

Payliance

Card Iframe SDK

v1

Contents

Overview	2
Endpoints	2
Description	2
Authentication Key	3
Sample Code (C#)	3
Configuration Object	4
Sample Code (Javascript)	4
FormInputs reference	6
Sample Code (Javascript)	6
Loading the iframe	7
Sample Code (HTML / Javascript)	7
Using the iframe	8
Sample Code (Javascript)	8
Putting it all together	9
Sample Code (C# / HTML / Javascript)	9

Overview

Endpoints

- Staging = <https://staging.cardproxy.payliance.com/>
- Production = <https://cardproxy.payliance.com/>

Description

The Payliance iframe provides merchants the ability to tokenize card data without needing to accept the card data, helping them reduce their scope for PCI compliance. While the form hosting the iframe will reside on your server, the iframe itself resides within Payliance's secure environment.

Authentication Key

The first step is to generate a temporary use Authentication Key using your API Key and some additional pieces of information.

* Note this needs to be done server side to avoid exposing your API Key publicly.

Input	Description	Example
userName	The Payliance portal user who generated the API Key	demo
accessKey	The unique identifier of the generated API Key	00000000-0000-0000-0000-000000000000
timeStamp	The current UTC timestamp in yyyyMMddHHmmss format	20180102030405
originUrl	The protocol, port (if specified), and host of your application	http://yourdomain.com
secretKey	The generated API Key	{Base64}

Sample Code (C#)

```
var userName = "demo";

var accessKey = "00000000-0000-0000-0000-000000000000";

var secretKey = "{Base64}";

var timeStamp = DateTime.UtcNow.Year.ToString() +
    DateTime.UtcNow.Month.ToString().PadLeft(2, '0') +
    DateTime.UtcNow.Day.ToString().PadLeft(2, '0') +
    DateTime.UtcNow.Hour.ToString().PadLeft(2, '0') +
    DateTime.UtcNow.Minute.ToString().PadLeft(2, '0') +
    DateTime.UtcNow.Second.ToString().PadLeft(2, '0');

var originUrl = "http://yourdomain.com";

var hmac = new System.Security.Cryptography.HMACSHA256();
hmac.Key = System.Text.Encoding.UTF8.GetBytes(secretKey);
var buffer = userName + accessKey + timeStamp + originUrl;
var hash = hmac.ComputeHash(System.Text.Encoding.UTF8.GetBytes(buffer));
var authenticationKey = Convert.ToBase64String(hash);
```

Configuration Object

The next step is to build a Configuration Object using the generated Authentication Key and some additional pieces of information.

Input	Description	Example
UserName	The Payliance portal user who generated the API Key	demo
AccessKey	The unique identifier of the generated API Key	00000000-0000-0000-0000-000000000000
TimeStamp	The current UTC timestamp in yyyyMMddHHmmss format	20180102030405
OriginUrl	The protocol, port (if specified), and host of your application	http://yourdomain.com
AuthenticationKey	The calculated hash value	{Base64}
SourceUrl	The Payliance endpoint, either staging or production	https://cardproxy.payliance.com/
ContainerId	The "id" of the <div> that will host the iframe	myiframe
CssUrl	An optional reference to a style sheet used to customize the appearance of the iframe	http://yourdomain.com/custom.css
FormInputs	An optional object used to customize the behavior of the iframe	FormInputs: { Address2: { IsHidden: true } }
CardId	An optional reference to card id that when set, can be used to make tokenized payment requests. This will give you a different view with just the CVV field that will use this value for Tokenized payment requests. The postform will have the capability to skip the doSubmit/tokenize and move immediately to the payment when this value is present.	00000000-0000-0000-0000-000000000000

Sample Code (Javascript)

```
var config = {  
  UserName: "demo",  
  AccessKey: "00000000-0000-0000-0000-000000000000",  
  TimeStamp: "20180102030405",  
  OriginUrl: "http://yourdomain.com",
```

```
AuthenticationKey: "{Base64}",
SourceUrl: "https://cardproxy.payliance.com/",
ContainerId: "myIframe",
CssUrl: "http://yourdomain.com/custom.css",
FormInputs: { Address2: { IsHidden: true } },
CardId: "00000000-0000-0000-0000-000000000000",
};
```

FormInputs reference

To alter the standard behavior of the iframe use the FormInputs object to specify the DefaultValue, IsRequired and IsHidden setting for each field.

* Note all settings are not available for all fields, please refer to the grid below.

Input	DefaultValue	IsRequired	IsHidden
CardNumber	n/a	always	never
ExpirationDate	n/a	always	never
CVV	n/a	true/false	true/false
FirstName	string	always	never
LastName	string	always	never
Address1	string	true/false	true/false
Address2	string	true/false	true/false
City	string	true/false	true/false
State	string	true/false	true/false
ZipCode	string	true/false	true/false
Phone	string	true/false	true/false
Email	string	true/false	true/false

Sample Code (Javascript)

```
FormInputs: {
  FirstName: {
    DefaultValue: "John"
  },
  LastName: {
    DefaultValue: "Smith"
  },
  Address2: {
    IsHidden: true
  },
  City: {
    IsHidden: true
  },
  State: {
    IsHidden: true
  },
  ZipCode: {
    IsRequired: true
  },
  Phone: {
    IsHidden: true
  },
  Email: {
    IsHidden: true
  }
}
```

Loading the iframe

The final step is to create and load the iframe using our Javascript library and passing the Configuration Object.

Sample Code (HTML / Javascript)

```
<script src="https://cardproxy.payliance.com/iframe/payliance-1.0.4.min.js"></script>
<div id="myIframe"></div>
<script>
    var iframe = new Payliance.PaymentCard.TokenizeIframe(config);
    iframe.load();
</script>
```

Using the iframe

To begin using the iframe add Event Listeners to handle messages passed from the iframe back to host form and implement the action methods that support your workflow.

Sample Code (Javascript)

```
// event listeners...
iframe.on("validate", function (data) { console.log(JSON.stringify(data)); });
iframe.on("tokenize", function (data) { console.log(JSON.stringify(data)); });
iframe.on("failure", function (data) { console.log(JSON.stringify(data)); });
iframe.on("verify", function (data) { console.log(JSON.stringify(data)); });
iframe.on("payment", function (data) { console.log(JSON.stringify(data)); });
iframe.on("updateCard", function (data) { console.log(JSON.stringify(data)); });
iframe.on("updateState", function (data) { console.log(JSON.stringify(data)); });

// methods...
iframe.load();
iframe.validate();
iframe.submit();
iframe.verify(cardId, cvv, memo);
iframe.payment(uniqueRef, cardId, cvv, amount, memo, convenienceFee);
iframe.updateCard();
iframe.updateState();
iframe.reset();
```

Putting it all together

The following example is written using the ASP.NET Razor (.cshtml) syntax, to learn more go to https://www.w3schools.com/asp/razor_intro.asp.

Sample Code (C# / HTML / Javascript)

```
@{
    var userName = "demo";
    var accessKey = "00000000-0000-0000-0000-000000000000";
    var secretKey = "{Base64}";
    var timeStamp = DateTime.UtcNow.Year.ToString() +
        DateTime.UtcNow.Month.ToString().PadLeft(2, '0') +
        DateTime.UtcNow.Day.ToString().PadLeft(2, '0') +
        DateTime.UtcNow.Hour.ToString().PadLeft(2, '0') +
        DateTime.UtcNow.Minute.ToString().PadLeft(2, '0') +
        DateTime.UtcNow.Second.ToString().PadLeft(2, '0');

    var originUrl = "http://yourdomain.com";

    var hmac = new System.Security.Cryptography.HMACSHA256();
    hmac.Key = System.Text.Encoding.UTF8.GetBytes(secretKey);
    var buffer = userName + accessKey + timeStamp + originUrl;
    var hash = hmac.ComputeHash(System.Text.Encoding.UTF8.GetBytes(buffer));
    var authenticationKey = Convert.ToBase64String(hash);
}
<script src="https://cardproxy.payliance.com/iframe/payliance-1.0.3.min.js"></script>

<div id="myIframe"></div>

<script>
    var config = {
        UserName: "@userName",
        AccessKey: "@accessKey",
        TimeStamp: "@timeStamp",
        OriginUrl: "@originUrl",
        AuthenticationKey: "@Html.Raw(authenticationKey)",
        SourceUrl: "https://cardproxy.payliance.com/",
        ContainerId: "myIframe",
        CssUrl: "http://yourdomain.com/custom.css",
        FormInputs: { Address2: { IsHidden: true } }
    };

    var iframe = new Payliance.PaymentCard.TokenizeIframe(config);

    // event listeners...
    iframe.on("validate", function (data) { console.log(JSON.stringify(data)); });
    iframe.on("tokenize", function (data) { console.log(JSON.stringify(data)); });
    iframe.on("failure", function (data) { console.log(JSON.stringify(data)); });
    iframe.on("verify", function (data) { console.log(JSON.stringify(data)); });
    iframe.on("payment", function (data) { console.log(JSON.stringify(data)); });
    iframe.on("updateCard", function (data) { console.log(JSON.stringify(data)); });
    iframe.on("updateState", function (data) { console.log(JSON.stringify(data)); });
};
```

```
// methods...
iframe.load();
//iframe.validate();
//iframe.submit();
//iframe.verify(cardId, cvv, memo);
//iframe.payment(uniqueRef, cardId, cvv, amount, memo, convenienceFee);
//iframe.updateCard();
//iframe.updateState();
//iframe.reset();
</script>
```