



Hosting Controller

Gateway Open API Manual

Issue 1.2

Control Panel Version 6.0

Contents

- 1. Introduction 3**
- 2. Gateway Open API in Action 4**
 - 2.1. Auto Sign-up Working for HTML Gateways 4
 - 2.2. Auto Sign-up Working for API Gateways 4
 - 2.3. Making Payment through Payment Form in Payments Section 4
 - 2.4. Recurring Payments5
- 3. Gateway Open API Integration 6**
 - 3.1 Prerequisites 6
 - 3.2. Implementation Overview 7
 - 3.3. Implementation Details 6
 - 3.4. Example Integrations 14

Hosting Controller Gateway Open API for Developers

1. Introduction

Hosting Controller provides an Open API for developers that allow your company to seamlessly integrate your payment gateway in Hosting Controller. Integrating gateway means that your gateway will be visible on front end and your clients will pay their bills through your gateway using Hosting Controller.

How Open API can help you accept online payments for your service

Open API is a tool that will let you accept payments for your service via the Hosting Controller control panel.

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.

Save time and money with Hosting Controller's hassle-free billing system:

- Easy to implement - flexible and automatic billing means frees you from sending invoices.
- Sell with ease – Hosting Controller maintains detailed transaction records in your control panel.

2. Gateway Open API in Action

How Gateway Open API works

2.1. Auto Sign-up Working for HTML Gateways

Step 1: Your buyer comes to your auto sign-up, selects a hosting package and clicks on “Make Payment” button on the final sign-up step.

Step 2: After your buyer makes the payment, he will see a confirmation message of “Verified” or “Invalid” transaction.

Step 3: At the time payment is made, your gateway will send a “verified” or “Invalid” notification regarding the payment at the URL specified in Gateway settings.

Step 4: On receiving the “Verified” notification, the payment will be settled in Hosting Controller against the customer. On receiving the “Invalid” notification, the customer can always try again.

Step 5: On receiving the “Verified” notification, customer clicks on “Create Domain” button to create his Account and Domain. Because the payment is already registered in HC in favor of customer therefore HC will create his account.

Step 6: Customer is shown his account information. A payment invoice is sent from Hosting Controller (e-mail templates are customizable).

2.2. Auto Sign-up Working for API Gateways

Step 1: Your buyer comes to your auto sign-up, selects a hosting package, provides his credit card information and clicks on “Create Domain” button on the final sign-up step.

Step 2: At the time payment is made, your gateway will send a “verified” or “Invalid” notification regarding the payment to Hosting Controller.

Step 3: On receiving the “Verified” notification, the payment will be settled in Hosting Controller against the customer. On receiving the “Invalid” notification, the customer can always try again.

Step 4: On receiving the “Verified” notification, HC will create his account.

Step 5: Customer is shown his account information. A payment invoice is sent from Hosting Controller.

2.3. Making Payment through Payment Form in Payments Section

Step 1: Customer signs-in to HC control panel.

Step 2: Customer clicks on Payment Form option in Payments menu. Payment form appears where user enters his billing information and clicks on “Pay Now!” button.

Step 3: After your customer makes the payment, he will see a confirmation message of “Verified” or “Invalid” transaction.

Step 4: At the time payment is made, your gateway will send a “verified” or “Invalid” notification regarding the payment at the URL specified in Gateway settings.

Step 5: On receiving the "Verified" notification, the payment will be settled in Hosting Controller against the customer. On receiving the "Invalid" notification, the customer can always try again.

Step 6: A payment invoice is sent to customer from Hosting Controller and customer can see a copy in Payments->Transaction Statement also.

2.4. Recurring Payments

Recurring payment are supported for **API gateways only** because HTML gateways require user to visit their website. For API base gateways, Hosting Controller Payment service charges credit card whenever payment is due.

Automatic recurring billing depends on certain parameters. See http://hostingcontroller.com/english/support/manuals/webadmin/index.htm#payment_options.htm for more details.

3. Gateway Open API Integration

Back-end Hosting Controller to Gateway integration

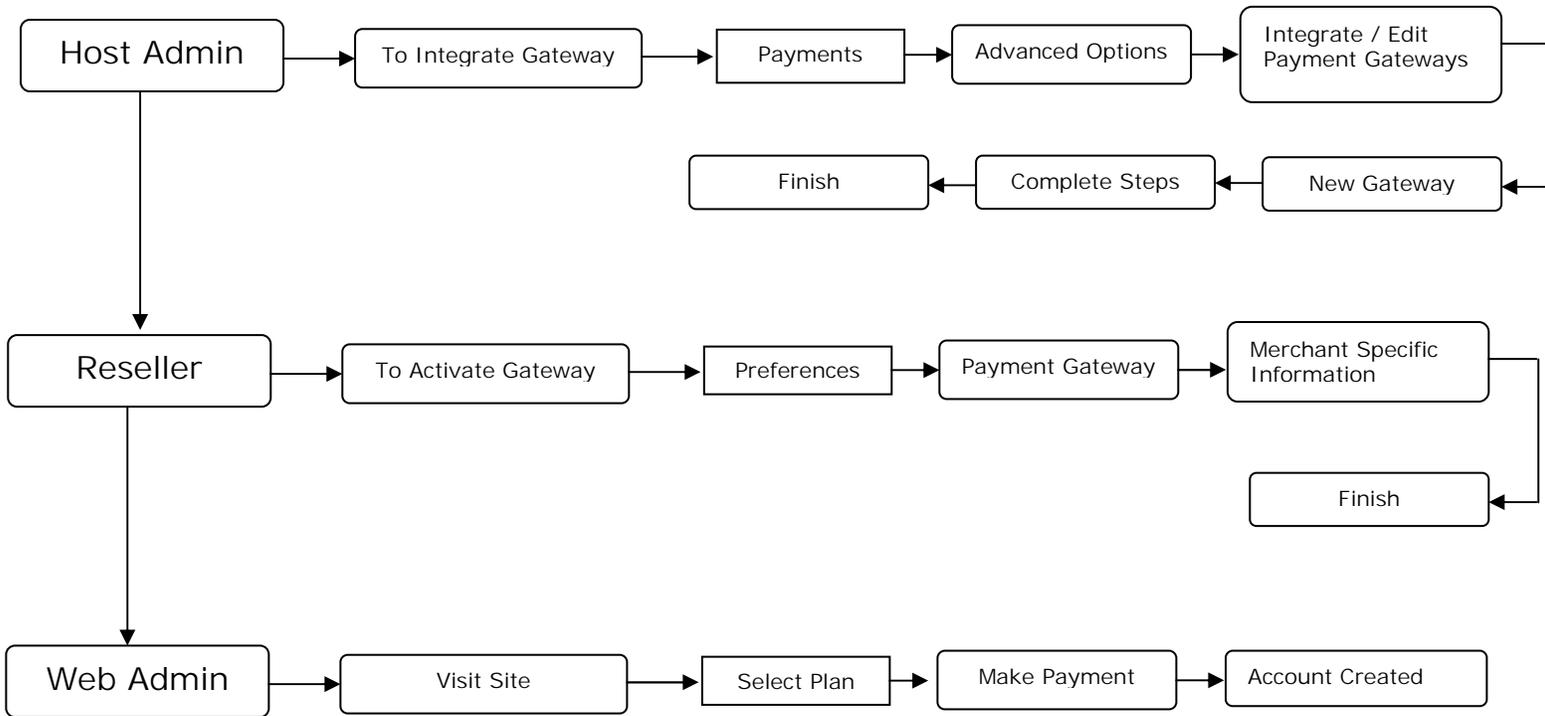
Gateway Open API allows you to integrate your gateway with Hosting Controllers, so you get instant payments through HC.

3.1. Prerequisites

When you integrate your gateway, you will need the following:

- You should have a basic understanding of ASP and HTML, and be able to write beginner level ASP code. *Required only if your gateway is API based.*
- A gateway account.
- Gateway API implementation documentation, shall be available from gateway website.

3.2. Implementation Overview



3.3. Implementation Details

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

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

Step 1: Login to Hosting Controller as Host Admin. Click "Billing Manager", click "Advanced Options" and then click on "Integrate / Edit Payment Gateways" button. Select "New Gateway Wizard" and click on "Next" Button.

Step 2: On New Gateway [Step 1] page, you will provide the basic gateway variables. When you provide values in the text fields on this page, they are saved as variables in Hosting Controller database and based on these variables Hosting Controller generates appropriate forms for making payment through your gateway.

You are not required to enter value in every text field because these variables depend on the gateway you are using. For example, one gateway may require "Customer Address" while another may not; therefore entering values (or creating variables in other words) in text fields depends on your gateway requirement.

The variables are described below. Click "Next" button after entering these variables.

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 3 for this URL. For HTML Gateway: In case of HTML based gateway, this will be an HTTP path. For example https://select.worldpay.com/wcc/purchase.
	API Gateway: YES	For API Base Gateway: For API based gateway, this URL shall point to the Gateway Handler page. For example “..\OpenApi\SendAuthorizeNet.asp” is Gateway Handler URL for Authorize.Net in HC.
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: YES	The variable which holds the CVV2 number.
	API Gateway: YES	
Date takes separate Variables for Month and year:	HTML Gateway: YES	Enable this check box if your gateway requires to send two separate variables holding card expiry Month and Year.
	API Gateway: YES	
Card Expiray Month Variable	HTML Gateway: YES	This variable is required only if check box above it is checked. This variable holds the credit card expiration Month.
	API Gateway: YES	
Card Expiry Year Variable	HTML Gateway:	This variable is required only if check box above

	YES API Gateway: YES	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: NO	This variable holds the Postal Code of credit card holder.
Card Phone Number Variable	HTML Gateway: NO API Gateway: NO	This variable holds the Phone Number of credit card holder.
Card Fax Variable	HTML Gateway: NO API Gateway: NO	This variable holds the Fax Number of credit card holder.
Card Email Variable	HTML Gateway: YES API Gateway: YES	This variable holds the Email of credit card holder.
Decimal Positions in Amount	HTML Gateway: YES	It will determine the decimal positions in the payment amount. This may be 0, 1, or 2. For example:

	API Gateway:	Amount	Decimal Positions	Final Amount
	YES	122.256	0	122
		122.256	1	122.3
		122.256	2	122.26
Transaction ID	HTML Gateway: NO	The variable which holds the transaction ID.		
	API Gateway: NO			
Amount Variable	HTML Gateway: YES	The variable which holds the payment amount.		
	API Gateway: YES			
Amount Multiplier	HTML Gateway: YES	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 Gateway: YES			
Is HTML Only	HTML Gateway: YES	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 Gateway: YES			
Currency Code Variable	HTML Gateway: YES	The variable which holds the currency code for the gateway.		
	API Gateway: YES			

Step 3: Note that each merchant holds his unique gateway account information and that information is required to send to the gateway along with transaction in order to settle amount against the merchant. You will create custom fields in this step to allow resellers using same gateway provide their account information. These variables will be available when you activate your gateway and click "Set Account" button.

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, when integrating Authorize.Net, we create two client side variables "referrer" and "x_Login" with variable description "HTTP Referrer URL" and "Authorize.Net Login Name". When you click "Account Settings" (Preferences->Payment Gateway->Account Settings) button, you may provide the values of these variables there. Each reseller can provide his own values for his Authorize.Net account.

The description of these variables is present on the interface.

Add/Delete Custom Variables

Variable Name	Variable Description	Fixed Value	HTTP Post Variable	Action
x_Login	Authorize.Net login name	No	Yes	Delete
Variable Name :		<input type="text" value="x_Id"/>		
Variable Description :		<input type="text" value="Authorize.Net password"/>		
Variable Value :		<input type="text" value="No"/>		
Has Fixed Value :		<input type="checkbox"/>		
HTTP Post Variable :		<input checked="" type="checkbox"/>		
		Save and add more		Save and Finish

Step 4: Write the gateway handler. Gateway handler is a small piece of code that actually sends the payment information to gateway and then registers payment in HC after “Verified” notification from gateway.

You will set the path to this file in gateway as gateway post back. Gateway will send HTTP post to this file. The HTTP post will be analyzed for success ad failure and appropriate action will be taken.

Example Gateway Handler for API Based Gateway: Authorize.Net is an API Based Gateway and already integrated in Hosting Controller using Open API. Hosting Controller submits payment information to “SendAuthorizeNet.asp” where they are handled for making payment and registering the payments in HC.

Set the handler page URL in “Gateway URL” field in Step 2 (see Page 8).

SendAuthorizeNet.asp

```

<!--#include FILE="..\Payments\inc_PaymentFunctions.asp"-->
<!--#include FILE="..\common\inc_Logging.asp"-->
<%
'// The Request.Form variables were set when gateway integration details were
'// given. You can see these variables at
'// HostAdmin->Preferences->Payment Gateway->Advance Options->Edit Gateway-
'//>Authrize.Net->Integration Details.

    x_Cust_ID           = Request("x_Cust_ID")
    x_firstname         = Request("x_firstname")
    x_lastname          = Request("x_lastname")
    x_address           = Request("x_address")
    x_City              = Request("x_City")
    x_State             = Request("x_State")
    x_Country           = Request("x_Country")
    x_zip               = Request("x_zip")
    x_phone             = Request("x_phone")
    x_Fax               = Request("x_Fax")
    x_Amount            = Request("x_Amount")
    x_card              = Request("x_card")
    x_CVV2              = Request("x_CVV2")
    x_exp               = Request("x_exp")
    referrer            = Request("referrer")
    x_Login             = Request("x_Login")
    
```

```

        x_email          =      Request("x_email")

'// This is payment handler. Function DoAuthnetCharging() sends the payment
'// to the gateway. You are required to write the payment handler yourself
'// for API based gateways.
        strResponse=DoAuthnetCharging(x_Login,trim(referrer),PayMode,TestMode,x_card,x_
exp,x_Amount,x_Cust_ID ,x_lastname,"Company",x_address,x_City,x_State,x_zip,x_phone,"
Fax",x_email)

'// The response from gateway is checked to determine if payment was a success or
failure.
        strResponse=Replace(strResponse,"","",")
        If ParseAuthnetResponse(strResponse)=1 Then

'// If payment was successfull then call AddPayment() function. this function
registers
'// the payment in Hosting Controller against the "CustName" login ID.

'// AddPayment(CustName,Amount, Description, OID, BAddr1, BCity, BState, BCountry,
Phone, Fax, CardNumber, EMail)
'// CustName = Customer Login Name
'// Amount = Paid Amount
'// Description = Payment details
'// OID = Order ID returned from gateway
'// BAddr1 = Billing Address 1
'// BCity = Billing City
'// BState = Billing State
'// BCountry = Billing Country
'// Phone = Phone Number
'// Fax = Fax number (optional)
'// CardNumber = Credit Card Number
'// EMail = Customer Email Address

        call AddPayment(x_Cust_ID,x_Amount,"Paid through HC Interface [Gateway:
Authorize.net]",x_trans_id,x_address,x_City,x_State,"Fax",x_Country,"Not
Specified",x_card,x_email)

'// In case of payment success the "x_response_reason_text" is returned by
Authrozie.Net
'// with message "The Transaction Has Been Approved".

        strResult=x_response_reason_text
        chk = "1"

Else
        chk = "0"
        strResult=x_response_reason_text&x_response_reason_code
        'Log Error for future debugging
        strResult1=strResult & "Payment method [Authorize.net] Customer ["&
x_Cust_ID &"]"
        WriteToLog strResult1
        'Redirect to Error displaying page
        Response.Redirect "paymenterror.asp?openapi=1&chk=" & chk & " &StrError="
& strResult
        End IF
%>

```

Example Gateway Handler for HTML Gateway: WorldPay is an HTML Gateway and already integrated in Hosting Controller using Open API. Hosting Controller submits payment information to the gateway and the gateway then sends payment notification to gateway handler page of HC.

The most important part of the configuration is to enter IPN URL in Gateway. This URL shall point to handler page. See your gateway documentation to learn more about setting IPN URL.

worldpay_ipn.asp

```

<!--#include FILE="..\Payments\inc_PaymentFunctions.asp"-->
<%
'// The Request.Form variables were set when gateway integration details were
'// given. You can see these variables at
'// HostAdmin->Preferences->Payment Gateway->Advance Options->Edit Gateway->WorldPay-
>Integration Details.

    amount           = request("authAmount")
    custname         = request("M_custname")
    postcode         = request("postcode")
    email            = request("email")
    tel              = request("tel")
    address          = request("address")
    rawAuthMessage   = request("authorised")
    transStatus      = request("transStatus")
    fax              = request("fax")
    desc             = "Paid thorough HC interface [Gateway:Worldpay]."
    transId          = request("transId")
    callbackPW       = request("callbackPW")

'// Please replace the value with your call back password and uncomment
'// the code below.

    mycallbackPW="callbackPW"

'// WorldPay server IPs
    ADDR1="195.35.90.62"
    ADDR2="195.35.90.61"

    THEADDR=request.ServerVariables("REMOTE_ADDR")

    If THEADDR= ADDR1 or THEADDR=ADDR2 then
'// Uncomment this code if security is required. Please do only if you know what you
are up to!!
        'if mycallbackPW=callbackPW then
        '    Source ="GOOD"
        'Else
        '    Response.Write "Hack attempt. The HTTP post was not sent from valid
WorldPay server."
        '    Response.End
        'End IF
    Else
        Response.Write "The HTTP post was not sent from valid WorldPay server."
        Response.End
    End If

'// If payment was successfull then call AddPayment() function. this function
registers

```

```

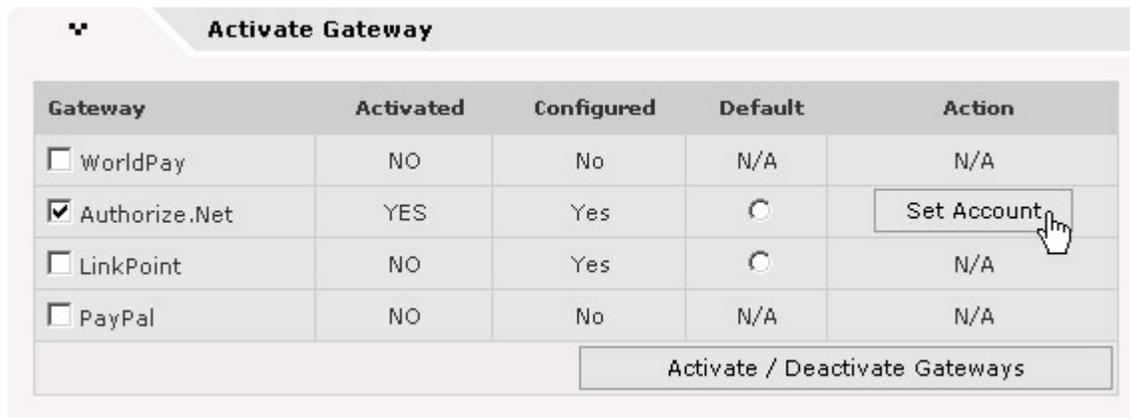
'// the payment in Hosting Controller against the "CustName" login ID.

'// AddPayment(CustName,Amount, Description, OID, BAddr1, BCity, BState, BCountry,
Phone, Fax, CardNumber, EMail)
'// CustName = Customer Login Name
'// Amount = Paid Amount
'// Description = Payment details
'// OID = Order ID returned from gateway
'// BAddr1 = Billing Address 1
'// BCity = Billing City
'// BState = Billing State
'// BCountry = Billing Country
'// Phone = Phone Number
'// Fax = Fax number (optional)
'// CardNumber = Credit Card Number
'// EMail = Customer Email Address

    If rawAuthMessage="authorised" and transStatus ="Y" then
        call AddPayment(custname,amount,desc,transId,address,"Not sepcified", "Not
sepcified", "Not Specified",tel,fax,"",email)
    End If
%>

```

Step 5: As you have successfully integrated your gateway, you may activate it now. To activate gateway, Login as Host or Reseller, click "Preferences", click "Payment Gateway", and select your gateway by clicking the check-box. Click on "Activate/Deactivate Gateways" button to activate your gateway. Click on "Set Account button" to provide merchant account information for selected gateway.



3.4. Example Integrations

Hosting Controller is officially supporting four gateways at the time of writing this document. These gateway integrations make use of same Open API framework therefore serving excellent examples of how gateways are integrated in HC. By studying these integrations, one can see practical examples for integrating own gateways.

To see examples, login to Hosting Controller as Host Admin. Click "Payment", click "Advance Options" and then click on "Integrate / Edit Payment Gateways" button. Select "Edit Gateway" and click on "Edit" button next to any gateway.

Note: We can integrate and officially support your gateway on the purchase of two extra licenses.