# INTEGRATING NEW PAYMENT GATEWAY WITH HOSTING CONTROLLER MARKETPLACE

Version 1.1

## Proprietary Notice

## Contents

## Target Audience

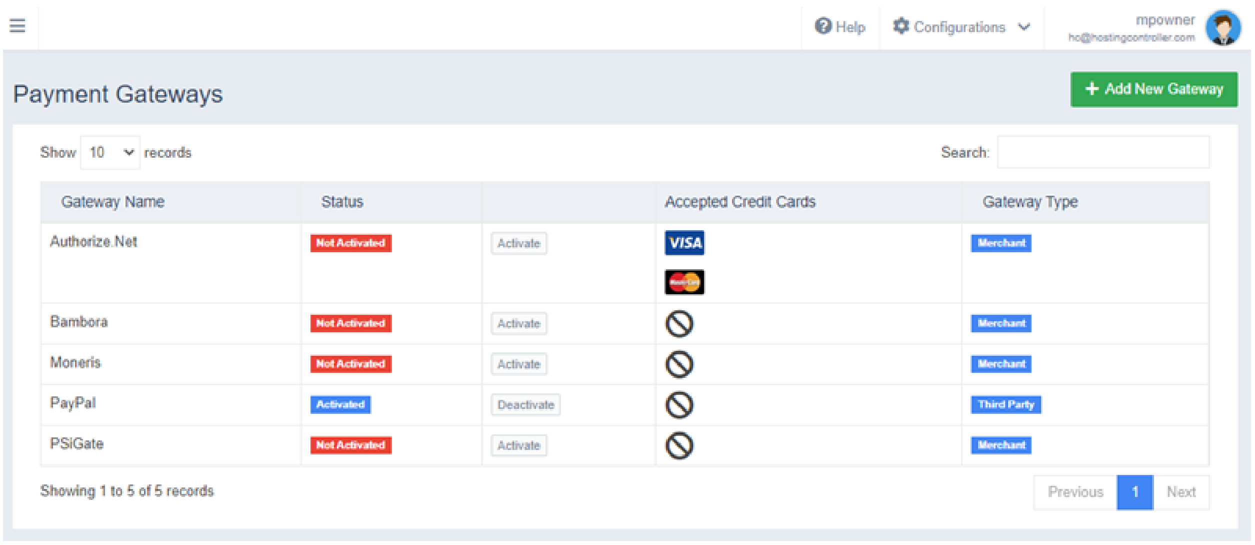
This guide is solely intended for those users who want to integrate their own payment Gateway with Hosting Controller Marketplace for processing online payments. Integrating a Gateway means that the intended Gateway will be visible at the front end and your clients will be able to pay their bills online through that Gateway. It is assumed that the integrator has sufficient working knowledge of Microsoft Visual Studio and C#.net and knows how to program in these tools.

## Introduction

Hosting Controller is a hybrid cloud control panel, designed to automate all aspects of your business including service provisioning, billing, metering, customer management and on-boarding of new services. It offers a fully integrated Marketplace, capable of accepting one-time and recurring online payments through tight integration of payment Gateways. If for any reason you don't want to utilize any of the already available payment Gateways in the Marketplace, you may choose to integrate your own Gateway. The objective of the guide is to elaborate the process of integrating your own Gateway in the Marketplace.

## Accepting Payments Online

For E-commerce merchants it's particularly important to accept payments online. Accepting payments online entails online credit card processing. A payment Gateway is the main link between your bank and the customer's credit card provider. HC Marketplace supports various payment Gateways. Gateways can be found in the "Payment Gateways" menu under "Configurations".



Payment Gateways

| Gateway Name | Status | | Accepted Credit Cards | Gateway Type |
|---|---|---|---|---|
| Authorize.Net | Not Activated | Activate | VISA, MasterCard | Merchant |
| Bambora | Not Activated | Activate | 🚫 | Merchant |
| Moneris | Not Activated | Activate | 🚫 | Merchant |
| PayPal | Activated | Deactivate | 🚫 | Third Party |
| PSiGate | Not Activated | Activate | 🚫 | Merchant |

Showing 1 to 5 of 5 records

*Figure 1: Third-party Gateways and Merchant Gateways in Marketplace.*

For users who don't want to accept payments with already available Gateways, the Marketplace sanctions easy integration of other payment Gateways of choice. This guide demonstrates the integration process of such Gateways of choice.

## Supported Gateways

HC Marketplace supports two types of payment Gateways.

### Third-party Gateways:

These Gateways direct the customers away from the Marketplace, taking them to the payment Gateway's website. Here customers fill in their payment details, and after paying, are redirected back to the Marketplace to complete the checkout process. A popular example of a Third-party Gateway is PayPal.

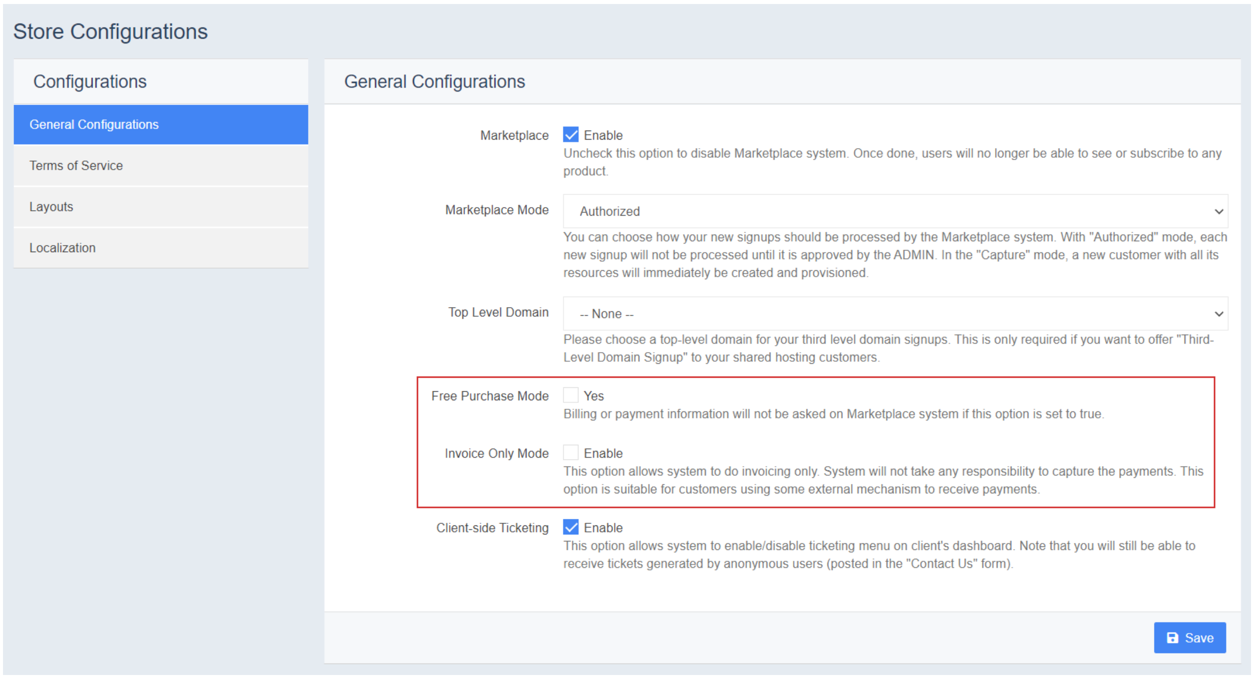
### Merchant Gateways:

These Gateways process the payments remaining within the Marketplace. They use the securely stored credit card information from the Marketplace database and process payment within it. An example of a Merchant Gateway is Authorize.Net.

HC Marketplace allows easy integration of new payment Gateways. There are two parts to these integrations. First where you add a new Gateway to Marketplace with its detail and second where you need to add its backend code to send request to the Gateway and parse response to decide the payment status.

## Part 1 (on Marketplace):

Before integrating a new Gateway make sure to uncheck "Free Purchase Mode" and "Invoice Only Mode" under "Store Configurations".



Marketplace: ✓ Enable
Uncheck this option to disable Marketplace system. Once done, users will no longer be able to see or subscribe to any product.

Marketplace Mode: Authorized
You can choose how your new signups should be processed by the Marketplace system. With "Authorized" mode, each new signup will not be processed until it is approved by the ADMIN. In the "Capture" mode, a new customer with all its resources will immediately be created and provisioned.

Top Level Domain: -- None --
Please choose a top-level domain for your third level domain signups. This is only required if you want to offer "Third-Level Domain Signup" to your shared hosting customers.

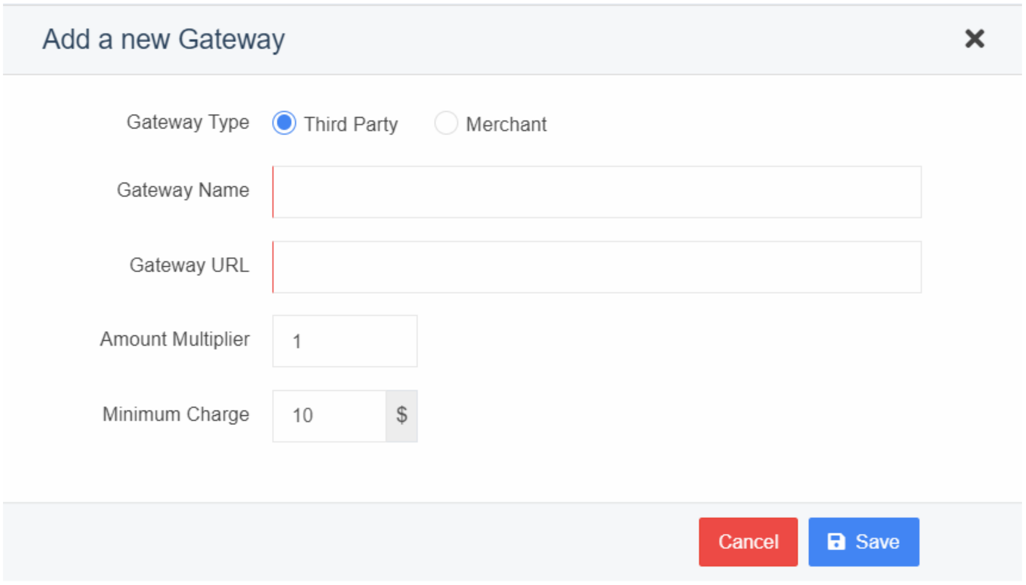
Free Purchase Mode: ☐ Yes
Billing or payment information will not be asked on Marketplace system if this option is set to true.

Invoice Only Mode: ☐ Enable
This option allows system to do invoicing only. System will not take any responsibility to capture the payments. This option is suitable for customers using some external mechanism to receive payments.

Client-side Ticketing: ✓ Enable
This option allows system to enable/disable ticketing menu on client's dashboard. Note that you will still be able to receive tickets generated by anonymous users (posted in the "Contact Us" form).

*Figure 2: Pre-requisites before integrating a new Gateway.*

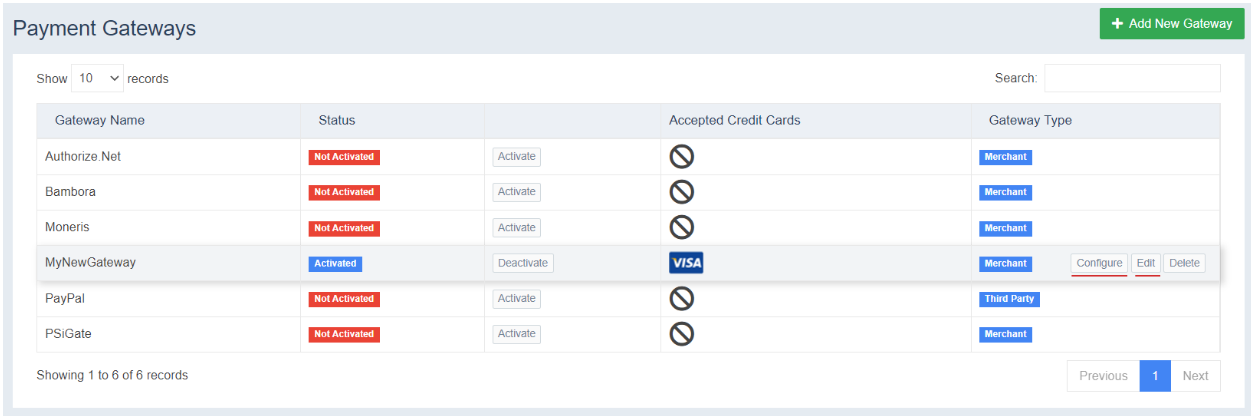
The first step in integrating a new Gateway is adding a new Gateway. Click "Add New Gateway" and depending on its type (Third Party or Merchant) add the new Gateway.


Add a new Gateway

| | |
|---|---|
| Gateway Type | [✓ Third Party] Merchant |
| Gateway Name | |
| Gateway URL | |
| Amount Multiplier | 1 |
| Minimum Charge | 10 $ |

Figure 3: Integrating a new payment Gateway.

Edit the newly added Gateway to specify Basic Settings, Custom Fields, Config Variables and Currency Settings. After specifying these configure the Gateway.

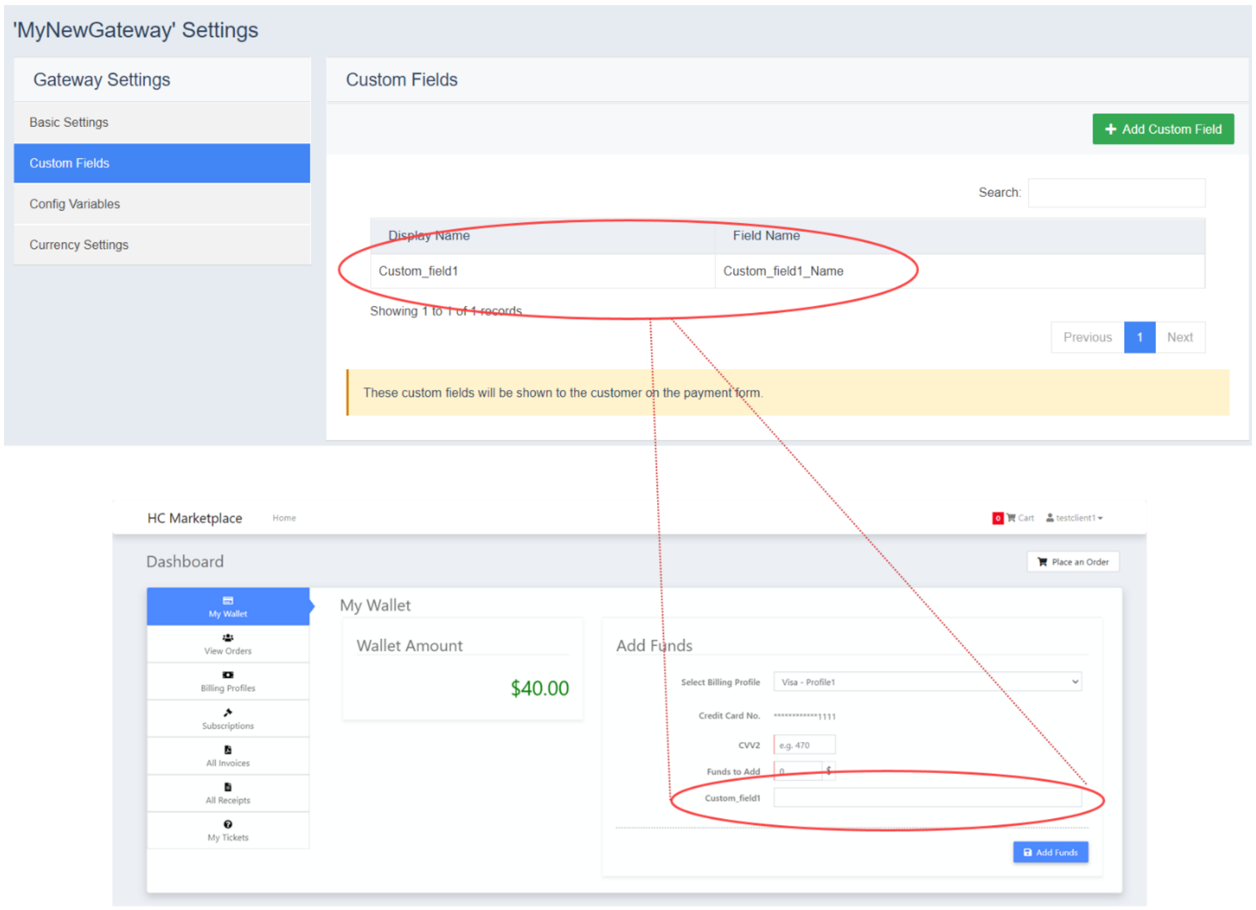

Payment Gateways

| Gateway Name | Status | | Accepted Credit Cards | Gateway Type |
|---|---|---|---|---|
| Authorize Net | Not Activated | Activate | ⊘ | Merchant |
| Bambora | Not Activated | Activate | ⊘ | Merchant |
| Moneris | Not Activated | Activate | ⊘ | Merchant |
| MyNewGateway | Activated | Deactivate | VISA | Merchant Configure Edit Delete |
| PayPal | Not Activated | Activate | ⊘ | Third Party |
| PSiGate | Not Activated | Activate | ⊘ | Merchant |

Showing 1 to 6 of 6 records

Figure 4: Editing and configuring a newly created Gateway.

The Gateway settings include additional fields (referred to as "Custom Fields") which will be displayed on actual payment form. Custom fields are used for asking additional information from the user while making payment.



'MyNewGateway' Settings

Gateway Settings

- Basic Settings
- Custom Fields
- Config Variables
- Currency Settings

Custom Fields

| Display Name | Field Name |
|---|---|
| Custom_field1 | Custom_field1_Name |

Showing 1 to 1 of 1 records

These custom fields will be shown to the customer on the payment form.

Wallet Amount

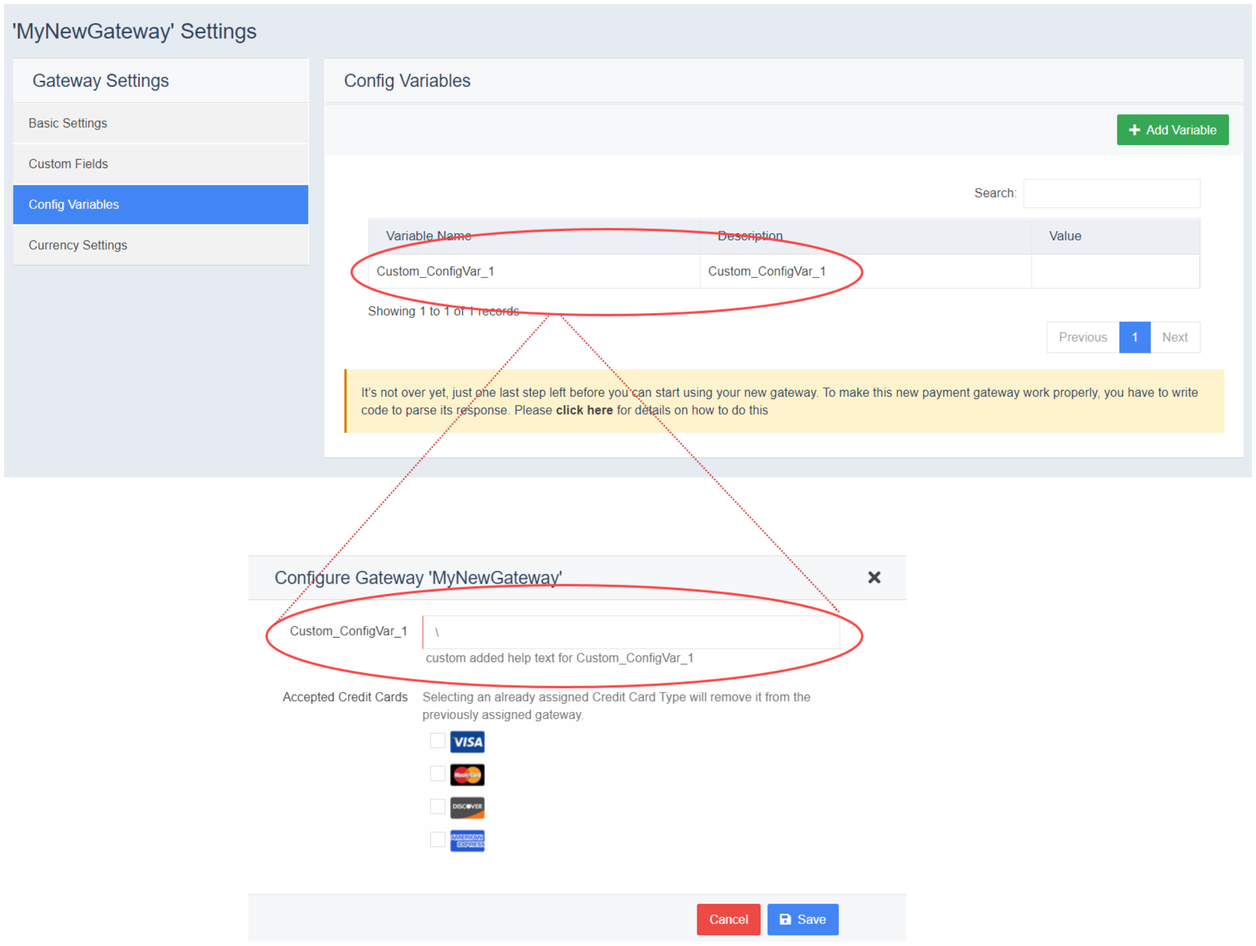
$40.00

Add Funds

| | |
|---|---|
| Select Billing Profile | Visa - Profile1 |
| Credit Card No. | ************1111 |
| CVV2 | e.g. 470 |
| Funds to Add | 0 $ |
| Custom_field1 | |

*Figure 5: Specifying custom fields.*

Once done, you can proceed to the next screen where you need to add Gateway configuration fields. These fields will be shown on Gateway configuration page mainly for Gateway credentials.



| Variable Name | Description | Value |
|---|---|---|
| Custom_ConfigVar_1 | Custom_ConfigVar_1 | |

Showing 1 to 1 of 1 records

It's not over yet, just one last step left before you can start using your new gateway. To make this new payment gateway work properly, you have to write code to parse its response. Please **click here** for details on how to do this

Custom_ConfigVar_1 \
custom added help text for Custom_ConfigVar_1

Selecting an already assigned Credit Card Type will remove it from the previously assigned gateway.

*Figure 6: Specifying configuration variables.*

## Part 2 (Backend Hook):

The second step is to add the backend code of the Gateway to process the payment requests. For backend-code you need to implement methods of an interface from an assembly "Marketplace.Hooks.Interfaces.dll" which is shipped with the Marketplace installation, and can be found in the "bin" directory. This assembly contains two interfaces related to payment Gateways; one each for the two types (Third-party, Merchant).

Download the skeleton project as per the type of your Gateway:

- ThirdPartyGateway.zip
- MerchantGateway.zip

Then perform the following steps to get the project ready:

a. Open the downloaded project in Visual Studio 2015. It's a class library type project.

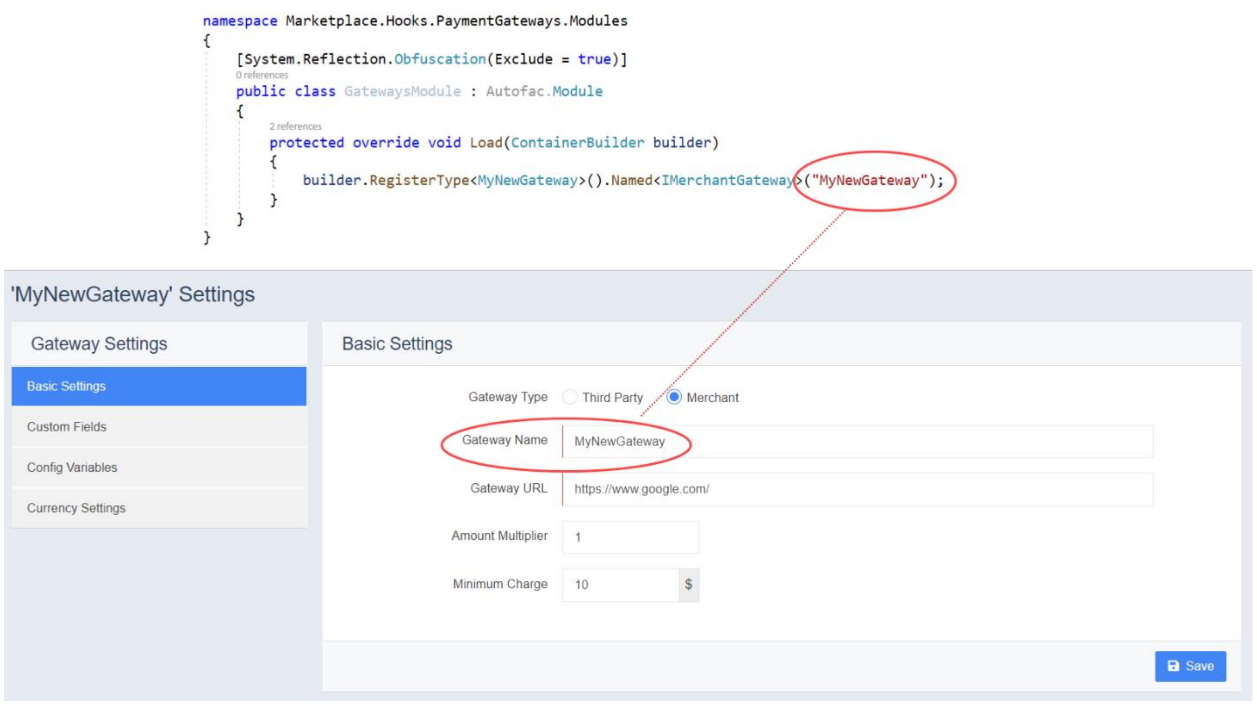
NOTE

Do not update the versions of any NuGet package already added in the skeleton project.

b. Add reference to the assembly “*Marketplace.Hooks.Interfaces.dll*” from “*bin”* folder in Marketplace Installation location and rebuild the project.

c. To give a name to your implementation of the provided interface, open “*Modules/GatewaysModule.cs*” file and update the component name (encircled string) with your new Gateway name.

   i. Name must be exactly the same as entered on the “*Add New Gateway*” form in the Marketplace.



```
namespace Marketplace.Hooks.PaymentGateways.Modules
{
    [System.Reflection.Obfuscation(Exclude = true)]
    0 references
    public class GatewaysModule : Autofac.Module
    {
        2 references
        protected override void Load(ContainerBuilder builder)
        {
            builder.RegisterType<MyNewGateway>().Named<IMerchantGateway>("MyNewGateway");
        }
    }
}
```

'MyNewGateway' Settings

Basic Settings

| | |
|---|---|
| Gateway Type | Third Party [✓ Merchant] |
| Gateway Name | MyNewGateway |
| Gateway URL | https://www.google.com/ |
| Amount Multiplier | 1 |
| Minimum Charge | 10 $ |

*Figure 7: Name your implementation of the provided interface.*

d. Open “MyNewGateway.cs” file which implements an interface (either IMerchantGateway or IThirdPartyGateway) and add your code as per requirements.

   i. You can change file or class name as per your requirement.

Following is the detail of the interfaces:

## 1. IMerchantGateway

This interface contains only a single method and is for Merchant Gateways. It receives user billing profile, its credit card detail, Gateway credentials and other details as JSON input parameter.

| Methods | | |
|---|---|---|
| PayNow | Input | see below |

```
{
  "UserProfile":{
    "UserId":0,
    "UserName":" string ",
    "FirstName":" string ",
    "LastName":" string ",
    "EmailAddress":" string ",
    "City":" string ",
    "State":" string ",
    "StateCode":" string ",
    "Country":" string ",
    "CountryCode":" string ",
    "AddressLine1":" string ",
    "AddressLine2":" string ",
    "ZipCode":" string ",
    "FaxNo":" string ",
    "PhoneNo":" string ",
    "Language":" string ",

    "CreditCardNo":" string ",
    "Cvv2":" string ",
    "ExpiryMonth":0,
    "ExpiryYear":0,
  },
  "Transaction":{
    "TransactionId":" guid ",
    "PaymentTypeName": "AUTH | CAPTURE | AUTHCAPTURE"
    "AmountToPay":0.00,
    "GatewayAmount":0.00,
    "CurrencyCode":" string "
  },
  "Gateway":{
    "Name":" string ",
    "GatewayUrl":" string ",
    "ConfVariables":[
      {
        "VariableName":" string ",
        "Value":" string "
      },
      {
        "VariableName":" string ",
        "Value":" string "
      },
      .
      .
      .
    ],
```

| | | |
|---|---|---|
| | | "CustomFields":[<br>{<br>"FieldName":" string ",<br>"Value":" string "<br>},<br>{<br>"FieldName":" string ",<br>"Value":" string "<br>},<br>.<br>.<br>.<br>]<br>}<br>} |
| | **Output** | {<br>"GatewayTransactionId": "string"<br>"Status":"*Approved* \| *Failed*",<br>"StatusDetail":" string ",<br>"GatewayResponse":" optional"<br>} |

Following image shows the payment process of Merchant Gateways and role of Hook in the process.

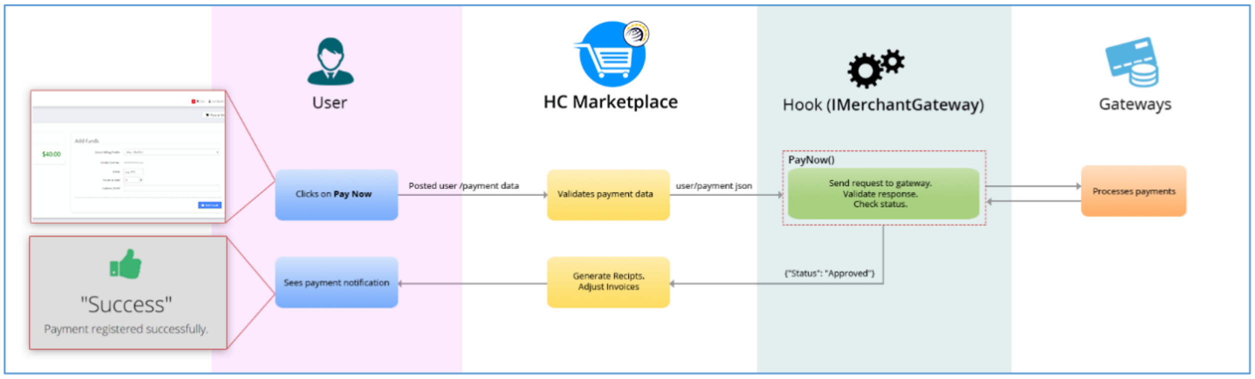


*Figure 8: How Marketplace, your implementation (Hook) and Merchant Gateway works.*

## 2. IThirdPartyGateway

This interface has two methods; one that returns a URL which HC uses to redirect user to the Gateway's website to make payment and, second, which receives the payment response from the Gateway after completion of payment and decides the payment status.

| Methods | | |
|---|---|---|
| ConstructRequestUrl | **Input** | {<br>"UserProfile":{<br>"UserId":0, |

```
    "UserName":" string ",
    "FirstName":" string ",
    "LastName":" string ",
    "EmailAddress":" string ",
    "City":" string ",
    "State":" string ",
    "StateCode":" string ",
    "Country":" string ",
    "CountryCode":" string ",
    "AddressLine1":" string ",
    "AddressLine2":" string ",
    "ZipCode":" string ",
    "FaxNo":" string ",
    "PhoneNo":" string ",
    "Language":" string "
  },
  "Transaction":{
    "TransactionId":" guid ",
    "AmountToPay":0.00,
    "GatewayAmount":0.00,
    "CurrencyCode":" string "
  },
  "Gateway":{
    "Name":" string ",
    "GatewayUrl":" string ",
    "NotifyUrl":" string ",
    "ReturnUrl":" string ",
    "ConfVariables":[
      {
        "VariableName":" string ",
        "Value":" string "
      },
      {
        "VariableName":" string ",
        "Value":" string "
      },
      .
      .
      .
    ],
    "CustomFields":[
      {
        "FieldName":" string ",
        "Value":" string "
      },
      {
        "FieldName":" string ",
        "Value":" string "
      },
      .
      .
      .
    ]
```

| | | |
|---|---|---|
| | | }<br>} .<br>]<br>}<br>}<br><br>Note: “NotifyUrl and “ReturnUrl” properties contain two URL values which have to be used in the constructed “RedirectionUrl” in the appropriate URL variables (as mentioned by Gateway)<br>e.g for PayPal<br>…&return={ReturnUrl}&notify_url={NotifyUrl}…. |
| | Output | {<br>"RedirectionUrl":" string "<br>} |
| ParseResponse | Input | Param1<br>Serialized string of data posted by the Gateway to our notify URL e.g. payer_id=43EDT6MZJM7L4&payer_status=VERIFIED&first_name=mark&last_name=jhon&address_name=….. |
| | | Param2<br>{<br>"UserProfile":{<br>"UserId":0,<br>"UserName":" string ",<br>"FirstName":" string ",<br>"LastName":" string ",<br>"EmailAddress":" string ",<br>"City":" string ",<br>"State":" string ",<br>"StateCode":" string ",<br>"Country":" string ",<br>"CountryCode":" string ",<br>"AddressLine1":" string ",<br>"AddressLine2":" string ",<br>"ZipCode":" string ",<br>"FaxNo":" string ",<br>"PhoneNo":" string ",<br>"Language":" string "<br>},<br>"Transaction":{<br>"TransactionId":" guid ",<br>"AmountToPay":0.00,<br>"GatewayAmount":0.00,<br>"CurrencyCode":" string "<br>},<br>"Gateway":{ |

| | | |
|---|---|---|
| | | "**Name**":" string ", "**GatewayUrl**":" string ", "**ConfVariables**":[ { "**VariableName**":" string ", "**Value**":" string " }, { "**VariableName**":" string ", "**Value**":" string " }, . . . ], "**CustomFields**":[ { "**FieldName**":" string ", "**Value**":" string " }, { "**FieldName**":" string ", "**Value**":" string " }, . . . ] } } |
| | Output | { "**Status**":"*Approved \| Failed*", "**StatusDetail**":" string ", "**GatewayResponse**":" optional" } |

Following image shows the payment process of Third-party Gateways and role of Hook in the process.

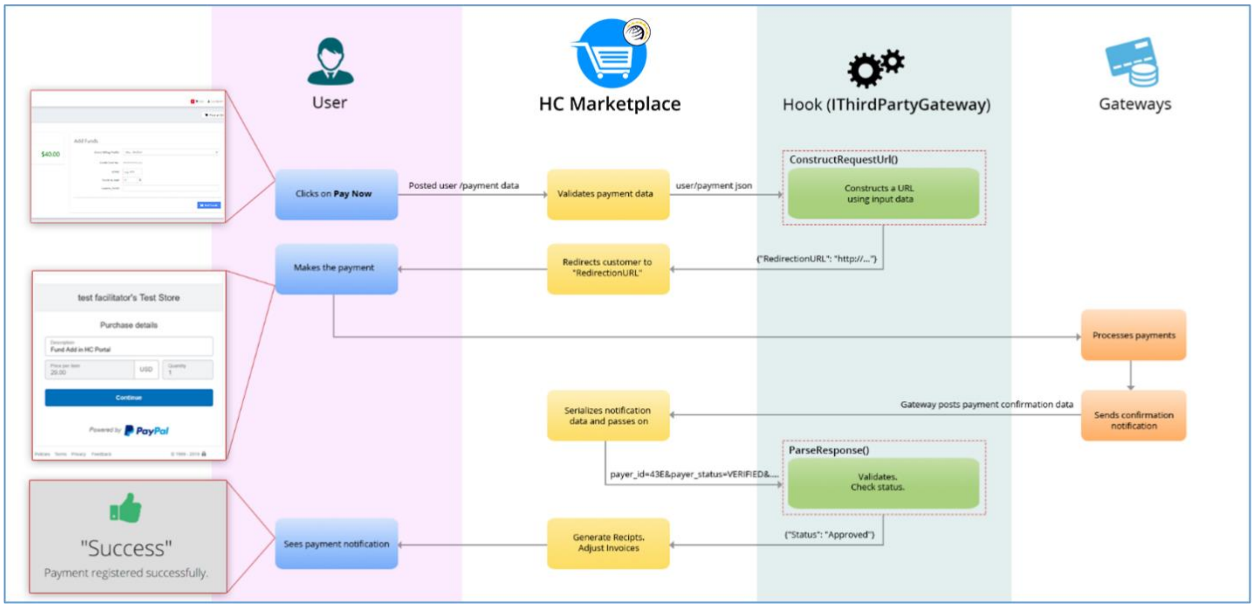

User
HC Marketplace
Hook (IThirdPartyGateway)
Gateways
Clicks on **Pay Now**
Posted user /payment data
Validates payment data
user/payment json
ConstructRequestUrl()
Constructs a URL using input data
{"RedirectionURL": "http://..."}
Redirects customer to "RedirectionURL"
Makes the payment
Processes payments
Sends confirmation notification
Gateway posts payment confirmation data
Serializes notification data and passes on
payer_id=43E&payer_status=VERIFIED&....
ParseResponse()
Validates. Check status.
{"Status": "Approved"}
Generate Recipts. Adjust Invoices
Sees payment notification
test facilitator's Test Store
Purchase details
Fund Add in HC Portal
Continue
Powered by PayPal
"Success"
Payment registered successfully.

*Figure 9: How Marketplace, your implementation (Hook) and Third-party Gateway works.*

Once done, just build the project and place the output assembly in the *"bin"* directory of Marketplace. That's it, restart Marketplace app pool and you are good to test your new integration.

NOTE Your assembly name must start with "Hook" e.g HookMyNewGateway.dll.

## Contact Us

In case of any ambiguity/query regarding integration of payment Gateways, please feel free to contact us at support@hostingcontroller.com.