

APEXX

HMAC Signature Validation Implementation

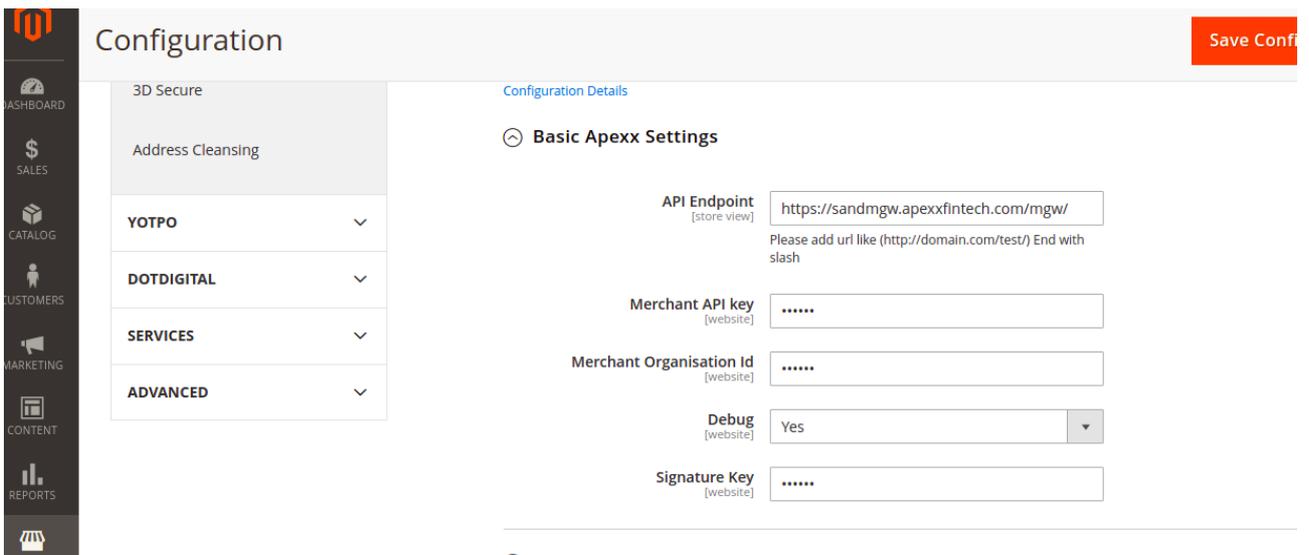
Document Version	Author	Signature
1.0	Priyanka Mulye	

Please find below the HMAC Signature Calculation process in Magento which is followed to validate the authenticity of a redirect POST request from APEXX.

1. We have used the following **Secret Key**.

97bc52f0cf904760b6c047b0ed28d2ac

2. From **Admin Configuration**, we have created a new field "Signature Key" under **Basic Apexx Settings**, here we have inserted our Secret key. It is stored in encrypted format.



The screenshot shows the Magento Admin Configuration interface. On the left is a sidebar with navigation icons for Dashboard, Sales, Catalog, Customers, Marketing, Content, and Reports. The main content area is titled "Configuration" and has a "Save Conf" button in the top right. A left-hand menu lists configuration sections: 3D Secure, Address Cleansing, YOTPO, DOTDIGITAL, SERVICES, and ADVANCED. The "Basic Apexx Settings" section is expanded, showing the following fields:

- API Endpoint** [store view]: (Note: Please add url like (http://domain.com/test/) End with slash)
- Merchant API key** [website]:
- Merchant Organisation Id** [website]:
- Debug** [website]:
- Signature Key** [website]:

3. In our code, we have Base Helper file which has function to fetch the Admin Configuration Secret Key. It decrypts and returns the Secret key which we have saved as described in previous point.



REDSTAGE

```
/**
 * @return string
 */
public function getSignatureKey()
{
    $signatureKey = $this->getConfigValue(self::XML_PATH_SIGNATURE_KEY);

    return $this->encryptor->decrypt($signatureKey);
}
```

4. We take the Response Array as it is, please find below sample response.

```
(
  [reason_code] => 0
  [_id] => e584c4f891eb422e82551ae240fb89c0
  [authorization_code] => 382419
  [merchant_reference] => aG9zdGVkMDAwMDAwMTU1
  [amount] => 1000
  [status] => AUTHORISED
  [card_number] => 454305XXXXXX9982
  [expiry_month] => 12
  [expiry_year] => 23
  [signature] => /jC+CTiVxw9j+AyjbxoWdpZwvMeByvW7T7IOfgfHS8RkpfmsSr5q8GY0FLKnjzNDni6010TmgxcLkTcYdkPQ==
)
```

5. We remove signature parameter from the response.

```
//Remove Signature parameter from response
unset($response['signature']);
```

6. Sort the response, Example below reflects the sorting*:

```
(
  [_id] => e584c4f891eb422e82551ae240fb89c0
  [amount] => 1000
  [authorization_code] => 382419
  [card_number] => 454305XXXXXX9982
  [expiry_month] => 12
  [expiry_year] => 23
  [merchant_reference] => aG9zdGVkMDAwMDAwMTU1
  [reason_code] => 0
  [status] => AUTHORISED
)
```

7. Converted sortedPairs into the JSON String.



REDSTAGE

```
unset($response[signature]);  
  
//Sort the response in ascending order  
ksort($response);  
  
//Encode the response  
$data = json_encode($response);
```

8. Example below reflects the JSON String:

```
{"_id":"e584c4f891eb422e82551ae240fb89c0","amount":"1000","authorization_code":"382419",  
card_number":"454305XXXXXX9982","expiry_month":"12","expiry_year":"23","merchant_reference":"aG9z  
dGVkMDAwMDAwMTU1","reason_code":"0","status":"AUTHORISED"}
```

9. We pass this JSON string to our Helper function which calculates the HMAC with the signing string, using the cryptographic hash function SHA-512.

Encode the result using the Base64 encoding scheme to obtain the signature.

```
public function signatureEncryptDecrypt($method, $string){  
    $secret = $this->getSignatureKey();  
  
    if($method == self::ENCRYPT){  
        $enc_key = hash_hmac('sha512', $string, $secret, true);  
  
        return base64_encode($enc_key);  
    }  
}
```

```
2021-12-17T10:38:39+00:00 INFO (6): signature encrypt |  
2021-12-17T10:38:39+00:00 INFO (6): /jC+ddCTiVxw9j+AyjbxoWdpZwvMeByvW7T7IOfgfHS8RkpfmsSr5q8GY0FlKnjnzNDni6010TmgxclKtCydKpQ==  
signature calculated, for example, key-value pairs and HMAC key is:
```

11. We check if the signature obtained from response and signature obtained after HMAC calculation matches.



```
//Check if signature is matched then place order
if (($signature == $responseSign) && ($status == 'AUTHORISED')) {
    $order = $this->order->loadByIncrementId($incrId);
    //$orderId = $order->getId();
    $orderObj = $this->orderRepository->get($order->getId());
}
```

12. If the signature matches then Order processing is further carried out.