Version 10.17.0

PROTECTPAY® PAYER MANAGEMENT INTERFACE: HOSTED PAYMENT PAGE (HPP)

Instructions to Interface with ProPay ProtectPay Payer Management Interfaces.

PROPAY
A TSYS® Company

Revision History:

Date	Version	Description	Author(s)
09/04/2017	9.17.0	Added technical implementation of HPP V2.	Kalyan Kapratwar Avinash Avate

Contents

1.0 ProPay® ProtectPay® Application Programming Interface

The ProtectPay Payer Management Interface: Hosted Payment Page is a Payer Management Interface (PMI) that allows merchants to maintain a payment page that mirrors the look and feel of their website without storing, transmitting or processing the data that their payment pages collect. The Hosted Payment Page enables a merchant to collect and process sensitive payment method information on a ProtectPay by redirecting them to hosted payment page.

How to use this manual

This manual is designed to facilitate developers in building software solutions to consume the Hosted Payments Page. It is not written for a single development platform. It provides basic information required to properly interact with the Hosted Payments Page.

A developer should have an understanding of Hyper Text Transfer Protocol (HTTP) communication, the consuming of external Web services, Web Form POST methodology, JavaScript, jQuery, and creating a Transport Layer Security (TLS) connection on the intended development platform.

While ProPay offers resources and materials that assist in creating and developing software solutions it is the responsibility of the integrating developer to design and develop his or her own software solution on the intended development platform to make use of and consume the services offered by ProPay.

For additional development resources please visit: https://developer.propay.com

Additional Resources

See ProtectPay API Manual for API Methods referenced in this manual.

Important Concepts

- ProtectPay is not a Processor or Gateway; it is a secure collection of sensitive payment data.
- ProtectPay stores data securely for both single and recurring or subsequent payments using industry best practices.
- ProtectPay utilizes a proprietary interface to process transactions through several major gateways, processors and services providers.
- The Hosted Payment Page will not return the results of a transaction; an additional API call is required to get the results of the transaction.

Disclaimer

ProPay provides the following documentation on an "AS IS" basis without warranty of any kind. ProPay does not represent or warrant that ProPay's website or the API will operate securely or without interruption. ProPay further disclaims any representation or warranty as to the performance or any results that may be obtained through use of the API.

Updated API manu

1.1 Description of the Hosted Payment Page

The Hosted Payment Page (HPP) is a Payer Management Interface (PMI) of the ProtectPay Application Programming Interface (API). ProtectPay ensures the payers' payment information is collected, updated, and stored in accordance with PCI standards. The Hosted Payment Page is an HTML5 page hosted by a secure ProtectPay server. The merchant's checkout page will be redirected to this page for secure payment. The Hosted Payment Page enables a merchant to collect sensitive payment method information without having it traverse the merchant's system. This minimizes the merchants PCI compliance requirements and limits the risk and exposure of the merchant by not handling sensitive payment information.

Why the Hosted Payment Page

The integration to most fully remove merchants from PCI scope is the Hosted Payment Page. The user can redirect to the Hosted Payment Page from the merchant's checkout page to perform the transaction. In this way there is no possibility of a merchant being able to pass any sensitive payment information back to their systems. The Hosted Payment Page is only needed when new payment method information must be collected. The Hosted Payment Page can be configured to create a PaymentMethodId from the payer entered data. Once a PaymentMethodId has been created for the specified PayerId it can be processed against using the ProtectPay API directly while maintaining minimal risk, exposure and PCI compliance scope.

Hosted Payment Page Processing Configurations

The Hosted Payment Page can be configured to perform various payment method storage and/or processing requests.
These include:

- Create a PaymentMethodId
- Create and Authorize a PaymentMethodId for a specified amount
- Create and Process a PaymentMethodId for a specified amount
- Authorize a payment method for a specified amount
- Process a payment method for a specified amount
- Authorize a payment method for a specified amount and create a PaymentMethodId only if successful

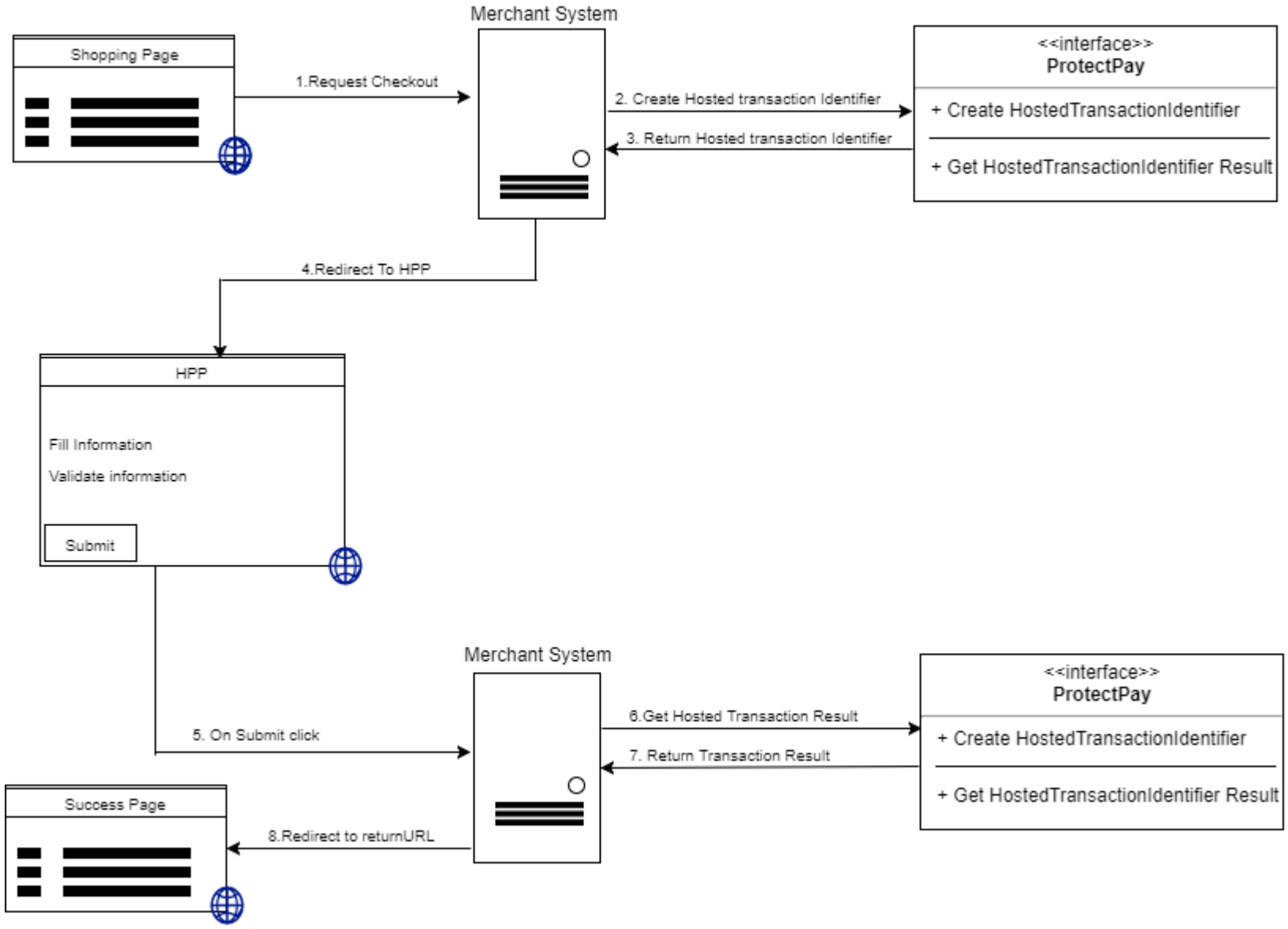
The values required for each configuration are determined by the type of transaction requested.

1.2 Processing with the Hosted Payment Page

Summary of Processing

1. Customer requests checkout from the merchant's system.
2. The merchant's system creates a HostedTransactionIdentifier using ProtectPay API Method 4.7.2 'Create HostedTransactionIdentifier'.
3. The ProtectPay API returns the HostedTransactionIdentifier.
4. The merchant's system redirect to HPP.
   The payer fills in payment method information and then clicks the "submit" button
5. The Hosted Payment Page validates the entered information and if passes processes the request.
   If the information does not pass validation the form is not submitted and the form displays the error to the user to correct the information entered.
6. The merchants system is notified of the successful submission of the Hosted Payment Page by redirecting to ReturnURL.
   The merchants system gets the results of the Hosted Transaction using ProtectPay API Method 4.7.3 'Get Hosted Transaction Results'.
7. The ProtectPay API returns the results of the Hosted Transaction.
8. The merchant's system redirect to the ReturnURL with "result" querystring. ReturnURL is specified while creating HostedTransactionIdentifier.
   (Eg: if ReturnURL is Http://www.abc.com then on success redirect it to Http://www.abc.com/?result=success and on failure redirect it to Http://www.abc.com/?result=failure&message=errormessage)

Summary of Processing Flow Diagram



2.0 Interface Testing and Certification

To improve the customer experience, ProPay requires that new developers test their software solutions before receiving credentials to process live transactions. This integration process is designed to assist the developer in building a robust solution that can handle and process all the various responses that come from real time credit card and ACH processing. This process ultimately improves the end-user experience. Please plan accordingly when developing timelines and schedules to accommodate for testing against the ProPay Integration environment. Negotiated fees are not refunded in the production environment.

Regardless of its cause, ProPay will not be liable to client for any direct, indirect, special, incidental, or consequential damages or lost profits arising out of or in connection with client's use of this documentation, even if ProPay is advised of the possibility of such damages. Please be advised that this limitation applies whether the damage is caused by the system client uses to connect to the ProPay services or by the ProPay services themselves.

Integrating a developed software solution to the ProPay web integration requires the following steps:

1. Request from a ProPay sales representative and/or account manager that integration API credentials be sent.
2. Begin interfacing the appropriate ProtectPay API methods for specific project scope.
   - ❖ A ProPay sales representative and/or account manager will help determine which methods are required for the specific project scope.
3. Design, develop, build and test the software solution using the ProtectPay integration environment.
   a. The ProtectPay integration Hosted Payment Page base URI: https://protectpaytest.propay.com/hpp/v2/
   - ❖ Testing payment processing requires additional information provided by the client's chosen processor.
4. Certify the developed software solution against the ProPay integration environment.
   *Review the status of the integration certification with a ProPay sales representative and/or account manager.
5. Request Production (Live) Credentials from a ProPay sales representative and/or account manager.

   - ❖ Live Credentials MUST be kept confidential

For additional information about ProPay testing and live environments see: ProPay Server Environments.

3.0 Technical Implementation

Required ProtectPay API Methods:

- ProtectPay API Method 4.2.1 'Create PayerId'
- ProtectPay API Method 4.7.2 'Create HostedTransactionIdentifier'
- ProtectPay API Method 4.7.3 'Get Hosted Transaction Results'

Web Browser support:

Desktop Browser	Version
Internet Explorer	10+
Microsoft Edge	13+
Fire Fox	44+
Safari	9+
Google Chrome	47+
Opera	36+
Chromium	47+

Mobile Browser	Version
Safari	8.4+
Android Browser	4.4+
Blackberry Browser	10+
Opera Mobile	12+
Chrome	49+
Fire Fox	45+
IE	11+
UC	9.9+

- This list is not exhaustive and only covers major browsers

Transport Layer Security (TLS):

ProPay recognizes the importance of handling financial transactions in a secure manner and ensures that ProtectPay offers the best transmission security available. ProPay ensures that ProtectPay API request information is transmitted using the latest Transport Layer Security (TLS) encryption practices. TLS creates a secure connection between client and server over which encrypted information is sent. ProPay hosts the SSL certificate for this connection type. Each ProtectPay API method request, regardless of the interface, will negotiate an SSL connection automatically over port 443.

CSS3:

The Hosted Payment Page can be customized to match the look at feel of a merchant's checkout page. One of the parameters passed when creating a HostedTransactionId is the 'CSSUrl'. If passed the Hosted Payment Page will appear by referencing this CSS file, the file must be publically accessible to the Hosted Payment Page. If the Hosted Payment Page is unable to reach the CSS URL or there was no URL passed the Hosted Payment Page will load with the default style. ProPay offers both a style guide and Hosted Payment Page HTML file to assist developers in styling the Hosted Payment Page to the merchant's specifications.

- See Section 3.2.4 Customization of the Hosted Payment Page for additional information.

Return Response: Once the payment done successfully Hosted Payment Page redirect to the ReturnURL. ReturnURL is one of the parameter passed while creating a HostedTransactionId. If the ReturnURL is not present then transaction will not be performed and merchant will get an error message.

The return response will be received in following formats. (e.g. ReturnURL = www.abc.com)
On Success: ReturnURL+ ?result=success (e.g. www.abc.com?result=success)
On Cancel: ReturnURL+ ?result=cancel (e.g. www.abc.com?result=cancel)
On Failure: ReturnURL+ ?result=failure&message=error message (e.g. www.abc.com?result=failure&message=errormessage)

ACH Processing:

The Hosted Payment Page can be used for ACH processing. ACH processing is limited to the Legacy ProPay processor and the receiving ProPay account must be ACH payment enabled with an approved and set SEC code of 'WEB'. Attempts to process against another processor will result in an error being returned after submission. Attempts to process against a ProPay merchant or business account that is not setup for WEB enabled ACH payments will result in a processor decline of 'Invalid Standard Class Entry' when using ProtectPay API method 4.7.3 'Get Hosted Transaction Results'.

'Name' is a required field for all ACH transaction requests and will be displayed as required independent of the submitted HostedTransactionIdentifier settings as well as AVS information being hidden.

3.1 Best Practices

- A PayerId is required when creating a PaymentMethodId. A PayerId can be created using either ProtectPay API method 4.2.1 'Create PayerId' or by using ProtectPay API method 4.7.1 'Create TempToken'. Once a PayerId is created it should be used in subsequent transaction requests instead of creating a new one for each transaction.

- The HPP is only required when creating a PaymentMethodId, or processing payment method information without wanting to store it. Once a PaymentMethodId is created it should be processed against directly using the ProtectPay API.

- The Hosted Payment Page will perform card number validation against a Mod 10 using the LUHN algorithm, CVV length validation, card expiration date validation and card number for card type validation upon submission. If an error is detected the form will raise a submission error for the merchants checkout page to handle and display an error on the Hosted Payment Page.

- Credit card transactions can take several seconds to process. This is caused by several variables with the gateway, the processor, and the issuer. There will be a wait during which a cardholder may become impatient. ProPay has provided a warning against clicking the back button, or refresh while a payment method is processing.

- Do not create a HostedTransactionIdentifier with the Name or AVS fields set to required and then use CSS to hide them. This will create undesired effects.

- Do not create a HostedTransactionIdentifier without ReturnURl. It will raise an error and does not allow to process the transaction.

Recommended API Request Best Practices Use Cases

1. Merchant known payer wants to use a merchant stored payment method
    a. Use ProtectPay API method 4.4.1 'Authorize a PaymentMethodId' or 4.5.1 'Process a PaymentMethodId'
2. Merchant known payer wants to add an additional stored payment method to merchant system
    a. Use ProtectPay API method 4.7.1 'Create TempToken' passing in known PayerId as parameter
    b. Set HPP parameter to store payment method
    c. Set HPP parameter to process payment method or not process payment method
3. Merchant unknown payer wants to process a payment method for future use
    a. Use ProtectPay API method 4.7.1 'Create TempToken' passing in unknown payer name as parameter
    b. Set HPP parameter to store payment method
    c. Set HPP parameter to process payment method or not process payment method
4. Merchant known or unknown payer wants to process a payment method and not store it for future use
    a. Use ProtectPay API method 4.7.1 'Create TempToken' with appropriate parameters
    b. Set HPP Parameter to not store payment method
    c. Set HPP parameter to process payment method

Checkout Process Recommendation

1. Pre-Checkout Page
    a. Client collects non-sensitive information
    b. Client performs special functions for pre-loading such as ZIP lookups
    c. Client creates a PayerID, if neccesary, and then proceeds to create HostedTransactionIdentifier
2. Checkout Page
    a. Client redirected to the Hosted Payment Page
    b. Customer fills out payment information
    c. Client will be redirected to ReturnURL with the response on completion of the transaction and uses ProtectPay API method 4.7.3 Get Hosted Transaction Results.
3. Post Checkout Page
    a. Client interprets results of Hosted Transaction
    b. Client presents results to customer or client performs error correction and handling and creates new Hosted Transaction Identifier

3.2 Interface

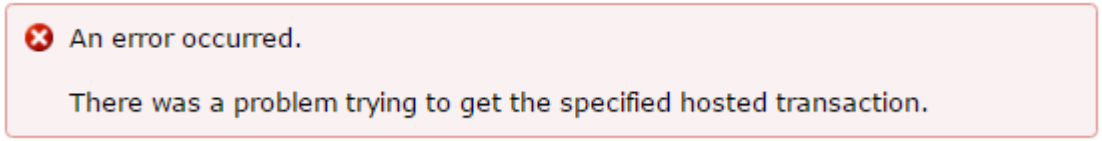
The merchant is redirected to Hosted Payment Page, as soon as page gets loaded and filled with required information it is ready to submit.

Invalid HostedTransactionIdentifier Errors

If an invalid HostedTransactionIdentifier is used to retrieve the Hosted Payment Page the following error will be displayed by the Hosted Payment Page:

Using an Already Used HostedTransactionIdentifier Error

A HostedTransactionIdentifier is a onetime use token. If in the event a client attempts to use a HostedTransactionIdentifier that has already been used the following error will be displayed by the Hosted Payment Page:



Submission of Hosted Payment Page when ReturnURL is not present

If the RetrunURL is not specified while creating HostedTransactionIdentifier following error message will be displayed by the Hosted Payment Page:

Successful submission of the Hosted Payment Page

The Hosted Payment Page, by design, will redirect to the 'ReturnURL' with query string including result as success, cancel or failure of the requested transaction type. If the response is failure the query string will also have the error message in it. For more details see section of 3.0 -> Return Responses

Please see section 3.1 for best practices during the submission of the Hosted Payment Page for additional recommendations.

3.2.1 Hosted Payment Page Configurations

The Hosted Payment Page configuration is set when creating the HostedTransactionIdentifier.

Store Payment Method Only

In order to use the Hosted Payment Page to store payment methods only please set the following attributes when creating the HostedTransactionIdentifier. The Hosted Payment Page will store the payment method information and the results will return the PaymentMethodId under the PayerId that is used to create the HostedTransactionIdentifier and once stored the PaymentMethodId can be processed against directly using the ProtectPay API.

Attribute	Value
StoreCard	True
AuthOnly	False
ProcessCard	False
OnlyStoreCardOnSuccessfulProcess	False

Validating Card Data prior to storage

In order to validate a card is valid prior to storing ProPay recommends the Hosted Payment Page be configured to authorize a card and store only on success.

Attribute	Value
StoreCard	False
AuthOnly	True
ProcessCard	False
OnlyStoreCardOnSuccessfulProcess	True

- Best Practice: the minimum authorization amount is 1.00 [Currency]. ProPay recommends a client authorize 1.00 [Currency] (submitted as 100 [currency]) to validate a card and store it only if successful and then immediately void the transaction

Authorize Payment Method Only

In order to use the Hosted Payment Page to authorize a payment method only and not store it please set the following attributes when creating the HostedTransactionIdentifier. The Hosted Payment Page will attempt to authorize the card and the result will return the result of the transaction only.

Attribute	Value
StoreCard	False
AuthOnly	True
ProcessCard	False
OnlyStoreCardOnSuccessfulProcess	False

Process Payment Method Only

In order to use the Hosted Payment Page to process a payment method only and not store it please set the following attributes when creating the HostedTransactionIdentifier. The Hosted Payment Page will attempt to process the card (Auth or Auth and Capture) and the result will return the result of the transaction only.

Attribute	Value
StoreCard	False
AuthOnly	False
ProcessCard	True
OnlyStoreCardOnSuccessfulProcess	False

Process and Store Payment Method

In order to use the Hosted Payment Page to process a payment method and store the payment method information please set the following attributes when creating the HostedTransactionIdentifier. The Hosted Payment Page will attempt to process the payment method (Auth or Auth and Capture) and also store the payment method. The results will return both the results of the process attempt and the PaymentMethodId

Attribute	Value
StoreCard	True
AuthOnly	False
ProcessCard	True
OnlyStoreCardOnSuccessfulProcess	False

Process and Store Payment Method only if the Payment Method Processed Successfully

In order to use the Hosted Payment Page to process a payment method and store the payment method information only if the payment method was successfully processed (Auth or Auth and Capture) please set the following attributes when creating the HostedTransactionIdentifier. The Hosted Payment Page will attempt to process the payment method (Auth or Auth and Capture) and only store the payment method if the process request was successful. The results will return both the results of the process attempt and the PaymentMethodId only if the process request was successful.

Attribute	Value
StoreCard	False
AuthOnly	False
ProcessCard	True
OnlyStoreCardOnSuccessfulProcess	True

Custom Preloading of Non-Sensitive Information

In order to provide an enhanced customer user experience, the Hosted Payment Page can load with non-sensitive payment information prefilled. This is useful in the following cases:

1. Merchant known payer wants to add an additional stored payment method to merchant system.
2. The Hosted Payment Page was unable to process the payment method.

The following fields can be prefilled into the Hosted Payment Page.

- Payer Name
- Payment Method Description
- Address 1
- Address 2
- City
- State
- Postal Code
- Country

In order to preload the Payer Name field the **CardHolderNameRequirementType** must be set to 1 or 2 when creating the HostedTransactionIdentifier. In order to preload the Address 1, Address 2, City, State, Postal Code and Country fields the **AvsRequirementType** must be set to 1 or 2 when creating the HostedTransactionIdentifier.

In the case of a payment method processing failure the ProtectPay API method 4.7.3 'Get Hosted Transaction Results' will return all the non-sensitive payment method information it was configured to collect. In this case the information returned can be used to preload a new Hosted Payment Page. The following chart indicates which response attribute returned should be passed to the appropriate attribute in ProtectPay API method 4.7.2 'Create HostedTransactionIdentifier'.

Response – Create Attribute Correlation

Hosted Transaction Result Attribute	Create HostedTransactionIdentifier Attribute
PaymentMethodInfo.AccountName	Name
PaymentMethodInfo.Description	Description
PaymentMethodInfo.BillingInformation.Address1	Address1
PaymentMethodInfo.BillingInformation.Address2	Address2
PaymentMethodInfo.BillingInformation.City	City
PaymentMethodInfo.BillingInformation.State	State
PaymentMethodInfo.BillingInformation.ZipCode	ZipCode
PaymentMethodInfo.BillingInformation.Country	Country

3.2.2 Customization of the Hosted Payment Page

The Hosted Payment Page can be styled to match a merchant's checkout page. A custom CSS can be used to replace the default CSS when the merchant redirected to Hosted Payment Page. ProPay offers a sample HTML file and the default CSS file to assist in styling the Hosted Payment Page. The custom CSS file must be publically accessible and declared when creating the HostedTransactionIdentifier.

Default CSS File

The following CSS file is provided to assist in creating custom styles for the Hosted Payment Page. A custom CSS file URI must be declared when creating the HostedTransactionIdentifier and it must be publically accessible to ProtectPay.

- The following CSS URI does not have to be declared when creating a HostedTransactionIdentifer as the Hosted Payment Page will automatically use it if one is not supplied.

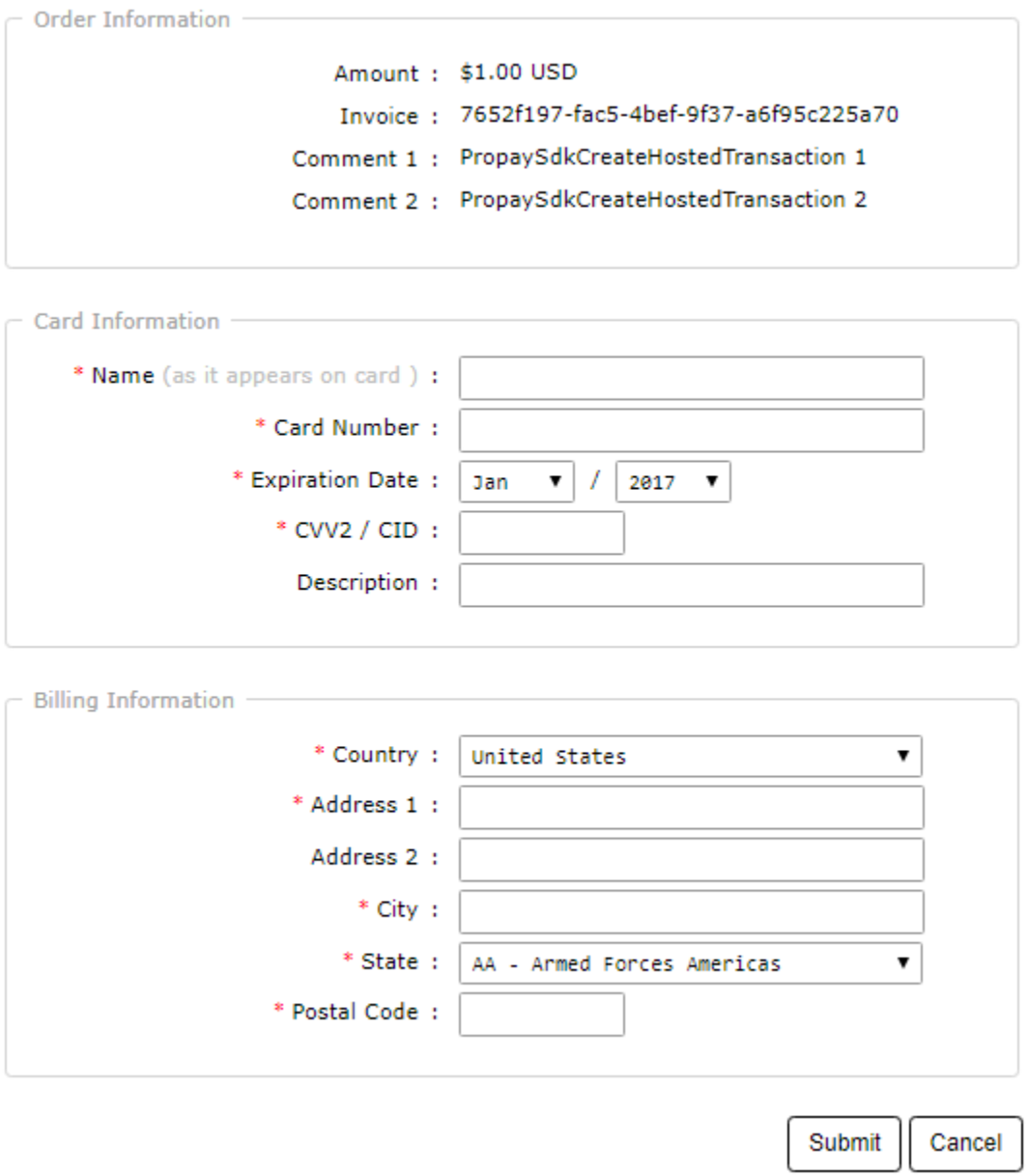
The ProtectPay Integration environment default CSS file: https://protectpaytest.propay.com/hpp/css/pmi.css

Notes on elements' styling

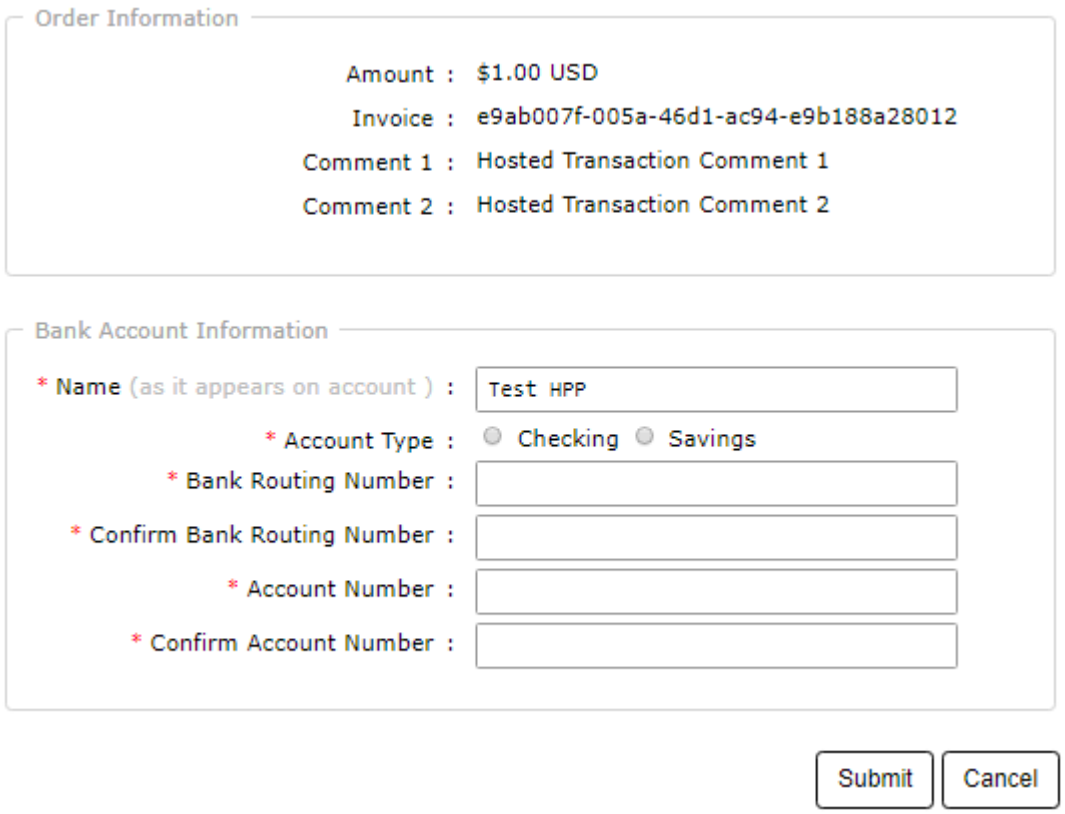
The following elements may not be immediately obvious and warrant extra explanation in how they impact styling.

.ErrorBackground - Used to provide highlighting to one of the ProPay error messages WHEN ErrorMessage is Displayed
.AcceptOnlyNumbers - Class applicable to multiple elements. Enables a check that data entered allows only for numeric characters.
.field-validation-error { display: table } - Text included when a validation error occurs underneath the Text Entry
.validation-summary-errors { display: none } -
.validation-summary-valid { display: none } -

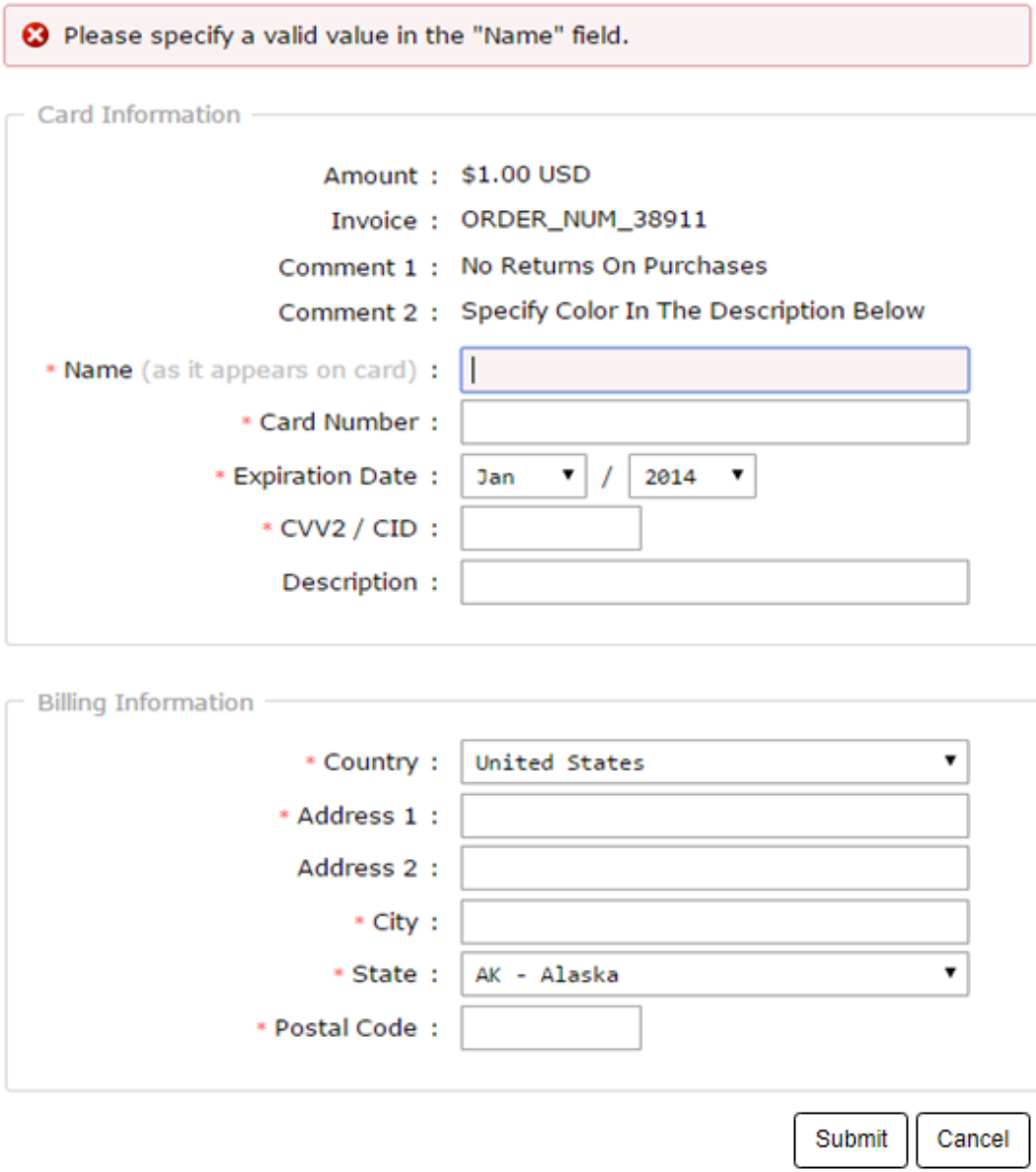
Default Appearance of the Hosted Payment Page (Credit Card Transactions)



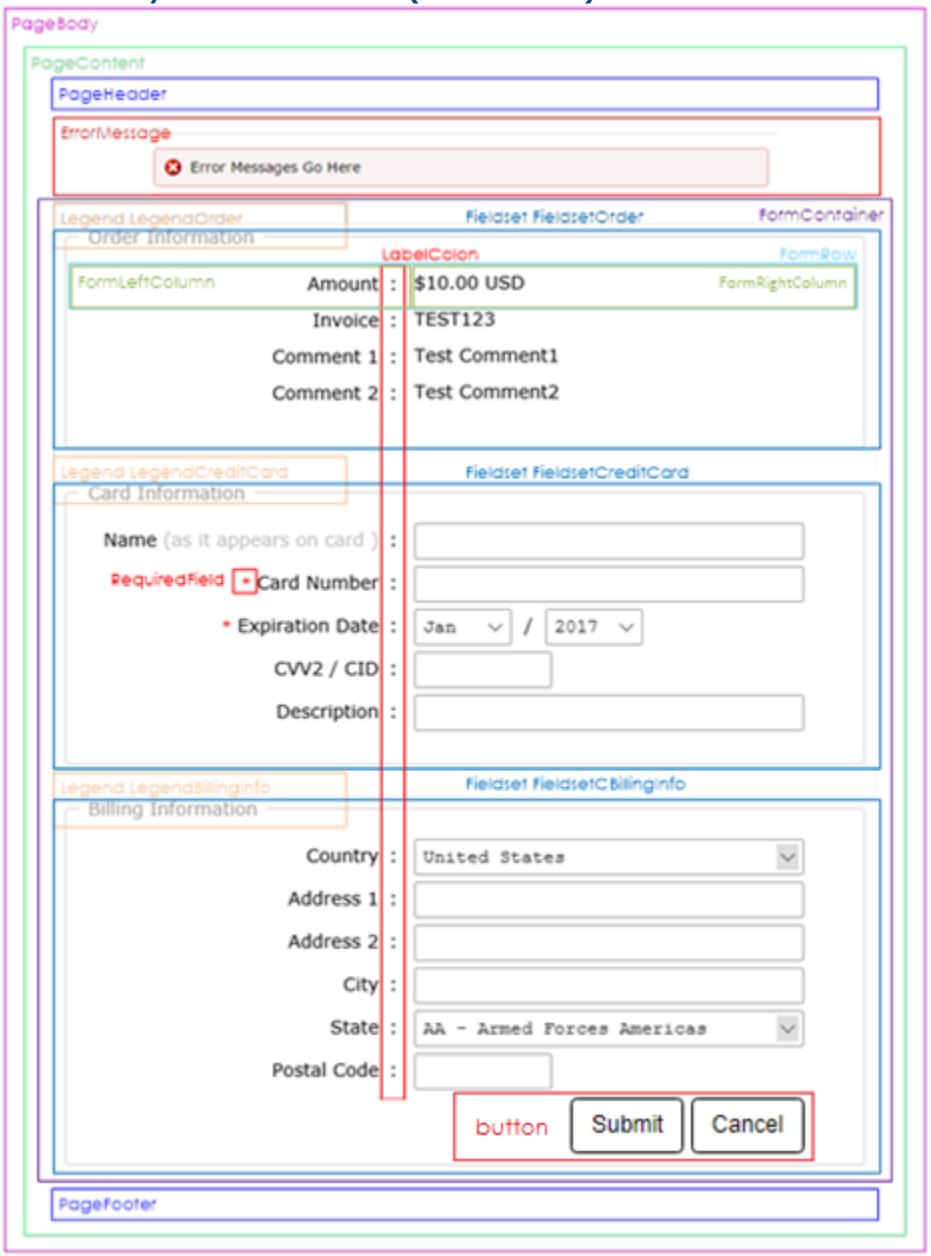
Default Appearance of the Hosted Payment Page (ACH Transactions)



Default appearance of HPP when failing form validation (Error Style is Identical for both Credit Card and ACH)

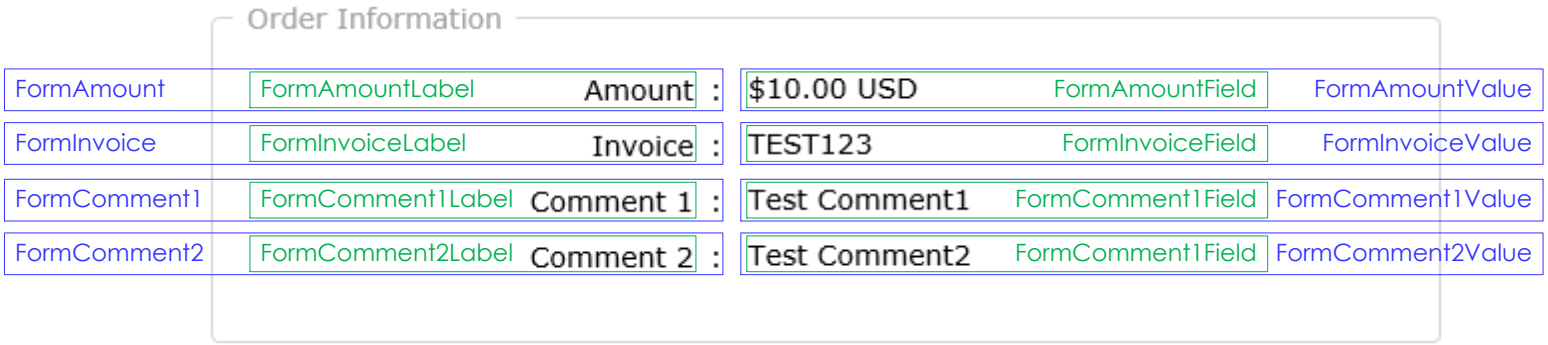


HPP CSS Style Section Guide (Credit Card)



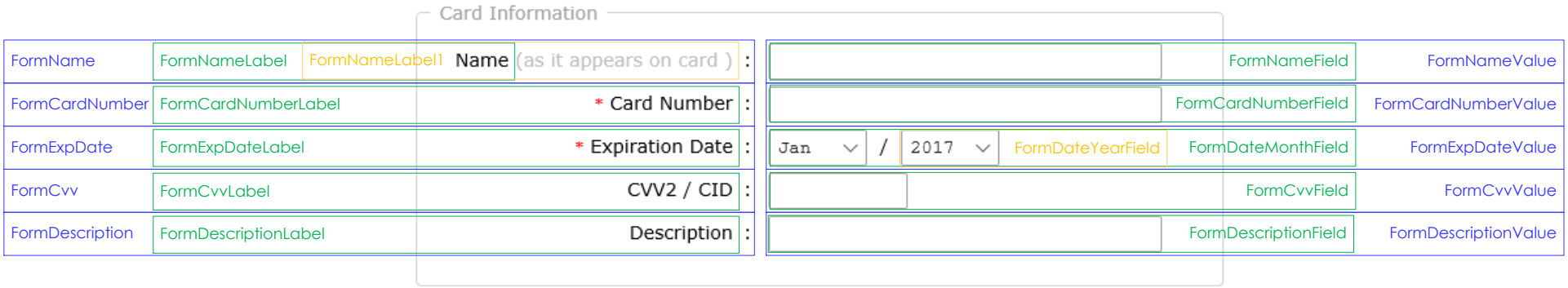
- Bank Account Information inherits LegendCreditCard and FieldsetCreditCard for simplicity of CSS for both payment types

HPP CSS Style Element Guide (Order Information)

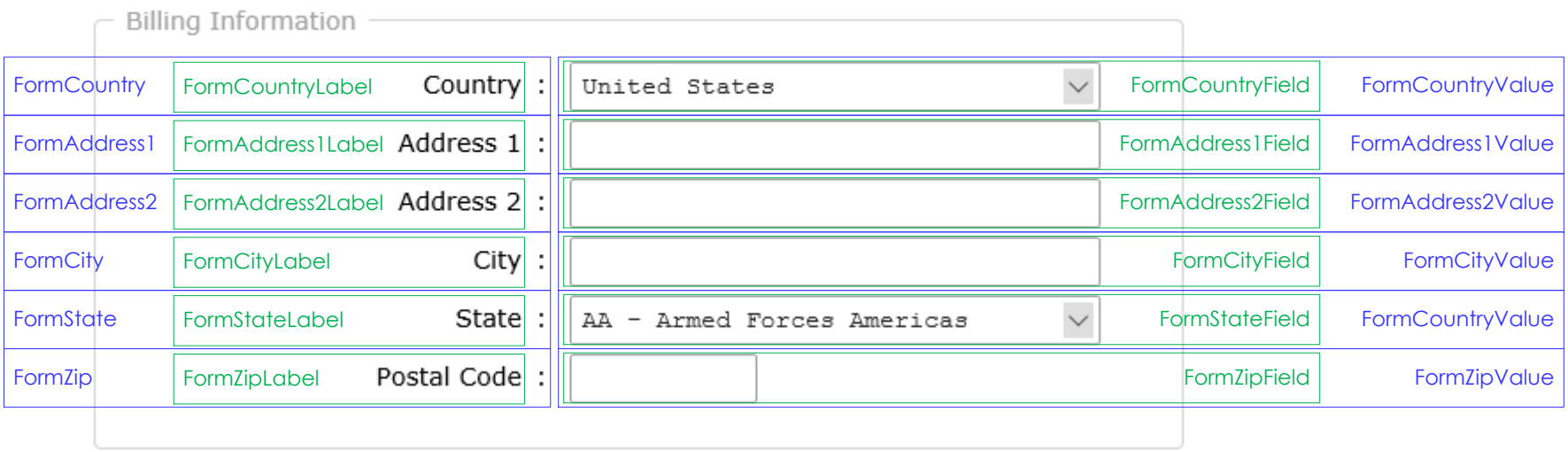


- All Labels inherit the "Label" class

HPP CSS Style Element Guide (Credit Card Information)

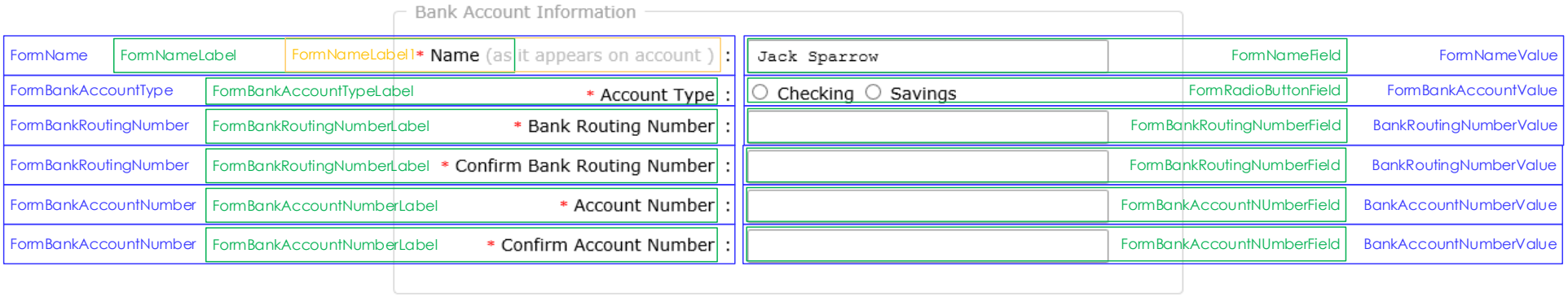


- All Labels inherit the "Label" class
- All TextFields inherit the "TextBox" Class
- All Drop Down Lists inherit the "DropDownClass"

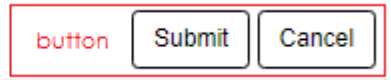


- All Labels inherit the "Label" class
- All TextFields inherit the "TextBox" Class
- All Drop Down Lists inherit the "DropDownClass"

HPP CSS Style Element Guide (ACH Information)



- All Labels inherit the “Label” class
- All TextFields inherit the “TextBox” Class
- All Radio Buttons Down Lists inherit the “RadioButton” Class



- All Buttons inherit the “button” class

Mobile Viewport CSS Customization Example

The following CSS snippet will help a developer to be able to change the default sizes of elements of the Hosted Payment Page for mobile view ports.

```
/*ProPay provides the following code “AS IS.”
*ProPay makes no warranties and ProPay disclaims all warranties and conditions, express, implied or statutory, including without limitation the implied warranties of title, non-infringement, merchantability, and fitness for a particular purpose.
*ProPay does not warrant that the code will be uninterrupted or error free, nor does ProPay make any warranty as to the performance or any results that may be obtained by use of the code.*/

.FormContainer {
    width: 570px;
    box-sizing: border-box;
    background-color: #ffffff;
    padding: 20px 20px;
    margin-top: 0px;
    margin-bottom: 35px;
    max-width: 1000px;
    width: 100%;
    margin: 0 auto;
    text-align: center;
    font: bold 14px sans-serif;
}

.FormContainer .FormRow {
    text-align: left;
    margin-bottom: 10px;
}

.FormContainer FormLeftColumn {
    text-align: center;
    margin-bottom: 10px;
}

.FormContainer .FormNameLabel {
    display: inline-block;
    text-align: left;
    padding: 0 0 10px;
}

.FormContainer input {
    color:  #5f5f5f;
```

```
    box-sizing: border-box;
    width: 200px;
    box-shadow: 1px 2px 4px 0 rgba(0, 0, 0, 0.08);
    padding: 5px 5px;
    border: 1px solid #dbdbdb;
}

.FormContainer select {
    background-color: #ffffff;
    color:  #5f5f5f;
    box-sizing: border-box;
    max-width: 240px;
    box-shadow: 1px 2px 4px 0 rgba(0, 0, 0, 0.08);
    padding: 12px 8px;
    border: 1px solid #dbdbdb;
}

.FormLeftColumn {
         font-weight: bold;
    display: inline-block;
         width: 49%;
}

.FormRightColumn {
    display: inline-block;
         width: 49%;
         text-align: right;
}

.FieldSet {
         margin-top: 20px;
         padding-top: 20px;
}

.TextBox {
         -webkit-appearance: none;
         -webkit-border-radius: 0;
         box-sizing: border-box;
         border: 1px solid #dfe3e9;
         padding: 0 .7em;
         border-radius: 4px;
         font-size: 13px;
```

```
        height: 33px;
        line-height: 33px;
        box-shadow: none;
        outline: none;
        color: #354052;
        width: 100%;
}

.FormLeftColumn {
        font-weight: bold;
        font: 400 16px/1.2 "proxima-nova", sans-serif;
        color: #7f8fa4;
        font-size: 14px;
        line-height: 1.2;
        font-weight: regular;
}

.FieldSet {

        border: 0px;
        margin-top: 20px;
        padding-top: 20px;
}

.Legend{
        font-weight: bold;
        font: 600 18px/1.4 "proxima-nova", sans-serif;
        color: #7f8fa4;
}
```

Language Customization CSS Example

The following CSS snippet will help a developer to be able to change the default language of the Labels in the Hosted Payment Page to various languages.

```
/*ProPay provides the following code “AS IS.”
*ProPay makes no warranties and ProPay disclaims all warranties and conditions, express, implied or statutory, including without limitation the implied warranties of title, non-infringement, merchantability, and fitness for a particular purpose.
*ProPay does not warrant that the code will be uninterrupted or error free, nor does ProPay make any warranty as to the performance or any results that may be obtained by use of the code.*/

/*The following CSS will changed the language for “Order Information” in LegendOrder from English to Spanish*/

.LegendOrder{
    color: transparent;
}
.LegendOrder::after{
    content: "Información del Pedido";
    color: #A3A3A3;
}
```

Drop Down Localization CSS Customization

The following CSS snippet will help a developer to be able to hide non-applicable countries in the drop down list in the Hosted Payment Page.

```
/*ProPay provides the following code “AS IS.”
*ProPay makes no warranties and ProPay disclaims all warranties and conditions, express, implied or statutory, including without limitation the implied warranties of title, non-infringement, merchantability, and fitness for a particular purpose.
*ProPay does not warrant that the code will be uninterrupted or error free, nor does ProPay make any warranty as to the performance or any results that may be obtained by use of the code.*/

/*The following CSS will hide all the country options and only display the United States and Canada*/


.FormCountryField{}
.FormCountryField option{display:none}
.FormCountryField option:nth-child(1){display:inline}
.FormCountryField option:nth-child(2){display:inline}


/*The following CSS will hide the non-Continental States, excluding HI and AK providing a list comprised of only the 50 States of the United States of America*/

.FormStateDropDown{}
.FormStateDropDown option:nth-child(1){display:none}
.FormStateDropDown option:nth-child(2){display:none}
.FormStateDropDown option:nth-child(5){display:none}
.FormStateDropDown option:nth-child(7){display:none}
.FormStateDropDown option:nth-child(16){display:none}
.FormStateDropDown option:nth-child(45){display:none}
.FormStateDropDown option:nth-child(58){display:none}
```

3.3 Development Resources

The following tools can be provided upon request to techincalsupport@propay.com. They are designed to understand a developer more quickly how to redirect to Hosted Payment Page from the merchant's checkout page.

HPP Sample Dev Application

The HPP-Dev sample application is designed to be used by developers integrating the Hosted Payment Page into their software solutions.
To use this sample, put a Valid HostedTransactionIdentifier in the text input field, then press the start button.
In a production setting this value is best stored in the browser session as it is needed to get the results of the transaction once the Hosted Payment Page process is complete.

After successful creation of Hosted Transaction Identifier Merchant needs to redirect to https://protectpaytest.propay.com/hpp/v2/somehostedtransactionid from the merchant's checkout page to access the Hosted Payment Page.

HPP Sample CSS Application

The HPP-CSS sample application is designed to be used by developers integrating the Hosted Payment Page into their software solutions by providing the raw HTML for both Hosted Payment Page configurations as well as the default CSS file and a blank css file containing all the classes contained in the HTML. This will allow a developer to more quickly create a custom css, if desired, before creating a Hosted Transaction Identifier.