



# Payeezy.JS Integration

---

Release Version 1.1

## Introduction

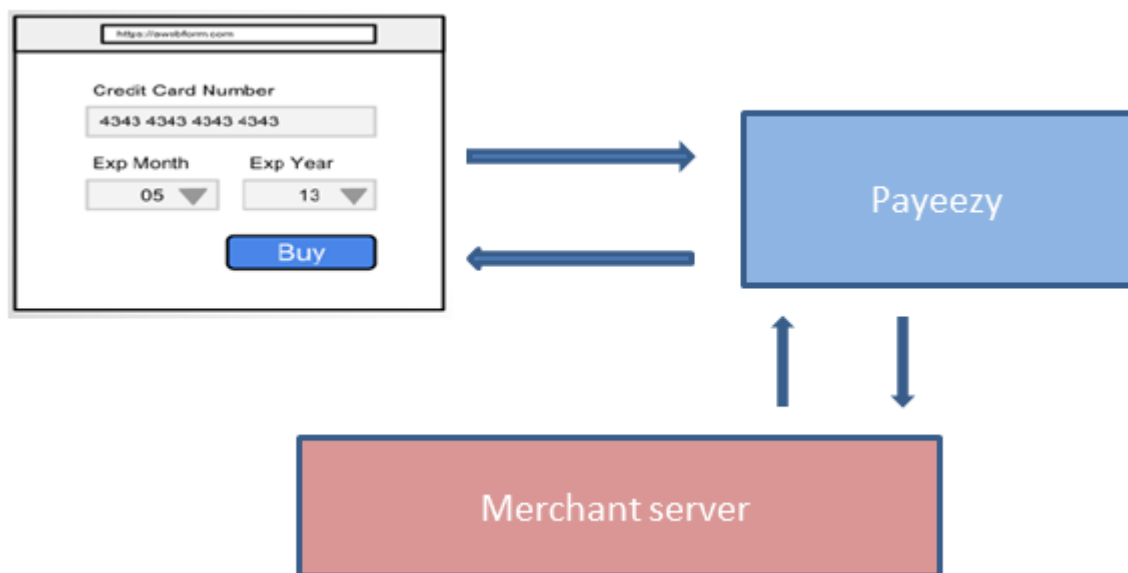
We developed Payeezy.com to provide a simple, powerful ecommerce payment option for developers that delivers easy, scalable and secure online and mobile payments. Essentially, we are providing access to our unparalleled ecommerce product offerings by helping you integrate your solutions through the site's open API environment ... for Apple Pay In-App, for shopping carts, for ecommerce platforms, and more. Code once and you're set to go.

Payeezy.js is the Payeezy SDK built to allow users to make API calls to Payeezy SDK via java script. Payeezy.js implements method to create "Time-Bound Single-Use" or "Multi-Use" tokens that tokenize credit cards. You will need TransArmor® enabled on your merchant account to do token based transactions.

## Conceptual Overview

Payeezy is for anyone who wants to accept payments online or via a mobile device. Whether looking for a more PCI compliant solution or wanting to seamlessly integrate payment processing to your website, the Payeezy solution makes it easy to accept a wide range of payments anytime, anywhere. A highly intuitive web-based interface and simple APIs allow you to remove the complexity of accepting card-not-present payments quickly and easily, keeping your focus on operating and growing your business.

## Flow Diagram

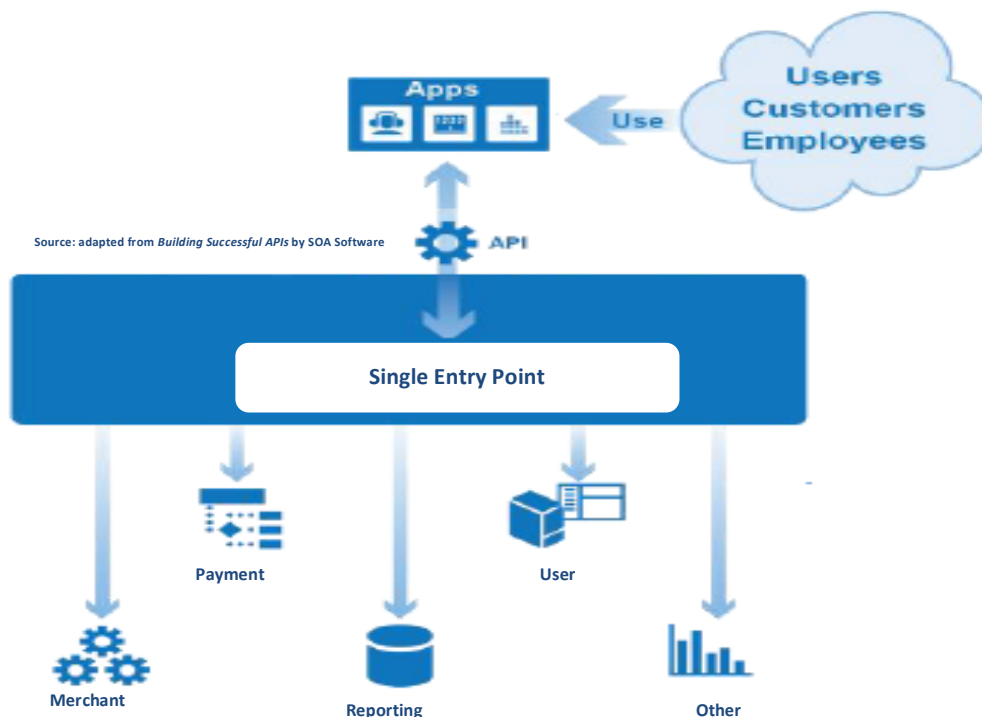


**User action:** Consumer does a checkout on the merchant site by clicking on buy/submit

1. Payeezy.js (which is included on the page using a <script> tag) intercepts the form submit, and then asynchronously posts the credit card details to Payeezy. This call uses JSONP over https, passing in API key and an identifier– trtoken. Both of these values are provisioned for the merchant’s developer via the developer portal.
2. On success Payeezy returns a one-time use token (type=payeezy) to payeezy.js. Payeezy.js saves the pertinent token information in a hidden form field on the checkout form. Failure messages are returned in the response, which can be handled as appropriate.
3. Payeezy.js then submits the form with the hidden token along with other form info to the merchant server. The credit card related fields that were used for tokenization are stripped out of the form i.e. the merchant server never sees them.
4. Merchants server submits the token to Payeezy to complete the transaction using the appropriate server side library (optional to use but recommended). This API call uses HMAC authentication, over https. Refer to the Payeezy developer portal – API Docs and Sandbox section to know more.
5. Payeezy completes the transaction with the provided token (authorization, purchase). The response contains a new token to use for the related secondary transaction (capture, void, refund). Success/failure message is handled accordingly by the merchant’s server

## User experience flow

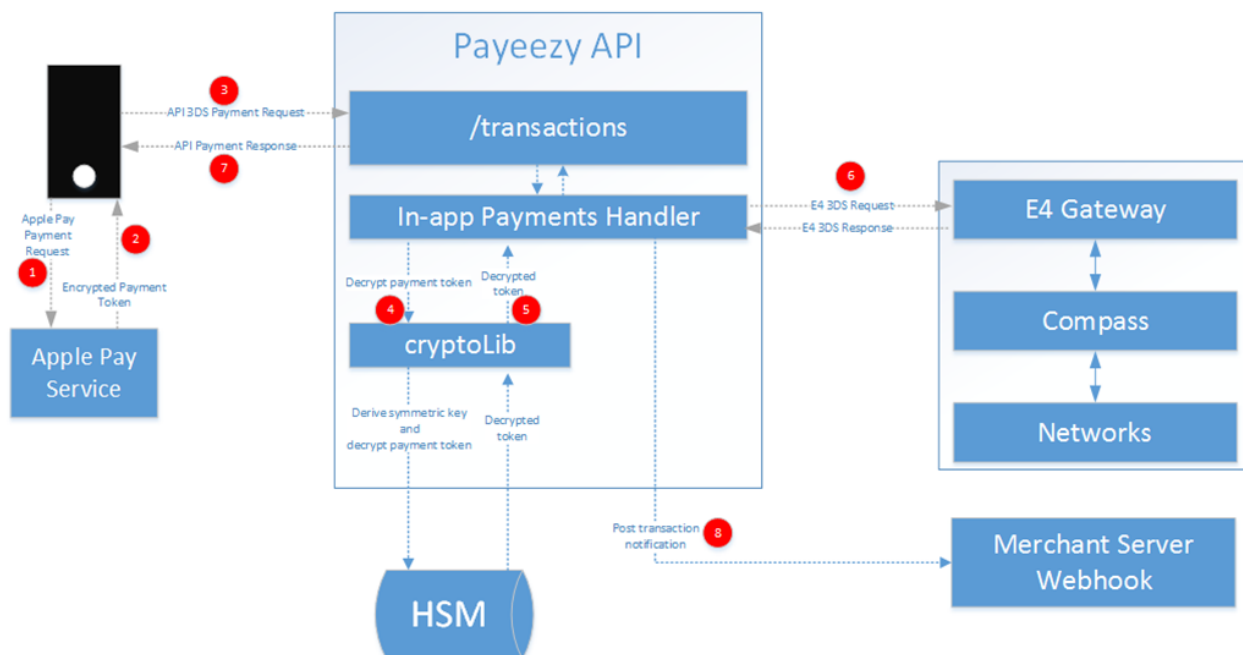
Following is a general user experience diagram. The app users interacts with the app which in turn makes the calls to the Payeezy SDK API which takes care of all the relevant functionality like merchant authorization, payments, transactions, reporting and any other required functions. This helps to build seamless user experience.



- **User action:** Buyer does a checkout on the merchant hosted site by clicking on buy/submit button. This needs to be a secure page.
- Upon Submit the form gets posted.
- The Payeezy API is invoked over a SSL connection and submits the order information securely. The call requires providing an authorization header with a HMAC hash. Refer to API docs section for more information on this.
- Payeezy process the transactions(such as authorize, purchase) with the provided transaction details and sends back the response
- Success/failure message needs to be handled accordingly by the client.

NOTE: The merchant/third party is responsible for PCI compliance and may be required to provide First Data with additional documentation and/or their attestation of compliance.

Following is the general process flow diagram for the system.



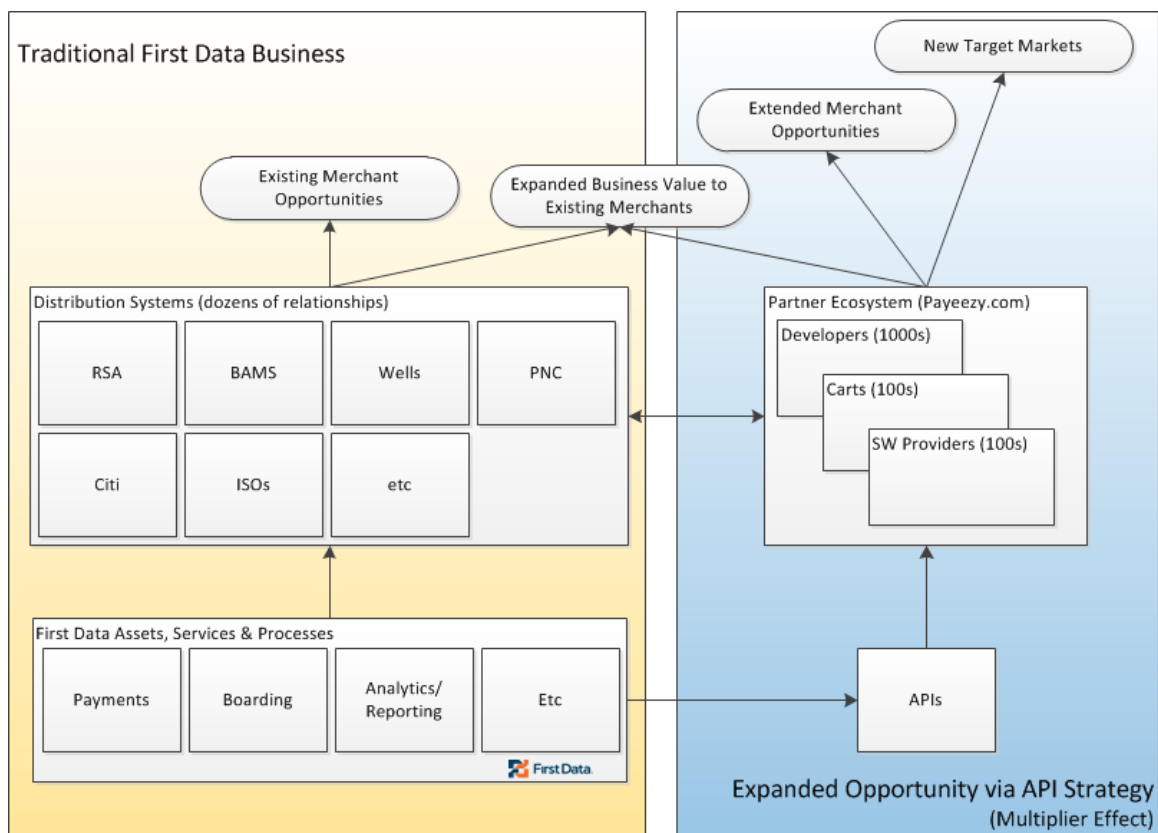
When the user initiates a transaction from a mobile device the payment details are passed to the Apple Pay service to generate the payment authorization token. This token with other payment details is then passed to the Payeezy API to proceed with the transactions. Payeezy API will generate another token mapping to this token and pass it to E4 Gateway via a First API call. The E4 Gateway will further continue with the transaction by passing it to payment networks via compass. The transaction success/failure will be notified to you via merchant server web hook notification to the Payeezy API payments handler.



Payeezy completes the transaction (such as authorize, purchase) with the provided token and sends back a new token to be used for any related secondary transaction (such as capture, void, refund). Success/failure message is handled accordingly by the merchant's server.

The user experience of the application revolves around the seamless payment experience for the customer on the request to process a transaction.

Following is the ecosystem architecture diagram for Payeezy.





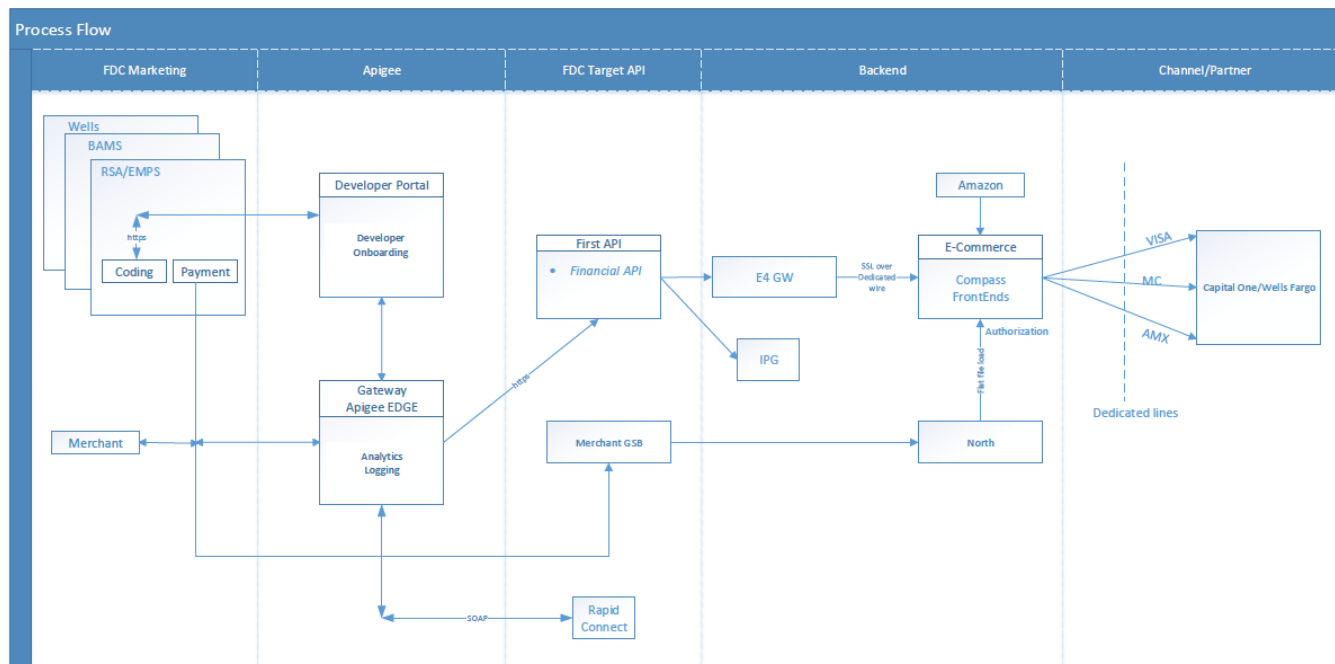
## Minimum Requirements

Following are the minimum requirements for Payeezy.js integration

1. The merchant account should be TransArmor enabled.
2. The Java SDK version should be 1.6 or higher.
3. Compatible browsers (any one)
  - a. The latest version of Internet explorer 6+
  - b. Google chrome 39.x
  - c. Mozilla Firefox 33.0
  - d. Safari : v5.1.7
4. Get started: Explore our Payeezy site anonymously! If you like what you see, visit our registration pages to [sign up for a test account](#) and begin the integration process!

Explore our site <http://developer.payeezy.com> anonymously if you like; you'll even be able to test some of the code samples we provide on the API Docs & Sandbox tab. Once you [create an account](#) with us, you'll receive a unique API Key that allows you to generate transactions against a test account on your My Merchants tab. Fully register and certify your solution, and you'll be able to begin bringing merchants on board and generating money!

As a predecessor to launching Payeezy, we established a strong community of shopping carts and ecommerce software platforms. Learning more about these partners and how they are leveraging our



APIs is easy! As a start, please check out our [partner directory](#).



This diagram elaborates our partnership program which includes merchant registration, onboarding and other steps in the process.

Becoming a Payeezy Partner is easy! Whether you are a developer and decide to integrate and certify your solution, or you are an ecommerce software provider and/or a shopping cart, you can decide which level of partnership is right for you. Please see our Payeezy [Partnership Program](#) to learn more about the benefits of the program.

Explore our Payeezy site anonymously! If you like what you see, visit our registration pages to [sign up for a test account](#) and begin the integration process!

### Brief integration sequence and steps

Following are the brief integration steps to follow in any integration.

#### Developer portal setup:

Your developer account on [developer.payeezy.com](#) needs to be provisioned for Payeezy JS. On the developer portal, select the API you want to entitle, and add Entitlements using the Entitlements Tab.

Add more, edit or delete them as you like.

+ ADD A NEW API

new My App

SANDBOX

KEYS PRODUCTS ENTITLEMENTS

Add more, edit or delete them as you like. + Add Entitlements

Once you've picked the merchant to enable, we will create a unique merchant identifier for you. You will need the API Key and the merchant identifier (highlighted below) to initialize `payeezy.js`.

new My App

SANDBOX

KEYS PRODUCTS ENTITLEMENTS

Merchant: CFO test Edit Entitlement

Merchant Identifier 5edf46316f6b

Referer

Supported Token Types payeezy

#### Initializing the Payeezy JS:

## Region specific information if (applicable)

If you are looking for any region specific information on the site which you are not able to find please check our API Docs & Sandbox area or collaborate with others in the Forums and if you still have questions, please contact us via support@payeezy.com or give us a call at 1-855-799-0790.

## Extra support information

If you have already tried our [API Docs & Sandbox](#) area or collaborated with others in the [Forums](#) and still have questions, please contact us via support@payeezy.com or give us a call at 1-855-799-0790.

## Download information (if applicable)

Following are the download links for the SDK and sample code.

<https://developer.payeezy.com/sdk-downloads>

<https://github.com/payeezy/payeezy>

## Transaction data requirements explanation -

Following are the various transactions that can be done using Payeezy API. There are various data requirements for each of these type of transactions.

- **Authorization**
  - **Following are the data properties required for the authorization transaction.**  
**Following are the parameters**
    - **Transaction Type – Type of transaction like authorize/purchase/capture etc.**
    - **the amount – Amount specified to be charged**
    - **Card Holder's Name : Name of the cardholder to**
    - **Card Number**
    - **Card Type**
    - **CVV number**
    - **Card Expiration**



- **Payment**
  - Following are the data properties required for the payment transaction. One of them is
    - Transaction Id
    - Transaction Tag Number
    - Transaction Type
    - the amount
    - Card Holder's Name
    - Card Number
    - Card Type
    - CVV number
    - Card Expiration
  -
- **Capture**
  - Following are the data properties required for the capture transaction. One of them is
    - Transaction Id
    - Transaction Tag Number
    - Transaction Type
    - the amount
    - Card Holder's Name
    - Card Number
    - Card Type
    - CVV number
    - Card Expiration
- **Void**
  - Following are the data properties required for the void transaction. One of them is
    - Transaction Id
    - Transaction Tag Number
    - Transaction Type
    - the amount
    - Card Holder's Name
    - Card Number
    - Card Type
    - CVV number
    - Card Expiration
- **Refund**
  - Following are the data properties required for the refund transaction. One of them is
    - Transaction Id
    - Transaction Tag Number

- Transaction Type
  - the amount
  - Card Holder's Name
  - Card Number
  - Card Type
  - CVV number
  - Card Expiration
- Split Shipment
    - Following are the data properties required for the split shipment transaction. One of them is
      - Transaction Id
      - Transaction Tag Number
      - Split shipment number
      - Transaction Type
      - the amount
      - Card Holder's Name
      - Card Number
      - Card Type
      - CVV number
      - Card Expiration

## Submitting/generating the transaction example:

Please refer to our website for details of our application API calls. The API calls are explained in detail on our website. Following is the link.

<https://developer.payeezy.com/payeezy-api-reference/apis>

---

Environment	URL
Sandbox	<code>https://api-cert.payeezy.com/v1/transactions</code>
Live	<code>https://api.payeezy.com/v1/transactions</code>

---