



HOSTING
CONTROLLER

GATEWAY OPEN API MANUAL
For Developers

Contents

Proprietary Notice	3
Document Conventions	3
Target Audience	3
Introduction.....	4
How Open API Facilitates Accepting Online Payments	4
Why Open API is Needed.....	4
Gateway Open API Integration.....	5
Implementation Details	5
Contact Us	15





Proprietary Notice

© 2010 Hosting Controller. All Rights Reserved.

This document is the property of, and contains proprietary information of Hosting Controller. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying or recording, for any purpose other than consideration of the technical contents without the written acquiescence of a duly authorized representative of Hosting Controller.

Document Conventions

TERMS/ SYMBOLS	WHAT THEY MEAN?
Bold	The 'Bold' text is used to highlight vital terms in the document.
	This symbol is used as a sign for NOTE. It reminds about certain noteworthy steps or takes your attention towards the significant tasks to be done or gives additional information for your ease.
	This symbol is used to show sample codes related to various XML nodes.

Target Audience

This document is intended for those who need to integrate their **PAYMENT GATEWAYS** with **HOSTING CONTROLLER (HC)**. It explains all the technicalities and functionalities of integrating Gateways to the panel.

Introduction

HC provides an **OPEN API** for developers that allow your company to seamlessly integrate your **PAYMENT GATEWAY(S)** in HC. Integrating Gateway means that your Gateway is visible on front end and your clients pay their bills through your Gateway using HC.

How Open API Facilitates Accepting Online Payments

OPEN API is a tool that lets you accept payments for your service via HC. When your client subscribe to your service, they will be billed automatically according to the terms you dictate, removing the hassle of keeping track of what payments you have and have not received.

HC facilitates with the proficient billing system to save your time and money:

- **Easy to implement** - Flexible and automatic billing frees you from sending invoices.
- **Sell with ease** – HC maintains detailed transaction records in your control panel.

Why Open API is Needed

HC ensures reliable and easy integration of new Gateway(s) as required. In case you don't want to accept payments with the existing Gateways in the panel, you can add/integrate new Gateway(s) of your choice as per your requirements.

Once these Gateway(s) are added you can send/receive request/response to/from the Gateway. While handling response from the Gateway, you are required to write a response parser of your own.

The following section of the document explains the entire process in detail.

Gateway Open API Integration

GATEWAY OPEN API allows you to integrate your Gateway with **HC**, so you get instant payments.

Prior to add a Gateway, you need the followings:



- A Gateway Account
- Gateway API Implementation Documentation (From Gateway Website)

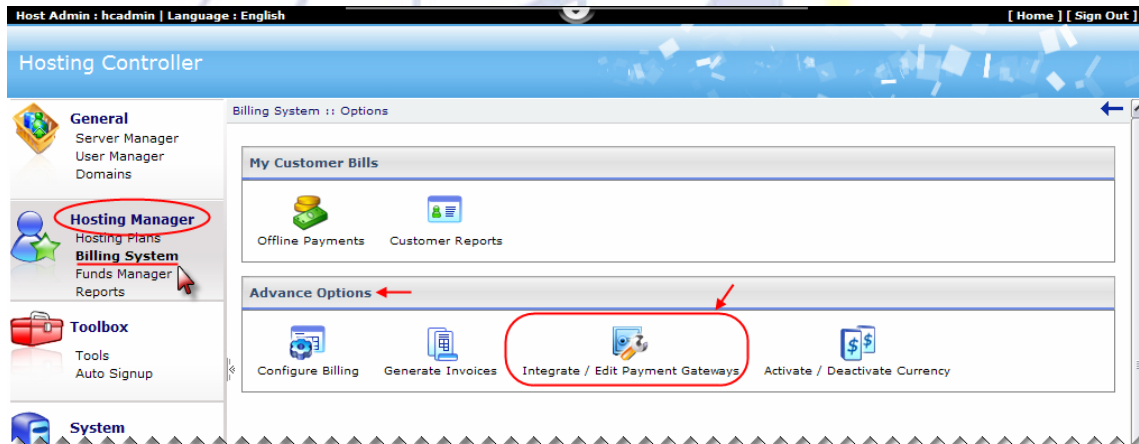
Implementation Details

Follow the steps below to integrate your Gateway with Hosting Controller.



Before proceeding any further, make sure you have read the Gateway API documentation in detail and know its required variables.

1. Login to **HC** as Host Admin. Select **Hosting Manager** >> **Billing System** >> **Advance Options** and then **Integrate / Edit Payment Gateways**.



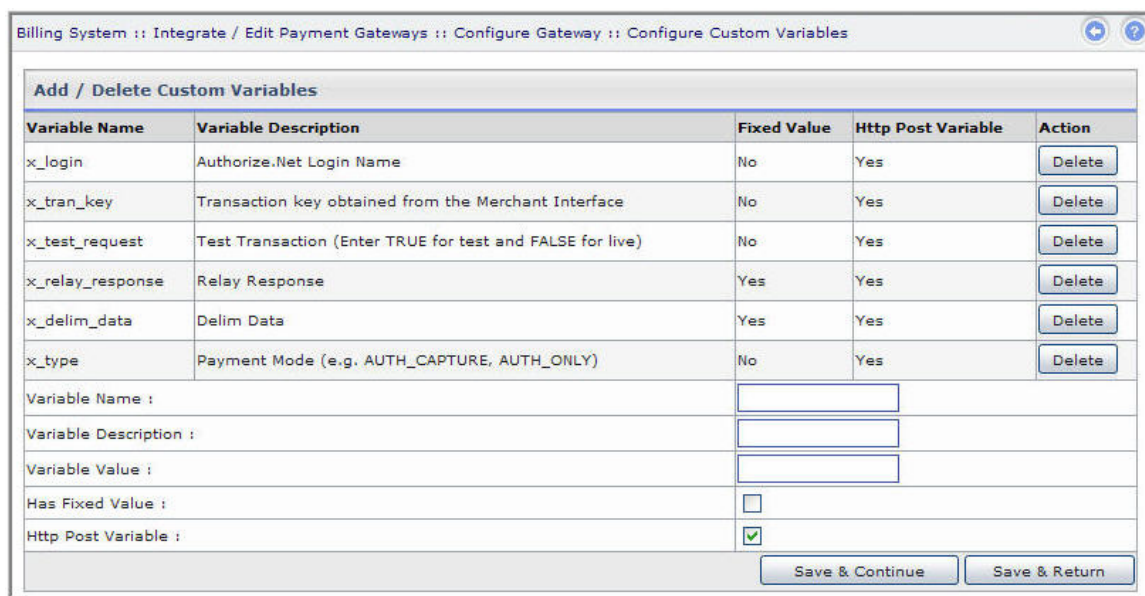
It opens following screen:

Gateway Open API Manual For Developers

2. Select “New Gateway” and then “Next”. It opens following screen:

Gateway Open API Manual For Developers

Specify the basic Gateway variables. These values are saved as variables in **HC** database and based on these variables Hosting Controller generates appropriate forms for making payment through your Gateway. Selecting “Next” opens following screen:



Variable Name	Variable Description	Fixed Value	Http Post Variable	Action
x_login	Authorize.Net Login Name	No	Yes	Delete
x_tran_key	Transaction key obtained from the Merchant Interface	No	Yes	Delete
x_test_request	Test Transaction (Enter TRUE for test and FALSE for live)	No	Yes	Delete
x_relay_response	Relay Response	Yes	Yes	Delete
x_delim_data	Delim Data	Yes	Yes	Delete
x_type	Payment Mode (e.g. AUTH_CAPTURE, AUTH_ONLY)	No	Yes	Delete

Variable Name :

Variable Description :

Variable Value :

Has Fixed Value : ☐

Http Post Variable : ☒

Save & Continue Save & Return

Select to Save and Add
Another Variable

Select to Save
and Return



All the variables depend on the selected Gateway. For example, one Gateway may require “Customer Address” and the other may not.



Each merchant holds unique Gateway account information that is sent to the Gateway with transaction to adjust the amount against the merchant.

3. Create custom variables as required. These variables are available when you activate your Gateway and select “Set Account”.

For example, Authorize.Net requires HTTP Referrer URL and Authorize.Net Login Name to be send to Authorize.Net with each transaction and the variable name shall be “referrer” and “x_Login” respectively. So, while integrating Authorize.Net, we create two client side variables “referrer” and “x_Login” with variable description “HTTP Referrer URL” and “Authorize.Net Login Name”.

On selecting “Account Settings” (Preferences->Payment Gateway->Account Settings) values of these variables are provided there. Each reseller can provide his own values for his Authorize.Net account.

The details of the variables are as follows:

Gateway Open API Manual For Developers

Variable Name	Is Compulsory	Description
Gateway Name	HTML Gateway: YES	The Gateway identification name. This is the name which will appear in Control Panel. This is not editable once entered.
	API Gateway: YES	
Gateway URL	HTML Gateway: YES	See step 4 for this URL. This could be the Gateway URL. For example: https://select.worldpay.com/wcc/purchase . OR This could be the user own written code/application which handles request to be sent to the Gateway. This URL shall point to the Gateway Handler page.
	API Gateway: YES	
Customer ID Variable	HTML Gateway: YES	This is the login ID variable of paying customer. This variable will return to Hosting Controller in order to settle the payment in favor of the paying customer.
	API Gateway: YES	
Card First Name Variable	HTML Gateway: NO	The variable which holds the First Name of paying customer.
	API Gateway: YES	
Card Last Name Variable	HTML Gateway: NO	The variable which holds the Last Name of paying customer.
	API Gateway: YES	
Payment Description Variable	HTML Gateway: NO	The variable which holds the reason or description of payment.
	API Gateway: NO	
Card Number Variable	HTML Gateway: NO	The variable which holds the credit card number.
	API Gateway: YES	
Card Type Variable	HTML Gateway: YES	The variable which holds the credit card type i.e. Visa, Master etc.
	API Gateway: YES	
Card CVV2 Variable	HTML Gateway:	The variable which holds the

Gateway Open API Manual For Developers

	YES API Gateway: YES	CVV2 number.
Date takes separate Variables for Month and year:	HTML Gateway: YES API Gateway: YES	Enable this check box if your Gateway requires to send two separate variables holding card expiry Month and Year.
Card Expiry Month Variable	HTML Gateway: YES API Gateway: YES	This variable is required only if check box above it is checked. This variable holds the credit card expiration Month.
Card Expiry Year Variable	HTML Gateway: YES API Gateway: YES	This variable is required only if check box above it is checked. This variable holds the credit card expiration Year.
Card Expiry Date Variable	HTML Gateway: YES API Gateway: YES	This variable appears only if "Date takes separate Variables for Month and year" check-box is clear. This variable holds the card expiration date.
Card Expiry Date Format	HTML Gateway: YES API Gateway: YES	Select the card expiration date format from this select box.
Card Address Variable	HTML Gateway: NO API Gateway: NO	This variable holds the Address of credit card holder.
Card City Variable	HTML Gateway: NO API Gateway: NO	This variable holds the City of credit card holder.
Card State Variable	HTML Gateway: NO API Gateway: NO	This variable holds the State of credit card holder.
Card Country Variable	HTML Gateway: NO API Gateway: NO	This variable holds the Country of credit card holder.
Card Postal Code Variable	HTML Gateway: NO API Gateway:	This variable holds the Postal Code of credit card holder.

Gateway Open API Manual For Developers

	NO																									
Card Phone Number Variable	HTML NO	Gateway:	This variable holds the Phone Number of credit card holder.																							
	API NO	Gateway:																								
Card Fax Variable	HTML NO	Gateway:	This variable holds the Fax Number of credit card holder.																							
	API NO	Gateway:																								
Card Email Variable	HTML YES	Gateway:	This variable holds the Email of credit card holder.																							
	API YES	Gateway:																								
Decimal Positions in Amount	HTML YES	Gateway:	It will determine the decimal positions in the payment amount. This may be 0, 1, or 2.																							
	API YES	Gateway:	For example: <table><tr><td>Amount</td><td>Decimal</td><td>Positions</td></tr><tr><td>Final Amount</td><td></td><td></td></tr><tr><td>122.256</td><td></td><td>0</td></tr><tr><td>122</td><td></td><td></td></tr><tr><td>122.256</td><td></td><td>1</td></tr><tr><td>122.3</td><td></td><td></td></tr><tr><td>122.256</td><td></td><td>2</td></tr><tr><td>122.26</td><td></td><td></td></tr></table>	Amount	Decimal	Positions	Final Amount			122.256		0	122			122.256		1	122.3			122.256		2	122.26	
Amount	Decimal	Positions																								
Final Amount																										
122.256		0																								
122																										
122.256		1																								
122.3																										
122.256		2																								
122.26																										
Transaction ID	HTML NO	Gateway:	The variable which holds the transaction ID.																							
	API NO	Gateway:																								
Amount Variable	HTML YES	Gateway:	The variable which holds the payment amount.																							
	API YES	Gateway:																								
Amount Multiplier	HTML YES	Gateway:	If amount multiplier is given then payment amount will be multiplied to it before sending to the Gateway. The default value is “1” which means amount will be sent as is to the Gateway.																							
	API YES	Gateway:																								

Gateway Open API Manual For Developers

Is HTML Only	HTML YES	Gateway:	If this check-box is checked then the Gateway is considered HTML based and for HTML base Gateways the Hosting Controller can not perform recursive billing.
	API YES	Gateway:	
Currency Code Variable	HTML YES	Gateway:	The variable which holds the currency code for the Gateway.
	API YES	Gateway:	
Response Parser URL	HTML YES	Gateway:	This URL points to the code/application used to parse the response received from the Gateway in result of payment transaction request. For example “..\GatewayScripts\Authorize.Net.asp” is Gateway Response Parser URL for Authorize.Net in HC.
	API YES	Gateway:	
Response Variable	HTML YES	Gateway:	It indicates Gateway return transaction result is in which variable.
	API YES	Gateway:	
Response Approval	HTML YES	Gateway:	It holds value what should Gateway return in ‘Response Variable’ after successful transaction.
	API YES	Gateway:	
Amount Variable	HTML YES	Gateway:	It holds the amount of the transaction in the response received from Gateway.
	API YES	Gateway:	
Amount Return	HTML YES	Gateway:	It tells the Gateway whether to return to Merchant site or show Gateway own transaction result page.
	API YES	Gateway:	

4. User has to write some Handler that sends request to the Gateway. HC (code) sends request to the Gateway if it accepts requests in HTTP Protocol. In such case you don't have to write code to send request to the Gateway rather you have to provide the Gateway URL.

Example: **HC** will send data to the Authorize.Net 'GatewayURL' using HTTP Protocol.



```
x_cust_id=reseller1&x_first_name=Jhon&x_last_name=Smith&
x_card_num=UserCreditCardNo&x_exp_date=UserCardExpiryDate&
x_CVV2=&x_address=22&x_City=Orlando&x_State=Florida&x_Country=USA&
x_zip=32899&x_phone=555121288&x_Fax=555121288&
x_email=testaccount@mydomain.com&x_amount=100.00&
x_invoice_num=10000003&x_currency=USD&x_login=UserLogin&
x_tran_key=UserTransactionKey&x_test_request=TRUE&x_relay_response=False&
x_delim_data=True&x_type=AUTH_ONLY
```

If Gateway accepts request in other format(s) of data then you have to write code to send request to Gateway and "GatewayURL" should point to that Gateway Handler that sends request to Gateway. **HC** sends request data on provided "GatewayURL" in the format shown in above example. It's up to you to modify it in the format that Gateway accepts.



Other formats of data could be XML structures, SOAP, Gateway COMs.

5. Write the Gateway Handler. It is a piece of code that actually parse the response, received from the Gateway and return payment response to **HC** "Approved" or "Failed" in XML packet. Then **HC** registers payment in **HC** after "Approved" notification from Gateway Handler.



Example: Gateway Handler for API Based Gateway

Authorize.Net is an API Based Gateway and already integrated in **HC** using Open API. **HC** submits payment information to Gateway and sends the Gateway response to "Authorize.Net.asp" where response is parsed to check the payment is successful/failure. Set the Handler page URL in "Response Parser URL" field in Step2.

HC sends four variables to the Response Parser script/application:

- **GatewayName:** It holds the name of Gateway.
- **GatewayResponse:** It holds all the response received from the Gateway.
- **GatewayVariables:** It holds the name value pair of the custom variables of Gateway configured in **HC** which are sent in response to the Gateway.

- **TransactionID:** It holds the unique ID that **HC** use to keep the track of the transaction.

Your application do the parsing of the response it receives in the variable named as 'GatewayResponse'. You should know in response which variable holds the response Status/Error/Error Description/Transaction Description or any other variable [these vary from Gateway to Gateway] and return the response of the transaction Approved/Failed in the form of XML packet to **HC**. The parser URL parses the response and generates an XML packet. The format of the XML packet return to the **HC** should be:

```
<GatewayResponseParser>
<GatewayName>Authorize.Net</GatewayName>
<TransactionResult>Approved</TransactionResult>
<ResultDescription>Approved (TESTMODE) This transaction has been approved.
</ResultDescription>
<TransactionID>6f87d3ee-28b9-40f5-b41b-d7cf36bc4198</TransactionID>
</GatewayResponseParser>
```

Example: Gateway Variables hold value in name value pair format separated by comma.



GatewayVariables = GateWayUrl=https://secure.authorize.net/Gateway/transact.dll,
x_login=UserLogin,x_tran_key=UserTransactionKey,x_test_request=TRUE,
x_relay_response=False,x_delim_data=True,x_type=AUTH_ONLY,
ResponseVariable=,ResponseApproval=

6. To integrate any new Payment Gateway, Host needs to perform the followings:
 - a. Configure the URLs and POST/GET variables in the panel.
 - b. If Gateway accepts request in other data formats than HTTP Protocol then write script to send request to Payment Gateway.
 - c. Write a script to parse the response returned from the Payment Gateway and transform it to a format that **HC** understands (preferably in ASP) and place it in "GatewayScripts" folder in **HC** installation.



Example: Gateway handler to parse response for Authorize.Net

Authorize.Net.asp

```
<%  
Option Explicit  
Dim szResponse, szResponseDetails, szResponseArray  
Dim szGatewayName, szGatewayResponse, szGatewayVariables, szTransactionID  
  
szGatewayName = Request.Form("GatewayName")  
szGatewayResponse = Request.Form("GatewayResponse")  
szGatewayVariables = Request.Form("GatewayVariables")  
szTransactionID = Request.Form("TransactionID")  
  
szResponseArray = Split(szGatewayResponse, ",")  
  
IF (CStr(szResponseArray(0)) = "1") Then  
    szResponse = "Approved"  
    szResponseDetails = "Approved:" & CStr(szResponseArray(3))  
Else  
    IF (CStr(szResponseArray(0)) = "2") Then  
        szResponse = "Failed"  
        szResponseDetails = "Failed:" & CStr(szResponseArray(3))  
    Else  
        szResponse = "Failed"  
        szResponseDetails = "Failed:" & CStr(szResponseArray(3))  
    End IF  
End IF  
  
Response.Write(CreateXml(szGatewayName, szResponse, szResponseDetails,  
szTransactionID))  
  
Function CreateXml(szGatewayName, szParseResult, szParseResultDescription,  
szTransactionID)  
    Dim XmlString  
  
    XmlString = "<GatewayResponseParser><GatewayName>" & szGatewayName &  
    "</GatewayName><TransactionResult>" & szParseResult &  
    "</TransactionResult><ResultDescription>" & szParseResultDescription &  
    "</ResultDescription><TransactionID>" & szTransactionID &  
    "</TransactionID></GatewayResponseParser>"  
    CreateXml = XmlString  
End Function  
  
%>
```

Contact Us

In case of any ambiguity/query regarding Gateway Integration in **HC**, please feel free to contact us at support@hostingcontroller.com

