenge). These additional data are provided in JSON-structure.
TransID=TID-187244205421671 70812	Your request identifier
RefNr=RG123-2021	Your payment process reference
Amount=1234	The desired amount in smallest currency unit, e.g. 1234 + EUR 12,34 EUR
Currency=EUR	and currency
URLSuccess, URLFailure, URLBack	URLs for forwarding the customer in case of success, failure, cancel
URLNotify	URL to receive notifies
Response=encrypt	shall respond with encrypted data
Language=en	Customer wants english language

Parameters before encryption
MerchantID=yourMerchantId&MsgVer=2.0&TransID=TID-187244205421671708 12&RefNr=RG123-2021&Amount=1234&Currency=EUR&URLSuccess=https://www.yourshop.info/success.php&URLFailure=https://www.yourshop.info/failure.php&URLNotify=https://www.yourshop.info/notify.php&Response=encrypt&MAC=ca3c75eaf2120dfd15de77af2398b1561d8473f647b72aa7270fde94df7756d6&Language=en

As "=" and "&" are used for building key-value-pairs these characters **must not** be part of **any value**.

Do not send empty values, but only keys which are required and really having values.

For credit card processing with 3-D Secure 2.x (EMV 3DS) you must add "MsgVers=2.0"

Hosted Payment Page works like a proxy for the other payment forms (i.e. Credit Card Form (PaySSL), Direct Debit Form (PaySDD), paymethod specific forms (e.g. PayPal))

- so you have to add "MsgVers=2.0" to enable 3-D Secure 2.x for Credit Card Form (PaySSL)
- you may supply other key-values for other paymethods (e.g. PayPal)

Encryt parameters into Data/Len

The raw parameters need to be encrypted via Blowfish ECB and then hex-encoded. We provide you predefined functions in our toolkits for a quick start. The above shown and unencrypted data will be encrypted into:

Encrypted Data / Len
Len=342&Data=550a705ffb8fb2d59f72a116a55b26ef97d92d6ebec45ea10efe79b05d93f6fd6a69e8f810509d7e754899153e459f05cfd2c9bd9e71f92acda33e453f329a641328b32411b5c08ded80711c13d64e01d2cbae26a884f8c8781db17f31434fe34032ec5ef961dfb53006add8abc7f9cfa39d582962f3a70af105eb1f2240376e358d7cf8a7dadfef6afeac3bf0f043f578f5040995a7b29ca23fbcc0f84f5e416fe1ef60c3ff58028b3aa017b2fb50715fc3ef42b3947cc89f2639f61f6dbd8fea1f0b17716115676b2762ea14f0ca8d6fe1ef60c3ff58028b3aa017b2fb50715fc3ef42b3947cc86f80ef28614f1f739345550e4a0fc83bc7ca605f8477fcb5fa93da00daa0bddf9faace1829eeea32c8ffd87cffaf85930479d79b121d2662172cd81022e35232777bedb997aeb1deba02dc2fc4af5297802ace7338757a9058e220fd1c0abb8b2f3d8309b9d7375cae9897dc2aeda201

Notes:

To ease your integration we provide predefined functions to help you with Blowfish ECB:

Your language	Where to find
ASP	txmsCrypto.dll // txmsCrypto.BlowFish
ASP.NET	CompuTop.Core.Crypto.dll // CompuTop.Core.Crypto.BlowFish
Java	Blowfish.java
PHP	function.inc.php ctHMAC ctEncrypt ctDecrypt

The value for "Len" is the **string length** of the **unencrypted parameter** string built in the step before.

Now putting all together - the API request

The API request is then built like:

Value	Comments
Basic parameters	

e.g. paymentPage.aspx	Selected desired endpoint of
MerchantID=yourMerchantId	Your MerchantId to identify your request at (here additionally as plain value)
Len=<Len>&Data=<Data>	Length of the uncrypted parameter string and Data returned by Blowfish-Encryption
Template parameters (plain)	
Template=PaymentPageDropDown_v1	Template for Hosted Payment Page (HPP)
CCTemplate=Cards_v1	Template for Credit Card Form (called by HPP)
SDDTemplate=DirectDebit_v1	Template for Direct Debit Form (called by HPP)
Language=en	Starting laguage for the customer
CustomField1..16	Some CustomField-Data to display additional information on the HPP depending on the template

Putting API-call together

The API-call consists of:

Category	Description
endpoint	e.g. https://www.computop-paygate.com/paymentPage.aspx
MerchantID=<>	MerchantID=YourMerchantID
Len & Data	Encrypted data containing request data
additional params	Additional key/values in plain, not encrypted

Building the API-call

```
https://paymentpage.axepta.bnpparibas/paymentPage.aspx?
MerchantID=yourMerchantId&Data=550a705ffb8fb2d5f5694cad1b6e61237d35e1834c87
28baa5b48f831cce342ffb0f6ee9876fea46f0c9da4224b3559fc316aa2a253347167c8922c
f40eb9a1d25b9b63ddf39f2c8c60bdadd0a6acda9b86e7c44a404f0bb93d974c6b6365d9c95
68d959ff666e47b145d79d030a22f9cf1da025435595926b7e2788fb0004190c3cfdb65d3b5
dce626385d41347aeb276a4dc2917b156a0101b0fd0788bb891a39d582962f3a70af105eb1f
2240376e358d7cf8a7dadfef2fd3fbb52edb398878f5040995a7b29c8d409787275b6ad1a39
d582962f3a70af105eb1f2240376e358d7cf8a7dadfef586eaac76b23528323aa63cb973493
76352eca4a77a129bda3b8fc184ffd9dfbe9cff61196ac80f1429815d4444150f543eb60c0a
1f3a706ea77f63eccd216a2188cf94505690adfbb95d59473f3ee6b7252654a65f1f6525238
d93eea0ae67abc82ed7ecd23bb957791f6e4e2c34fa4&Len=341&Template=PaymentPageDr
opDown_v1&Language=en&CCTemplate=Cards_v1&SDDTemplate=DirectDebit_v1&URLBac
k=https%3A%2F%2Fwww.yourshop.info%2F&CustomField1=12%
2C34+EUR&CustomField2=Order+Text&CustomField3=https%3A%2F%2Fwww.paytest.
info%2Fphantasy-logo.
png&CustomField4=Shopping+Cart&CustomField5=Company+Name&CustomField6=First
+Name&CustomField7=Stra%C3%
9Fe+4&CustomField8=12345+City&CustomField9=Shipping+Company&CustomField10=S
hipping+Name&CustomField11=Shipping+Street&CustomField12=23456+Shipping+Cit
y&CustomField13=RG-Inv+123%
2F2021&CustomField14=Some+Label&CustomField15=Some+Text&PayType=0
```

This URL doesn't work and will result in "Unexpected exception", because MerchantId "yourMerchantId" doesn't exist. You will find some working URLs below.

Sending the API request

A request can be sent either via GET or POST. We recommend to use POST for two reasons:

with GET the parameter length is restricted to 2048 bytes depending on the browser, while with POST the request length is limited to 5120 bytes; If you require longer strings please contact

via GET the parameters are attached to the URL which can be easily manipulated by a customer - therefore prevents manipulation using Blowfish encryption

Checking the response

Server-2-server response

Some tips on responses

With server-2-server requests a request will respond with a direct response containing

- a status indicating success or failure of transaction
- a code (response code) explaining details of transaction along with a description
- other data like PayId for each payment process
- and other data depending on the type transaction

Payment page / ansynchronous notification

In case of a redirect payment an ansynchronous notification is sent to your system indicated by a URLnotification.

The response can be either encrypted or in plain text - we recommend an encrypted response.

Common

Please check:

- just checking whether "URLFailed" or "URLSuccess" has been called is not sufficient and can easily misused
  - Response code: only "code=00000000" indicates a successful and completed action
- HMAC in response is valid - to ensure that the message is not manipulated

Some tips on responses

Result	Description
Unexpected exception	response not containing "code" and "status" a very likely reason is that you simply used • a **wrong template** name or **wrong MerchantId**. See more details in our documentation for Payment Pages
• PayId=0000....0000 • code=8 digit	detected some error in request and a payment process has not been created. *These payment processes can not be found in .* An overview of response codes can be found here: Response codes
• valid PayId • code=8 digit	A payment process with PayId has been created but the subsequent systems detected an error. An overview of response codes can be found here: Response codes
• code=xxxx0005	The amount is the first parameter which is checked. The amount must be given in numbers without decimals. But you may also simply have **mixed up your MIDs and Blowfish-keys** - just doublecheck.
• valid PayId • code=00000000	Payment / process successful and completed
• valid PayId • code=0	Payment / process pending

A few URL calls to play with

Please find some test data to play here: Test Guide. However, the payments may result in error to prevent abuse.

Click and try	Comments	Notes
Click and try	Link to Hosted Payment Page without specific template data to initiate a payment process for 12,34 EUR	no template data specified

Click and try	The **same data** (Len + Data) can be used with Hosted Payment Page using a different template for Hosted Payment Page itself and with specific templates for selected credit card payments and direct debit payments	As we start "Hosted Payment Page" the template refers to a HPP-template and we add template names for subsequent payment forms for credit card and direct debit payments: Template=PaymentPageDropDown_BNP_v1&Language=en&CCTemplate=Cards_BNP_v1&SDDTemplate=DirectDebit_BNP_v1 We also add some CustomFields to display some additional customer information, **just by changing** "CustomField3" you can refer to your own logo. (ℹ Only available if the "CustomFields" are supported by the templates)
Click and try	The **same data** (Len + Data) can be used withCredit Card Form (PaySSL)	As we start "PaySSL" the initial template name refers to a specific credit card template named "Cards_v1": Template=Cards_BNP_v1&Language=en **Just by changing** "Template=ct_responsive" you can use a different payment form design