

# How to: Internal Javascript Library Usage For An HPP Template

- [Setting up library script](#)
- [Suggested scripting](#)
  - [External Library files](#)
    - [jQuery v3.6.1](#)
    - [jQuery v3.5.1](#)
    - [jQuery Validation Plugin - v1.16.0](#)
    - [jQuery Mask Plugin v1.14.16](#)
  - [Built in functions](#)
  - [Merchant Javascript Library functions](#)
    - [Display Formatted Amount](#)
    - [Set Element Property Value](#)
    - [Set Element Value](#)
    - [Create Card brands Dropdown](#)
    - [Set Element Html](#)
    - [Set CSS Property Value](#)
    - [Use Floating Labels On Fields](#)
    - [Show Tab By Id](#)
    - [Set Input Keypress Regex Filter](#)
    - [Apply Form Validations](#)
    - [Check Required Field Validation](#)
    - [Save Token Checkbox Logic](#)
    - [Remove Spaces From Card Number](#)
    - [Remove Non Numeric From Card Expiration](#)
    - [Mask Input Field With Pattern](#)
    - [Credit Card Validation And CVV Masking](#)
    - [Card Address Validation](#)
    - [Check Routing Number Validation](#)
    - [Hide Element On Event](#)
    - [Allow Numeric On Keypress](#)
    - [Call Event Prevent Default For Element](#)
    - [Bind Function For Element Event](#)
    - [Join Card Number Four Fields](#)
    - [Format Card Expiration MMY](#)
    - [Enable Tab & ShowHide On Click](#)
    - [Check Required Fields](#)

## Introduction

Historically we have allowed the usability of unique inline Javascript functions for HPP integrations. Due to our internal security analysis we have changed that process to avoid and combat the risk of any malicious scripts being used. As a solution we have developed a custom library containing behavioral functions that can be leveraged in the HPP. This process will replace the use of inline scripts. In many cases, the unique functionalities used inline before will be available as a merchant function within the merchant Javascript library.

## Referencing the merchant library

- Instead of using the script tag to reference your own custom Javascript file you will use the script tag to reference the merchant js library as shown below:

```
<script src="/hpp/scripts/MerchantScripts.js"
type="text/javascript"></script>
```

The scripts within the merchant library are broadly grouped under the **PageLoad and FormSubmit event handlers**. The desired function can be consumed by passing appropriate parameters [array of operations] to these event handlers.

```
//Suggested usage to invoke functionality after template loads
//PageLoad event handler example:
<body onload="onPageLoad([
    ['DisplayFormattedAmount', '#Amount', '#AmountSpan']
])">
```

```
//Suggested usage to invoke functionality after form submission button is
clicked
//FormSubmit event handler example:
<form name="hppform" action="" method="POST" autocomplete="off"
onsubmit="return submitButtonClicked([
    ['SetElementPropertyValue', '#submitButton', 'disabled', true],
    ['SetElementValue', '#submitButton', 'Processing. Please wait...']
]);">
```

In the example above, the first parameter in the array is the functionality name.

# Suggested scripting references

## External Library files

### jQuery v3.6.1

```
<script src="/hpp/libs/jquery/v3.6.1/jquery-3.6.1.min.js" type="text/javascript"></script>
```

### jQuery v3.5.1

```
<script src="/hpp/libs/jquery/v3.5.1/jquery-3.5.1.slim.min.js" type="text/javascript"></script>
```

### jQuery Validation Plugin - v1.16.0

```
<script src="/hpp/libs/jquery-validation/v1.16.0/jquery.validate.min.js" type="text/javascript"></script>
```

### jQuery Mask Plugin v1.14.16

```
<script src="/hpp/libs/jquery.mask/v1.14.16/jquery.mask.min.js" type="text/javascript"></script>
```

## Built in functions

It is suggested to analyze your HPP and confirm if built in Javascript functions can be used. The following is an example where RegEx functions can be used without creating new inline functions.

```
//Inline Example
function onlyNumberKey(evt) {
    // Only ASCII character in that range allowed
    var ASCIICode = (evt.which) ? evt.which : evt.keyCode
    if (ASCIICode > 31 && (ASCIICode < 48 || ASCIICode > 57)) return false;
    return true;
}
//Current Usage:
<input type="text" onkeypress="return onlyNumberKey(event)"></input>
//Suggested Usage:
<input type="text" onkeypress="return /[0-9]/i.test(event.key)"></input>
```

## Merchant Scripts Library functions

### Display Formatted Amount

```
//Inline example - Display Formatted Amount:
$(function () {
    $('#PaymentType').val('Credit');
    var amount = parseFloat(document.getElementById("Amount").value)
    var amountSpan = document.getElementById("AmountSpan")
    amountSpan.textContent = '$' + amount.toFixed(2);
});
//Suggested Usage for Display Formatted Amount:
<body onload="onPageLoad([
    ['DisplayFormattedAmount', '#Amount', '#AmountSpan']
])">
```

### Set Element Property/Value

```
//Inline example - Set Element Property/Property Value:
function validate(form)
{
    document.getElementById("submitButton").disabled = true;
    document.getElementById("submitButton").value = "Processing. Please
wait...";
    return true;
}
//Suggested Usage for Set Element Property/Property Value:
<form name="hppform" action="" method="POST" autocomplete="off"
    onsubmit="return submitButtonClicked([
        ['SetElementPropertyValue', '#submitButton', 'disabled', true],
        ['SetElementValue', '#submitButton', 'Processing. Please wait...']
    ]);">
```

### Create Card brands Dropdown

```
//Suggested Usage for Create Card Brands Dropdown:
<body onload="onPageLoad([
    ['CreateCardbrandsDropdown', 'input[name=CardBrandValues]',
'#cardBrand']
])">
```

### Set Element Html

```
//Inline example - Set Element Html:
$('#hppform').submit(function (event) {
    $('#submitButton').prop('value', 'Please Wait...');
});
//Suggested Usage for Set Element Html:
<form name="hppform" action="" method="POST" autocomplete="off"
    onsubmit="return submitButtonClicked([
        ['SetElementHtml', '#submitButton', 'Please wait...]]);">
```

## Set CSS Property Value

```
//Inline example - Set CSS property value:
$('#hppform').submit(function (event) {

    $('#submitButton').css('background-color', 'gray');

});
//Suggested Usage for Set CSS property value:
<form name="hppform" action="" method="POST" autocomplete="off"
    onsubmit="return submitButtonClicked([

        ['SetCssPropertyValue', '#submitButton', 'background-color',
'gray']]);">
```

## Use Floating Labels On Fields

## Show Tab By Id

## Set Input Keypress Regex Filter

## Apply Form Validations

*See Merchant Library Usage B*

## Check Required Field Validation

```
//Inline example - Check Required Field Validation:
function ClickSubmitButton(e) {
    e.preventDefault();
    e.stopPropagation();
    var requiredElements =
document.getElementById('hppform').querySelectorAll("[required]");
    for (var i = 0; i < requiredElements.length; i++) {
        if (!requiredElements[i].validity.valid) {
            return;
        }
    }
    document.getElementById("SubmitButton").disabled = true;
    document.getElementById('hppform').submit();
}
//Suggested Usage for Check Required Field Validation:
<form name="hppform" id="hppform" action="" method="POST" autocomplete="off"
class="form-horizontal"
    onsubmit="return submitButtonClicked([
        ['CheckRequiredFieldValidation', 'hppform'],
        ['SetElementPropertyValue', '#SubmitButton', 'disabled',
true]]);">
```

## Save Token Checkbox Logic

```
//Inline example - Save Token Checkbox Logic:
function Setup() {
    var hppForm = document.getElementById("hppform");
    hppForm.addEventListener("submit", function (e) {
        ClickSubmitButton(e)
    });
    if ($("#name=cf_show_save_token_checkbox_Name").val() !== undefined &&
    $("#name=cf_show_save_token_checkbox_Name").val().toUpperCase() === "TRUE")
    {
        $(".save_token_checkbox_div").show();
    } else {
        $(".save_token_checkbox_div").hide();
    } // set up change handler for the checkbox
    $("#SaveTokenCheckbox").on("change", function () {
        if ($(this).is(":checked")) {
            $("#name=cf_save_token_Name").val("True");
        } else {
            $("#name=cf_save_token_Name").val("False");
        }
    });
    $("#name=cf_save_token_Name").val("False");
    $("#PaymentType").val("ACH");
}
$(document).ready(Setup());
```

### **//Suggested Usage for Save Token Checkbox Logic:**

```
<body onload="onPageLoad([
    ['SetElementValue', '[name=cf_save_token_Name]', 'False'],
    ['SetElementValue', '#PaymentType', 'ACH'],
    ['SaveTokenCheckboxLogic', '[name=cf_show_save_token_checkbox_Name]',
'.save_token_checkbox_div', '#SaveTokenCheckbox',
'[name=cf_save_token_Name]'],
])">
```

## Remove Spaces From Card Number

### **//Inline example - Remove Spaces From Card Number:**

```
function ClickSubmitButton(e) {
    var cardNumber =
document.getElementById("hppform").querySelector("input[name=\"CardNumber\"]"
);
    cardNumber.value = cardNumber.value.replace(" ", "");
}
```

### **//Suggested Usage for Remove Spaces From Card Number:**

```
<form name="hppform" id="hppform" action="" method="POST" autocomplete="off"
class="form-horizontal"
    onsubmit="return submitButtonClicked([
        ['RemoveSpacesFromCardNumber', 'input[name=' +
'CardNumber'+']']]);">
```

## Remove Non-Numeric From Card Expiration

```
//Inline example - Remove Non-numeric from card expiry:
    var cardExpiration =
document.getElementById("hppform").querySelector("input[name=\"CardExpiration
\"]");
    let expVal = cardExpiration.value;
    if (expVal.includes("/")) {
        expVal = expVal.replace("/", "");
    }
    if (expVal.length === 6) {
        expVal = expVal.substring(0, 2) + cardExpiration.value.substring(5);
    }
    cardExpiration.value = expVal;
    var requiredElements =
document.getElementById('hppform').querySelectorAll("[required]");
    for (var i = 0; i < requiredElements.length; i++) {
        if (!requiredElements[i].validity.valid) {
            return;
        }
    }
}
//Suggested Usage for Remove Non-numeric from card expiry:
<form name="hppform" id="hppform" action="" method="POST" autocomplete="off"
class="form-horizontal"
        onsubmit="return submitButtonClicked([
            ['RemoveNonNumericFromCardExpiration', 'input[name='
+ 'CardExpiration'+']' ]
        ]);">
```

## Mask Input Field With Pattern

### Credit Card Validation And CVV Masking

### Card Address Validation

*See Merchant Library Usage D*

## Check Routing Number Validation

*See Merchant Library Usage E*

## Hide Element On Event

## Allow Numeric On Keypress

## Call Event Prevent Default For Element

## Bind Function For Element Event

*See Merchant Library Usage F*

## Join Card Number Four Fields

**//Inline example - Join Card Number Four Fields:**

```
document.getElementById("cardNumber").value =
document.getElementById("cardNumberA").value +
document.getElementById("cardNumberB").value +
document.getElementById("cardNumberC").value +
document.getElementById("cardNumberD").value;
```

**//Suggested usage for Join Card Number Four Fields:**

```
<form name="hppform" action="" method="POST"
  onsubmit="return submitButtonClicked([
    ['JoinCardNumberFourFields', '#cardNumber', '#cardNumberA',
'#cardNumberA', '#cardNumberC', '#cardNumberD']
  ]);">
```

## Format Card Expiration MMY:Y

**//Inline example - Format Card Expiration MMY:Y:**

```
var mmVal = '';
if (document.getElementById("cardExpirationMM").value.length == 1) {
  mmVal = '0' + document.getElementById("cardExpirationMM").value;
} else {
  mmVal = document.getElementById("cardExpirationMM").value;
}
document.getElementById("cardExpiration").value = mmVal +
document.getElementById("cardExpirationYY").value.substr(-2);
```

**//Suggested usage for Format Card Expiration MMY:Y:**

```
<form name="hppform" action="" method="POST"
  onsubmit="return submitButtonClicked([
    ['FormatCardExpirationMMYY', '#cardExpiration', '#cardExpirationMM',
'#cardExpirationYY']
  ]);">
```

## Enable Tab & ShowHide On Click

**//Inline example - Enable Tab & Show/Hide on Click:**

```
<script>
    $(function ()
    {
        $('.nav-tabs a').on('click', function (e)
        {
            e.preventDefault();
            $(this).tab('show');
            $('#PaymentType').val($(this).text());

            $('.nav-tabs a').hide();
            $(this).show();
        });
    });
```

**//Suggested usage for Enable Tab & Show/Hide on Click:**

```
<body onload="onPageLoad([
    ['EnableTabShowHideOnClick']

])">
```

## Check Required Fields

**//Inline example - Check Required Fields:**

```
function cmdNullCheck() {
    if (document.getElementById("nameOnCard").value.length == '' ||
        document.getElementById("cardNumber").value.length == '' ||
        document.getElementById("cardExpirationMM").value.length == '' ||
        document.getElementById("cardExpirationYY").value.length == '' ||
        document.getElementById("cardCvv").value.length == '' ||
        document.getElementById("cardNumber").value.length < 16) {
        alert('Please verify that all fields have been completed.');
```

```
    } else {
        document.getElementById("submitbutton").click();
    }
}

<input type="button" onclick="cmdNullCheck()" id="submitbuttonPreCheck"
value="Submit">
<button type="submit" name="submitbutton" id="submitbutton" >
```

**//Suggested usage for Check Required Fields:**

```
<input type="button" id="submitbuttonPreCheck" value="Submit"
onclick="submitButtonClicked([
    ['CheckRequiredFields', '#submitbutton', ['#nameOnCard',
'#cardNumber', '#cardExpirationMM', '#cardExpirationYY', '#cardCvv' ],
[['#cardNumber', 16]]
]);">
```

## **Merchant Library Usage Examples**

**Usage A** – Create Cardbrand dropdown, Set Element Property Value, Set Element HTML, Set CSS Property Value

**Usage B** – Floating Labels, Show Tab By ID, Keypress Regex Filter, Form Validations

**Usage C** – Remove Spaces From CC, Remove Non-Numeric From Card Expiration, Check Required Fields, Set Element Property Value, Create Cardbrand Dropdown, Save Token checkbox Logic, Set Element Value

**Usage D** – Mask Input Field With Pattern, CC Validation And CVV Masking, Card Address Validation

**Usage E** – Check Routing Number Validation

**Usage F** – Hide Element On Event, Allow Numeric On Keypress, Prevent Default For Element, Bind Function For Element Event