



The next way to get paid.

**PAYMENT GATEWAY
TECHNICAL SPECIFICATION
AND
APPLICATION PROTOCOL INTERFACE**

Version 1.17
Release Date October 04, 2019

Global Electronic Technology
970 W 190th St #650
Torrance CA 90502
www.gettrx.com

2018 Global Electronic Technology, "GET" and EdgePay are federally registered service marks of Global Electronic Technology, Inc, in the United States. The information contained herein is the property of GET and contain confidential information that is produced solely for the benefit of GET and its approved partners. All parties should keep all information contained herein confidential and should not disclose or disseminate to other parties without the express written consent of GET.

DISCLAIMER

Global Electronic Technology EdgePay Application Protocol Interface Specification documents are licensed on an 'As Is' basis. No representations or warranties are expressed or implied, including, but not limited to, warranties of quality, suitability, or fitness for a use or purpose. Global Electronic Technology or its affiliates shall have no liability or responsibility to you nor any other person or entity with respect to liability, loss, damages, including profits whether foreseeable or not, or any other Global Electronic Technology specifications. Use of this Specification document signifies agreement with this Disclaimer.

LICENSE

Ownership of all Global Electronic Technology Specifications documents, related to all intellectual property rights therein, shall remain always with Global Electronic Technology. All rights not expressly granted to you herein are reserved for Global Electronic Technology. Use of this and all other Global Electronic Technology Specifications will require a signed services agreement by the end user of the service. Global Electronic Technology grants you the right to use the EdgePay Application Protocol Interface Specification for the sole purpose of transmitting transactions directly to Global Electronic Technology from any originating device.

RESTRICTED USE

Copying of this Specification, or other files related to this service for use other than for transmitting transactions using this Specification to systems other than Global Electronic Technology is strictly prohibited. You may not remove any trademark, copyright, or other proprietary or confidential notices from the documentation. Under no circumstances can you reverse engineer, translate, re-direct, emulate, disseminate to other entities the information contained herein, nor shall you attempt to create source code/object code to emulate the EdgePay Application Protocol Interface Specification with the purpose of emulating the host services.

The EdgePay Application Protocol Interface Specification are provided solely to enable you to develop services to transmit transactions to Global Electronic Technology Use of any services, using this Specification to transmit transactions to other systems that do not in turn send transactions to Global Electronic Technology is strictly prohibited and a violation of this agreement and subject to all remedies available by law.

Use of the EdgePay Application Protocol Interface Specification documents do not grant you any ownership rights of this Specification, technology, or services. Except as provided herein, any other use of the EdgePay Application Protocol Interface Specification is strictly prohibited.

Global Electronic Technology
970 W 190th St #650
Torrance CA 90502
888.775.1500
www.gettrx.com

TABLE OF CONTENTS

| | | |
|----------|--|-------------|
| 0 | PREFACE | 1-6 |
| 0.1 | PURPOSE OF THIS DOCUMENT | 1-6 |
| 0.2 | CHANGES AND MODIFICAITONS..... | 1-6 |
| 1 | SYSTEM OVERVIEW | 1-8 |
| 1.1 | GENERAL FEATUES AND OPTIONS | 1-8 |
| 1.2 | AUTHENTICATION:..... | 1-9 |
| 1.3 | HOST BASED, TIME INITIATED PROCESSING..... | 1-9 |
| 1.4 | DUPLICATE CHECKING | 1-9 |
| 1.5 | TOKENIZATION | 1-10 |
| 1.6 | DEVELOPMENT AND PRODUCTION URL'S | 1-10 |
| 1.7 | INTEGRATION SUPPORT..... | 1-10 |
| 2 | SUMMARY OF INTERFACES AVAILABLE AND THEIR USES | 2-12 |
| 2.1 | EDGEPAY INTERFACES | 2-12 |
| | <i>EdgePay exposes a list of interfaces, each of which may be used as a 'stand-alone' function or in combination to achieve a certain set of business objectives. Below is a list of each interface and the team's original thoughts on its use. You may find other uses or benefits that can be applied. Please feel free to contact us to discuss other options and uses as we have constructed the system with great flexibility and are always open to discussing your specific business needs.</i> | <i>2-12</i> |
| 2.1.1 | <i>EdgePay API – A set of RESTful interfaces to allow for sending in Requests via HTTP 1.1 and receiving responses. The API supports all Payments, including ACH, Refunds of previous transactions, Recurring Payments, and Tokenization of Card or Check data.....</i> | <i>2-12</i> |
| 2.1.2 | <i>EdgePay Pivot – A Library and supporting utility that allows for the exchange of Sensitive Credit Card data at the Consumer's browser for API processing using a token in lieu of the Credit Card information.....</i> | <i>2-12</i> |
| 2.1.3 | <i>EdgePay Hosted Pay Page – A Payment function that routes the payment from your "Pay" button to our secure Hosted Payment page for inputting payment information, card or echeck. We handle all the processing and deliver a response to a page you design with a Redirect Post back for your server.</i> | <i>2-12</i> |
| 2.1.4 | <i>EdgePay Batch Pay – A function within the Processing Center that allows you to upload a batch of payments to process. Using a simple csv file, you can deliver payments, (Ex. Capture transactions for all those items you shipped today) and we are processing in bulk, delivering responses to you back within the Processing Center reporting within minutes.</i> | <i>2-12</i> |
| 2.1.5 | <i>EdgePay Recurring Payment Manager – Through the Processing Center you create Billing Plans that represent products or services or offers. Then you attached Contracts and Customers to those Billing Plans and we will bill the customers for you!.....</i> | <i>2-12</i> |
| 2.1.6 | <i>EdgePay Token Service – Either through the Processing Center of via our RESTful API we provide you the ability to input either Credit Card, Debit Card, or Checking Account information and we return you a 16-digit token you can safely store for making future payment requests.....</i> | <i>2-12</i> |
| 2.1.7 | <i>EdgePay eCommerce Plug In's – Use one of our Plug-in's with your eCommerce site to support payment processing. The plug-in works in conjunction with our Hosted Payment page and automates the features to simplify the start-up process.....</i> | <i>2-12</i> |
| 2.1.8 | <i>EdgePay Fraud Management Service – We provide you a tool set within our Processing Center to allow you to establish Static and Dynamic filters to analyse payment transactions while they are happening. Transactions caught by the filter will be displayed for your review. Additionally, we</i> | |

have an Enhanced version that can identify a computer to allow you to filter based on this dimension for decisioning. 2–13

| | | |
|----------|--|-------------|
| 3 | EDGEPAY API, (APPLICATION PROTOCOL INTERFACE)..... | 3–14 |
| | WORKING TOGETHER! | 3–14 |
| | 3.1.1 <i>Introduction</i> | 3–14 |
| | 3.1.2 <i>EdgePay Payment API</i> | 3–15 |
| | 3.1.3 <i>Before You Begin Q&A</i> | 3–15 |
| | 3.2 PAYMENT API..... | 3–18 |
| | 3.3 FULL PAYMENT API TABLE..... | 3–19 |
| | 3.3.1 <i>Refund</i> | 3–26 |
| | 3.3.2 <i>Token Request</i> | 3–27 |
| | 3.3.3 <i>Recurring Payments</i> | 3–28 |
| | 3.3.4 <i>ACH</i> | 3–30 |
| | 3.4 TRANSACTION RESPONSES..... | 3–31 |
| | 3.4.2 <i>Payment Response</i> | 3–33 |
| | 3.4.3 <i>Refund Response</i> | 3–34 |
| | 3.4.4 <i>Token Response</i> | 3–34 |
| | 3.4.5 <i>Recurring Response</i> | 3–34 |
| 4 | RESPONSE CODES – WORKING WITH THE RESPONSE | 4–36 |
| 5 | EDGEPAY PIVOT..... | 5–47 |
| | 5.1 DEVELOPMENT AND PRODUCTION URL's | 5–47 |
| | 5.2 THE IMPLEMENTATION | 5–47 |
| 6 | HOSTED PAYMENT PAGE | 6–58 |
| | 6.1 STEP 1: SETTING UP A HOSTED PAYMENT PAGE | 6–59 |
| | 6.1.1 <i>To setup the hosted payment page the merchant will</i> | 6–59 |
| | 6.2 STEP 2: PAYMENT BUTTON INTEGRATION..... | 6–61 |
| | 6.2.1 <i>To integrate the Payment button</i> | 6–61 |
| | 6.2.2 <i>Requesting the OTAC</i> | 6–62 |
| | 6.3 HOW HPP WORKS | 6–63 |
| 7 | SHOPPING CART INTEGRATION | 7–66 |
| 8 | EDGEPAY BATCH PAY | 8–67 |
| | 8.1 UPLOADING THE BATCH | 8–68 |
| | 8.1.1 <i>Drag and drop</i> | 8–70 |
| | 8.1.2 <i>Browse</i> | 8–70 |
| | 8.2 BATCHPAY REPORTS..... | 8–72 |
| | 8.2.1 <i>Report Download</i> | 8–75 |
| | 8.2.2 <i>Transaction Detail Report</i> | 8–75 |
| | 8.3 BATCHPAY FILE FORMAT..... | 8–76 |
| | 8.4 LIST OF SAMPLE BATCH FILES FOR DIFFERENT ACH OR CARD PAYMENT TYPES: | 8–79 |
| 9 | EDGEPAY RECURRING PAYMENT MANAGER | 9–82 |
| | 9.1 RECURRING PAYMENT FEATURES:..... | 9–82 |
| | 9.2 BENEFITS OF RECURRING BILLING: | 9–82 |
| | 9.3 ADDING BILLING PLANS | 9–83 |
| | 9.4 ADDING CUSTOMERS | 9–86 |
| | 9.5 RECURRING CUSTOMER ADD API | 1 |
| | 9.5.2 <i>The Recurring Customer Add JSON Body</i> | 1 |

| | | |
|-----------|--|----------|
| 10 | EDGEPAY FRAUD MANAGEMENT SERVICE..... | 4 |
| 11 | CERTIFICATION | 5 |
| 11.1 | TESTING AND CERTIFICATION INFORMATION | 5 |
| 11.2 | TEST TRANSASCTIONS..... | 5 |
| 12 | APPENDIX | 7 |
| 12.1 | SAMPLE TRANSACTIONS | 7 |

0 PREFACE

0.1 PURPOSE OF THIS DOCUMENT

- #1 *This document is to be used by the Development community to integrate Payments and payment processing into their software. By doing so, the developer will allow for payment transactions to occur in their work flow and software for the purposes of improving customer interaction and speed of the sale of goods and services. This is a technical document and while it will assist a larger audience in the capabilities of EdgePay, the primary use is for the development community to execute code to perform such integrations.*

0.2 CHANGES AND MODIFICAITONS

| Version | Date | Author | Changes and Modifications |
|---------|------------|-----------------------|--|
| 1.0 | 3/10/2018 | S. Hoofring | Original Draft |
| 1.1 | 4/11/2018 | S. Hoofring | Release Candidate |
| 1.2 | 5/16/2018 | W. Watley | Added Hosted Pay Page |
| 1.3 | 5/18/2018 | W. Watley | Added Recurring Payment Mgr. |
| 1.4 | 5/22/2018 | W. Watley | Added API Specification |
| 1.5 | 5/23/2018 | W. Watley | Expanded API to include message examples |
| 1.6 | 6/6/2018 | S. Hoofring | Added Pivot Functions |
| 1.7 | 6/11/2018 | S. Hoofring | Added Code examples and completed EdgePay Pivot |
| 1.8 | 6/19/2018 | S. Hoofring W. Watley | Response Code Table, HPP, eCommerce plug-In's |
| 1.9 | 6/22/2018 | S. Hoofring | Added Summary of all services |
| 1.10 | 7/16/2018 | S. Hoofring | Added Hosted Pay Page |
| 1.11 | 7/20/2018 | W. Watley | Updated Recurring Payment Manager |
| 1.12 | 08/22/2018 | R. Sato | Updated EdgePivot URLs |
| 1.13 | 9/17/2018 | S.Hoofring | Updated API with eCheck options and updates to captureToo method. Updates to Recurring. Updates to HPP. Update to Payment API with cvv2 element. |
| 1.14 | 9/20 | S.Hoofring | Revised Response Code mapping and tables. Updated AVS Response information. Updated Recurring API |

| | | | |
|------|------------|-----------|---|
| 1.15 | 10/12/2018 | W. Watley | Update Pivot |
| 1.16 | 03/11/219 | W. Watley | Added Payment Facilitator indicators Added Amex AVS Response Added PayFac Sample Transactions |
| 1.17 | 10/04/2019 | W. Watley | Added Make Recurring during Initial Sale Added RPM Customer Member Username, Member Password and Affiliate fields Added Make Recurring during Initial Sales and Refund with Token Sample Transactions |

1 SYSTEM OVERVIEW

1.1 GENERAL FEATURES AND OPTIONS

The EdgePay Payment Gateway is a Payment Services Platform that includes a complete set of Bank Card and ACH payment services for internet, retail, and mobile merchants and businesses.

The Services are flexible by way of the use of a concept called “Action Codes.” Action Codes enable industry or product specific features to allow for very basic to complex payment transactions. So, if your needs are for a simple sale type transaction you can enable it in a matter of minutes. If your needs are more complex, say a healthcare auto substantiation transaction then you will enable that Action Code and supporting data.

The system is a Host Based service, meaning that transactions will be processed financially for money movement based upon a daily cut off time. Each day all transactions that are ‘Captured’ will be presented to the Card Brands or the Federal Reserve to debit or credit the customer and pay the merchant account appropriately. Because the API allows for separate Authorization and Capture transactions you can manage the timing of the financial portion of the transaction when it is appropriate for your business.

The set of Application Protocol Interfaces, (API’s), are designed to allow you to integrate payments into your software application. As previously mentioned, they can be for Card Present or Card Not Present situations. If you wish to integrate Card Present, (Point of Sale or Software solution with a Card Reader), please contact our Developers Support Services before you begin your development.

One of the early decisions to consider is will you want to handle, (store, process, or transmit) sensitive data, (credit card numbers or checking accounts) on your server. If yes, the handling this information is governed by the Payment Card Industry. (More information may be obtained at www.pcisecuritystandards.org). This may require a formal audit of your services. If this is something you wish to avoid GET does provide other options such as EdgePay Pivot, a service that can be used in conjunction with this API and avoid PCI audit or certification needs and the risks of manging such data.

The EdgePay API is a RESTful web service utilizing a JSON format of representation. Messages are communicated via HTTP using the POST method.

In addition to providing credit card and ACH services, the API also extends a Recurring Payment Services, Tokenization Services, and Fraud Management Services.

For Recurring Payment Services, the API integrates with the EdgePay Recurring Payment Manager and provides for establishing and managing customers and their respective payment contracts.

For the Tokenization Services the API allows you to request a token, (replacement value), for both credit cards or bank accounts that you can store for future payment uses.

The EdgePay Fraud management Services work in conjunction with or without the use of the EdgePay Processing Center to help you detect risky transactions and make business decisions on acceptance and shipping of goods or services.

In addition to the API, EdgePay' s Processing Center provides for all the features described above through a set of user interfaces. The Processing Center also provides real-time reporting as well as options to manage fraud, users, products, and provides a Hosted Pay page option for integration.

1.2 AUTHENTICATION:

To Authenticate payment requests, Global Electronic Technology issues a Merchant Key. This key is included in the header of your POST transaction. You must protect this key and never share or expose it. We will use this value along with your Merchant ID to authenticate all transactions for processing.

1.3 HOST BASED, TIME INITIATED PROCESSING

The system is "Host Based, Time Initiated" for processing Clearing, (Collecting payments from the Card Holder or Check Writer), and Settlement, (Payment collected funds to the Merchant). At the merchant configured cut-off time, all transactions submitted as either 'Sales' or 'Authorized' transactions with corresponding 'Capture' transactions will be processed for payment.

1.4 DUPLICATE CHECKING

EdgePay uses the 'externalRefernceID' in the header and the 'merchantID' in the payload for duplicate checking. If you submit a duplicate transaction within a 3-day period, EdgePay will decline the transaction. We recommend a randomized externalRefernceID versus a serialized method.

1.5 TOKENIZATION

Tokens are unique reference numbers generated upon request and are sent back in the response in place of the credit card account number. Tokens can be used in subsequent transactions in lieu of the credit card number. You can generate tokens during the payment transaction or individually if a payment is not required.

1.6 DEVELOPMENT AND PRODUCTION URL'S

Development

| Function | URL |
|-----------|---|
| Payment | https://api.edgepay-uat.com/payment |
| Refund | https://api.edgepay-uat.com/refund |
| Token | https://api.edgepay-uat.com/token |
| Recurring | https://api.edgepay-uat.com/recurring/customer/add |

Production

| Function | URL |
|-----------|---|
| Payment | https://edgepayapi.com/payment |
| Refund | https://edgepayapi.com/refund |
| Token | https://edgepayapi.com/token |
| Recurring | https://edgepayapi.com/recurring/customer/add |

1.7 INTEGRATION SUPPORT

EdgePay has a dedicated team of integration specialists that are available to assist you through the integration process. They will also generate a merchant key and Id for you when you are ready to start coding. Please contact our team at developersupport@gettrx.com or call us at 888-775-1500

2 SUMMARY OF INTERFACES AVAILABLE AND THEIR USES

2.1 EDGEPAY INTERFACES

EdgePay exposes a list of interfaces, each of which may be used as a 'stand-alone' function or in combination to achieve a certain set of business objectives. Below is a list of each interface and the team's original thoughts on its use. You may find other uses or benefits that can be applied. Please feel free to contact us to discuss other options and uses as we have constructed the system with great flexibility and are always open to discussing your specific business needs.

- 2.1.1 EdgePay API** – A set of RESTful interfaces to allow for sending in Requests via HTTP 1.1 and receiving responses. The API supports all Payments, including ACH, Refunds of previous transactions, Recurring Payments, and Tokenization of Card or Check data.
- 2.1.2 EdgePay Pivot** – A Library and supporting utility that allows for the exchange of Sensitive Credit Card data at the Consumer's browser for API processing using a token in lieu of the Credit Card information.
- 2.1.3 EdgePay Hosted Pay Page** – A Payment function that routes the payment from your "Pay" button to our secure Hosted Payment page for inputting payment information, card or echeck. We handle all the processing and deliver a response to a page you design with a Redirect Post back for your server.
- 2.1.4 EdgePay Batch Pay** – A function within the Processing Center that allows you to upload a batch of payments to process. Using a simple csv file, you can deliver payments, (Ex. Capture transactions for all those items you shipped today) and we are processing in bulk, delivering responses to you back within the Processing Center reporting within minutes.
- 2.1.5 EdgePay Recurring Payment Manager** – Through the Processing Center you create Billing Plans that represent products or services or offers. Then you attached Contracts and Customers to those Billing Plans and we will bill the customers for you!
- 2.1.6 EdgePay Token Service** – Either through the Processing Center or via our RESTful API we provide you the ability to input either Credit Card, Debit Card, or Checking Account information and we return you a 16-digit token you can safely store for making future payment requests.
- 2.1.7 EdgePay eCommerce Plug In's** – Use one of our Plug-in's with your eCommerce site to support payment processing. The plug-in works in

conjunction with our Hosted Payment page and automates the features to simplify the start-up process.

- 2.1.8 EdgePay Fraud Management Service** – We provide you a tool set within our Processing Center to allow you to establish Static and Dynamic filters to analyse payment transactions while they are happening. Transactions caught by the filter will be displayed for your review. Additionally, we have an Enhanced version that can identify a computer to allow you to filter based on this dimension for decisioning.

3 EDGEPAY API, (APPLICATION PROTOCOL INTERFACE)

WORKING TOGETHER!

3.1.1 Introduction

The EdgePay Application Protocol Interface is a series of separate methods and associated URL's to process payment information. They are:

- Payment – Making a Payment request, which can include:
 - o Authorization Request
 - o Capture Request
 - o Sale Request (Both and Auth and Capture together)
 - o AVS Only
 - o CVV Only
 - o Adjust an Amount
 - o Card as well as ACH
- Refund – Reverse all or part of a previous transaction. They include”
 - o Reversing an Authorized Transaction
 - o Voiding a Captured Transaction that has not yet settled
 - o Updating the Amount on a Transaction that was not yet previously settled
 - o Crediting a Previously settled transaction
- Token – Converting a Credit Card or Checking Account information with a Token that may be used for you to store locally and then use later in the Payment, Refund, or Recurring API.
- Recurring – This interface allows you via API to add Contract and Customers to previously established Billing Program via our Processing Center. As well as adding recurring payment during the initial payment request

These interfaces are used by your server and systems to accept payments for goods and services your software or business supplies. Transactions processed via these interfaces are associated with a Merchant Account that has been previously implemented and approved by GET. Once the transactions are completed and approved, the GET Back-End systems will Clear the money, (taking it from the Card Holder or Check Writer) and Settle the money into the Merchant's Settlement Bank Account.

If you wish to get started please read on. But, once you are ready for Production you will need to call us for a quick certification and setup an account in Production. At that point we will begin to move money for you!

You can reach us at:

888.777.1500 – Developers Support

Or at Developersupport@gettrx.com

3.1.2 EdgePay Payment API

The Payment API is a RESTful and Stateless Application Protocol Interface that utilizes a HTTP Header for Authentication and a JSON body for the required data to generate a Payment Request. The system will always prepare a Payment Response and return it to your server.

The Payment API carries a deep, rich set of functions and uses a concept called "Action Codes." These Action Codes allow you to expand or contract your JSON Body according to either your specific needs or for the specific needs of a transaction. This way, you can construct the minimum data set required to process your transaction.

Before you begin, it may be important to consider a few questions to help guide your work. We find by answering these few questions first it greatly helps in selecting the right strategy and build the right tactical implementation.

3.1.3 Before You Begin Q&A

The Payment function has evolved over 50+ years. Some of that change has occurred because of new technology, (Like the Internet, or the ATM Machine). Others have occurred as the Card Brands began to look at their customers differently. This combination of dimensions can create some uncertainty in choosing the best path. Below are some topics with insights to help guide you.

| TOPIC | DISCUSSION |
|--|---|
| Authorize and Capture or Sale? | The Credit Card system is a 2-message system with Authorization preceding Capture. Money only moves after Capture. There are some good reasons for a 2-message system like Tip Adjust or if you ship your goods or deliver your services later then the Authorization Date/Time. But, if everything happens all at once, consider a Sale. With this message we will automatically generate the Capture for you using a pre-determined cut off time. |
| Tokenization | Do you want to store Credit Card or Banking information on your server? If the answer is no then request a Token, either separately or in the Payment request and we will generate a Token for you and store the Card or Checking Account information for future reference. Many customers place this on their Customer Information and then enable the users or systems to reuse the Token for processing future 1-time or recurring payments. |
| Verification Only | Sometimes verifying a Card by using Address Verification, (AVS) or the Security Code (CVV) can be helpful to your business. These a zero-amount transaction but, tell you that the Card Holder knows their zip code or Security Code and positions you for a Payment with more confidence later. |
| Don't know the Totals yet? | Sometimes the Total Amount cannot be determined at Authorization. You can adjust the amount with the Capture transaction or, you can also use the Adjust function to set the amount properly. Think about your work flow then design a pattern that works best for you! |
| I don't want any Card Data | If you present to users in a Browser and do not want to handle Card data, then consider EdgePay Pivot. This Library/Utility will replace the Card Data even before it hits your server. |
| Are there any special data needs for my business? | Some businesses and in some cases some transactions will benefit from more information in the transaction. These data enhancements work to reduce Interchange, the fees the Card Brands charge all Merchants for Card Acceptance. |

| | |
|--|---|
| | <p>eCommerce or Keyed Transactions – To get the best Interchange rate for transactions from the internet or, keyed such as Mail Order or Telephone Order you must include the Billing Address, Billing City, Billing State, and Billing Zip.</p> <p>Additionally, you can add the Security Code (CVV) for improving the Authorization and rates.</p> |
| | <p>Recurring Transactions – If you submit transactions in a recurring basis such as a monthly membership use the Recurring Action Code and Fields to optimize both Interchange and Approval Rates. Issues, we find, view Recurring transactions slightly different then 1-time transactions.</p> |
| | <p>HealthCare Transactions – If you take FSA or HSA Credit Card (Health Savings Accounts), then you should consider the HealthCare Action Code and supporting fields.</p> |
| | <p>Level 2 – If you accept Business Credit Cards or Purchasing Cards, you can include Tax amount and Po Number to qualify for up to ½% savings on Interchange rates for those Cards</p> |
| | <p>Level 3 – Much Like Level 2, Level 3 will save even more on Interchange for Business to Business transactions. The amount of data for level 3 is much bigger and typically involves a System to System approach. If you process B2B and can map Level 3, there is good opportunity for savings!</p> |
| | <p>Tips – If your business takes tips then Consider and Auth and Capture and adjust the Amount either on the Capture or, as an Adjustment that will mark the transaction as Captured.</p> |

CONCLUSIONS: For some circumstances you may choose to solve your Payment needs through multiple interfaces:

- You can use Pivot to gather a key via browser and avoid Card Data
- You can use the Payment API to generate Authorizations, Captures, Tokens, or Recurring payments
- You can use Processing Center to Capture transactions after they have been shipped or through the API

- Use BatchPay to upload Credits nightly from the system and let us process over night

Make yourself familiar with all the interfaces and tools and select the right combination that most benefits your customers!

3.2 PAYMENT API

Below is the layout for the Payment Request Header

3.2.1.1 Payment Header

| Header | Character | Min - Max | Notes |
|--|-----------|-----------|---|
| POST /payment HTTP/ 1.1 | AN | | Current HTTP version |
| HOST: api.edgepay-uat.com/payment | AN | | The API Request location |
| Content-Type: application/json | AN | | |
| merchantKey: E3DFE3D0150EB516F4E351CE7FE5A2FB7 99B320BDC5D523B | AN | 48 | This is a TEST KEY. Please use for testing purposes only |
| externalReferencelD: 1234561523473261 | N | 16 - 19 | Create your own ID using your phone number and increment! |
| transactionDate: 2018-05-23T02:00:00 | Calendar | 20 | Reset this to today UTC! |

3.2.1.2 The Payment JSON Body - The Body of the Payment request will vary according to your specific business needs. Some items marked as **BOLD** are required but the others are Conditional or Optional based on your specific business or transaction needs. Let start with a few simple examples:

Simple Manual Entry Authorization Transaction with cURL.

Feel free to update the externalReferencelD using your phone number and increment and the transactionDate to today UTC

```

1. curl -X POST\ https://api.edgepay-uat.com/payment \
   -H 'Cache-Control: no-
   cache' -H 'externalReferencelD: 1234561528752675' -H 'merchantKey: E3DFE3D0150EB516F4E351CE7FE5A2FB79
   9B320BDC5D523B' -H 'transactionDate: 2018-06-11T02:00:00' - d '{
2. "merchantID": "6320340123456900", "terminalID": "88800000282601", "paymentDataInput": "Manualentry", "amount": "1.
   00", "cardNumber": "4012000098765439", "cardExpirationDate": "1128", "billingZip": "83202"
3. }
4. '
   }
```

Here is a Java Example:

```
1. OkHttpClient client = new OkHttpClient();
2. MediaType mediaType = MediaType.parse("application/octet-stream");
3. RequestBody body = RequestBody.create(mediaType, "{ \r\n  \"merchantID\": \"6320340123456900\", \r\n  \"terminalID\": \"88800000282601\", \r\n  \"paymentDataInput\": \"Manualentry\", \r\n  \"amount\": \"1.00\", \r\n  \"cardNumber\": \"4012000098765439\", \r\n  \"cardExpirationDate\": \"1128\", \r\n  \"billingZip\": \"83202\" \r\n}");
   Request request = new Request.Builder().url("https://api.edgepay-
   uat.com/payment").post(body).addHeader("merchantKey", "E3DFE3D0150EB516F4E351CE7FE5A2FB799B320BDC5D523B
4. ").addHeader("externalReferenceID", "1234561528754657").addHeader("transactionDate", "2018-06-
   11T02:00:00").addHeader("Cache-Control", "no-cache").build();
5. Response response = client.newCall(request).execute();
```

Here is a Python Example

```
import http.client conn = http.client.HTTPConnection("api.edgepay-
uat.com") payload = "{ \r\n  \"merchantID\": \"6320340123456900\", \r\n  \"terminalID\": \"88800000282601
1. \", \r\n  \"paymentDataInput\": \"Manualentry\", \r\n  \"amount\": \"1.00\", \r\n  \"cardNumber\": \"4012000098765439\", \r\n  \"c
   ardExpirationDate\": \"1128\", \r\n  \"billingZip\": \"83202\" \r\n}"
2. headers = {
   'merchantKey': "E3DFE3D0150EB516F4E351CE7FE5A2FB799B320BDC5D523B
3. ",
4. 'externalReferenceID': "1234561528754878",
5. 'transactionDate': "2018-06-11T02:00:00",
6. 'Cache-Control': "no-cache",
7. }
8. conn.request("POST", "payment", payload, headers) res = conn.getresponse() data = res.read() print(data.decode("utf-
   8"))
```

3.3 FULL PAYMENT API TABLE

Here is the full API for the Payment Interface. **ITEMS IN Bold are required.**
All others are Conditional or Optional based on your use and need.

| Name | Character | Max | Comments |
|---------------------------|-----------|-----|--|
| merchantID | N | 19 | Will be supplied by GET at Boarding |
| terminalID | N | 19 | Provided by GET as part of Merchant Boarding. Required Field. |
| cardNumber | N | 19 | Keyed Card |
| cardExpirationDate | N | 4 | MMYY – Provided for Keyed card only |
| trackData | AN | 76 | Track 1 or Track 2 unencrypted – Swipe or EMV |
| tokenID | N | 16 | Generated by GET at Request |
| paymentDataInput | AN | 14 | Define the Point of Sale Card Entry Mode. <ul style="list-style-type: none"> - Phone (Card Not Present) - Mail (Card Not Present) - Internet (Card Not Present) - Swipe - NFC - EMV - EMVContactless - Fallbackswipe - Barcode - Manualentry |

| | | | |
|--------------------------------|------------------|------------|--|
| eCheckRoutingNumber | N | 9 | Optional – Required for eCheck |
| eCheckAccountNumber | N | 19 | Optional – Required for eCheck |
| eCheckTransactionType | A | 3 | Optional – Required for echeck. See table |
| eCheckSerialNumber | AN | 15 | Conditional – Required for ACH ARC, POP |
| eCheckAccountType | AN | 1 | Defines the Receivers account type C – Checking S – Savings G – General Ledger |
| eCheckReceivingCompany Name | A | 50 | Receiving Company Name Required for CCD transactions |
| eCheckReceiverName | ANC | 50 | Receiver Name or Customer Name is Required for ARC transactions |
| eCheckPaymentTypeCode | A | 1 | |
| amount | N | 12 | With decimal or without. All minor units after decimal. No decimal provided the amount will be processed as major units, (Dollars, Euros, etc) |
| currencyCode | N | 3 | Optional - Multi-Currency Code. Use ISO Currency Code Numbers for Validation |
| PayFac Fields | Character | Max | Comments |
| siD | | 15 | When TYPE is SubMerchant use MID on merchant record as siD (SubMerchant MID will be used to locate merchant record in Edgepay to populate Submerchants fields for the TSYS request) |
| acctSplits | | Array | Settlement Split Account and amount that funds will be settled to. "acctSplits":[{"accountID": 1, "amount": 1.00}, {"accountID": 1, "amount": 1.00}] |

| | | | |
|------------------------|-----|-----|--|
| | | |] |
| Optional Fields | | | |
| customerName | AN | 30 | Conditional – Required for Recurring Payment Customer and eCheck processing |
| customerNumber | AN | 17 | Conditional – Customer Order Number, see below for use. Required for Recurring Payment Customer and eCheck CCD |
| processor | AN | 10 | Optional - Used for multiple processors |
| billingAddress | AN | 200 | Conditional - Needed if card data manually keyed. |
| billingCity | AN | 50 | Conditional – Needed if card data manually keyed. |
| billingState | A | 2 | Conditional – Needed if card data manually keyed. |
| billingZip | AN | 10 | Conditional – Needed if card data manually keyed. Space, “-“ allowed |
| Cvv2 | N | 4 | Optional – If provided will return CVV2 Response |
| merchantField1 | AN | 200 | Optional - Returned in Response |
| merchantField2 | AN | 200 | Optional – Returned in Response |
| posData | | | Future sig.cap Reference ICO-250 Spec 6144 |
| goodsOrService | AN | 10 | Optional Description for ACH processing |
| memberUsername | AN | 60 | Optional - Recurring Payment Manager Customer |
| memberPassword | ANS | 60 | Optional - Recurring Payment Manager Customer |
| siteUrl | ANS | 100 | Optional - Recurring Payment Manager Customer |
| affiliate | ANS | 12 | Optional - Recurring Payment Manager Customer |
| cardholderIPAddress | ANS | 39 | Optional – Enhanced Fraud Management Service IPv6 supported |
| eMailAddress | ANS | 128 | Customer's email address |
| Action Fields | | | |
| recurring | A | 3 | “yes” to Set the Recurring Flag for Recurring payment. Conditional to ACH Sec Code TEL |
| makeRecurring | A | 3 | “yes” This flag can only be used in conjunction with the “captureToo” flag set to “yes.” the initial sale |
| avsOnly | A | 3 | “yes” to Perform AVS Only Transaction |
| cvvOnly | A | 3 | “yes” to Perform Security Check Only Transaction |
| adjust | A | 3 | “yes” to Adjust the amount on a transaction to increase or decrease settlement. Includes TIP |
| healthCare | A | 3 | “yes” to utilize an FSA/HSA Card and qualify |
| level2 | A | 3 | “yes” to Support charge functions with appended data |
| level3 | A | 3 | “yes” to Support charge functions with appended data |

| | | | |
|----------------------------|------------------|------------|---|
| Token | A | 3 | "yes" to return a Token Value |
| captureOnly | A | 3 | "yes" will update Auth for Capture with no changes. Requires Authcode. SEE NOTE BELOW FOR SPECIAL CONDITIONS. |
| captureToo | A | 3 | "yes" Will Auth, and if successful, will mark the transaction for settlement as well. (SALE) |
| payFac | A | 3 | "yes" to Support charge functions with appended data |
| Recurring | Character | Max | |
| First Payment | | | For the initial payment on a recurring transaction only the cardOnFile indicator |
| cardOnFile | A | 1 | <p>"y" Indicates if the transaction is a Card On File (COF) or Merchant Initiated Transaction (MIT) type. Use for 1st payment only.</p> <p>Required to be used with Visa only if the card being used is being put on file for future Merchant Initiated Transactions (i.e. Recurring and Installments)</p> |
| Subsequent Payments | | | For all Subsequent payments use the Recurring Action Code and associated sub fields as needed. |
| billingType | A | 25 | <p>Required - The type of the billing schedule opted by a customer for recurring payment.</p> <p>Allowed values:</p> <ul style="list-style-type: none"> • INSTALLMENT • FIXED_AMOUNT_RECURRING • SINGLE_INSTALLMENT • VARIABLE_AMOUNT_RECURRING • ONE_TIME_RETURN |
| paymentCount | N | 4 | <p>Conditional – Required for Payments if billingType = INSTALLMENT. Otherwise, not required. The count of payments in a recurring schedule payment, set by a customer.</p> <p>Note: Send either the endDate or the paymentCount in a request.</p> |
| currentPaymentCount | N | 3 | Conditional – Required for Payments if billingType = INSTALLMENT. Otherwise, not required. Indicates the current recurring payment count |
| Make Recurring | | | |
| description | ANS | 96 | Required - Text in description field |
| schedulePaymentType | | | Required - Subscriptions Installments |
| paidIn | N | 3 | Conditional – Number of payments (When Schedule Payments Type = Installments) |

| | | | |
|------------------|----|----|---|
| startDate | | | Required |
| endDate | | | Optional |
| totalAmount | N | 12 | Conditional – |
| recurringAmount | N | 12 | Amount field |
| frequency | AN | 2 | Required - D = Numbers of Days, O = Once, W = Weekly, M=Monthly, Q = Quarterly, A = Annual, |
| eMailReceipt | N | 1 | Default yes |
| recurringID | | | The recurring transaction assigned to this Customer returned in response |
| avsOnly | A | 3 | “yes” in the field, amount field must be left off request. |
| Zip | AN | 10 | Send Customer ZIP Space, “-“ Allowed |
| cvvOnly | A | 3 | “yes” in the field, amount field must be left off request. |
| cvv2 | N | 4 | Send keyed Cvv information |
| adjust | A | 3 | “yes” to Adjust the amount prior to Settlement. Marks Auth as Captured. |
| amount | N | 12 | Send if Amount is different from Auth. With decimal or without. All minor units after decimal. No decimal provided the amount will be processed as major units, (Dollars, Euros, etc) |
| tip | N | 12 | Send if Tip added after Auth. Send new Amount too. With decimal or without. All minor units after decimal. No decimal provided the amount will be processed as major units, (Dollars, Euros, etc) |
| tax | N | 12 | Send if Sales Tax is added With decimal or without. All minor units after decimal. No decimal provided the amount will be processed as major units, (Dollars, Euros, etc) |
| healthCare | N | 16 | For Healthcare Auth, must use an FSA/HSA card. Visa and MC only. Must contain a Visa Merchant Verification Value,(Registration process added to Merchant record) MCC must be valid for Healthcare. Partial Auth must be supported. Both CP/CNP supported. |
| paymentDataInput | AN | 14 | Define the Point of Sale Card Entry Mode. For FSA/HSA only allowable entries are: <ul style="list-style-type: none"> - Phone (Card Not Present) - Mail (Card Not Present) - Internet (Card Not Present) - Manualentry |

| | | | |
|----------------------|----|----|--|
| healthCareAccoutType | AN | 3 | HSA or FSA required |
| prescriptionAmt | N | 12 | With decimal or without. All minor units after decimal. No decimal provided the amount will be processed as major units, (Dollars, Euros, etc) |
| visionAmt | N | 12 | With decimal or without. All minor units after decimal. No decimal provided the amount will be processed as major units, (Dollars, Euros, etc) |
| dentalAmt | N | 12 | With decimal or without. All minor units after decimal. No decimal provided the amount will be processed as major units, (Dollars, Euros, etc) |
| clinicAmt | N | 12 | With decimal or without. All minor units after decimal. No decimal provided the amount will be processed as major units, (Dollars, Euros, etc) |
| posQualified | AN | 6 | “yes”, “no”, or “exempt” Purchase is qualified by POS |
| rxNumber | AN | 19 | Optional – Customer number |
| level2Auth | A | 3 | “yes” In addition to “Auth only” supply the following |
| tax | N | 12 | Send if Sales Tax is added. With decimal or without. All minor units after decimal. No decimal provided the amount will be processed as major units, (Dollars, Euros, etc) |
| purchaseOrder | AN | 17 | Buyers Order Number |
| level2Capture | A | 3 | “yes” In addition to “Captureonly” supply the following. |
| tax | N | 12 | Send if Sales Tax is added. With decimal or without. All minor units after decimal. No decimal provided the amount will be processed as major units, (Dollars, Euros, etc) |
| purchaseOrder | AN | 17 | Buyers order Number |
| level3Auth | A | 3 | “yes” In addition to “Authonly” supply the following |
| tax | N | 12 | Send if Sales Tax is added. With decimal or without. All minor units after decimal. No decimal provided the amount will be processed as major units, (Dollars, Euros, etc) |
| purchaseOrder | AN | 17 | Buyers order Number |
| shipZipCode | AN | 10 | The Ship from Zip Space, “-“ allowed |
| destinationZipCode | AN | 10 | The Ship to Zip Code Space, “-“ allowed |
| customerNumber | AN | 17 | Customer ID Number |
| itemCode | AN | 20 | |
| itemCommunityCode | AN | 12 | |
| ItemDescription | AN | 50 | |
| itemQuantity | N | 12 | |

| | | | |
|------------------------|------|----|--|
| itemUnitOfMeasure | AN | 20 | Call Global Electronic Transactions for table |
| itemExtendedAmount | N | 21 | With decimal or without. All minor units after decimal. No decimal provided the amount will be processed as major units, (Dollars, Euros, etc) |
| freightAmount | N | 12 | With decimal or without. All minor units after decimal. No decimal provided the amount will be processed as major units, (Dollars, Euros, etc) |
| dutyAmount | N | 12 | With decimal or without. All minor units after decimal. No decimal provided the amount will be processed as major units, (Dollars, Euros, etc) |
| PurchaseOrder | AN | 17 | Buyers order Number |
| ChargeDescriptor | AN | 40 | Amex Additional Descriptor |
| CustomerVATNumber | AN | 13 | Government Tax ID |
| CustomerReferenceID | AN | 17 | Buyer Reference Number |
| SummaryCommodityCode | AN | 4 | Call Global Electronic Transactions for table |
| VatInvoice | AN | 15 | VAT Invoice Number |
| taxCategory | AN | 10 | The type of tax. SERVICE DUTY VAT ALTERNATE NATIONAL TAX_EXEMPT |
| taxAmount | N | 12 | With decimal or without. All minor units after decimal. No decimal provided the amount will be processed as major units, (Dollars, Euros, etc) |
| taxRate | N | 12 | With decimal or without. |
| externalReferenceID | N | 19 | reference id for transaction, part of header, generated at merchant end |
| transDiscountName | AN | | The name of the discount applied to the transaction. This field is used for reporting. |
| transDiscountAmount | N | 12 | The dollar amount of discount applied to a transaction. Corresponds with transDiscountAmount. This field is used for reporting. With decimal or without. All minor units after decimal. No decimal provided the amount will be processed as major units, (Dollars, Euros, etc) |
| destinationCountryCode | AN | 3 | The destination country code indicator. |
| itemDiscountName | ANCA | 50 | The name of the discount applied to a product. This does not impact transaction functionality. It is used for reporting purposes. |
| itemDiscountAmount | N | 12 | The dollar amount of discount applied to a product. Corresponds with productDiscountName. This does not impact transaction functionality. It is used for reporting purposes. With decimal or |

| | | | |
|------------------------|----|----|--|
| | | | without. All minor units after decimal. No decimal provided the amount will be processed as major units, (Dollars, Euros, etc) |
| itemDiscountPercentage | N | 3 | The discount percentage applied to a product. Corresponds with productDiscountName. This does not impact transaction functionality. It is used for reporting purpose |
| itemDiscountType | AN | 20 | This field defines the transaction types that the discount can be applied to. Corresponds with productDiscountName. This does not impact transaction functionality. It is used for reporting purposes. |
| transactionID | AN | 17 | "CS" for card "AS" for ACH for Generated by EdgePay, used for capture, adjust and refund |
| level3Capture | A | 3 | "yes" In additions 'Captureonly" supply the following: |
| shipZipCode | AN | 10 | The Ship from Zip Space, "-" allowed |
| destinationZipCode | AN | 10 | The Ship to Zip Code Space, "-" allowed |
| customerNumber | AN | 25 | Customer ID Number |
| itemCode | AN | 20 | |
| itemCommunityCode | AN | 12 | |
| ItemDescription | AN | 50 | |
| itemQuantity | N | 12 | |
| itemUnitOfMeasure | AN | 20 | Call Global Electronic Transactions for table |
| itemExtendedAmount | N | 21 | With decimal or without. All minor units after decimal. No decimal provided the amount will be processed as major units, (Dollars, Euros, etc) |
| freightAmount | N | 12 | With decimal or without. All minor units after decimal. No decimal provided the amount will be processed as major units, (Dollars, Euros, etc) |
| dutyAmount | N | 12 | With decimal or without. All minor units after decimal. No decimal provided the amount will be processed as major units, (Dollars, Euros, etc) |
| PurchaseOrder | AN | 17 | Buyers order Number |
| ChargeDescriptor | AN | 40 | Amex Additional Descriptor |
| CustomerVATNumber | AN | 13 | Government Tax ID |
| CustomerReferenceID | AN | 17 | Buyer Reference Number |
| SummaryCommodityCode | AN | 4 | Call Global Electronic Transactions for table |
| VatInvoice | AN | 15 | VAT Invoice Number |
| taxCategory | AN | 10 | The type of tax. SERVICE DUTY VAT ALTERNATE NATIONAL |

| | | | |
|------------------------|------|-----|---|
| | | | TAX_EXEMPT |
| taxAmount | N | 12 | With decimal or without. All minor units after decimal. No decimal provided the amount will be processed as major units, (Dollars, Euros, etc) |
| taxRate | N/12 | 12 | With decimal or without. |
| externalReferenceID | N | 19 | reference id for transaction, part of header, generated at merchant end |
| transDiscountName | AN | 30 | The name of the discount applied to the transaction. This field is used for reporting. |
| transDiscountAmount | N | 12 | The dollar amount of discount applied to a transaction. Corresponds with transDiscountAmount. This field is used for reporting. With decimal or without. All minor units after decimal. No decimal provided the amount will be processed as major units, (Dollars, Euros, etc) |
| destinationCountryCode | AN | 3 | The destination country code indicator. |
| itemDiscountName | AN | 50 | The name of the discount applied to a product. This does not impact transaction functionality. It is used for reporting purposes. |
| itemDiscountAmount | N | 12 | The dollar amount of discount applied to a product. Corresponds with productDiscountName. This does not impact transaction functionality. It is used for reporting purposes. With decimal or without. All minor units after decimal. No decimal provided the amount will be processed as major units, (Dollars, Euros, etc) |
| itemDiscountPercentage | N | 100 | The discount percentage applied to a product. Corresponds with productDiscountName. This does not impact transaction functionality. It is used for reporting purpose |
| itemDiscountType | AN | 20 | This field defines the transaction types that the discount can be applied to. Corresponds with productDiscountName. This does not impact transaction functionality. It is used for reporting purposes. |
| transactionID | AN | 17 | "CS" for card "AS" for ACH for Generated by EdgePay, used for capture, adjust and refund |
| Token | A | 3 | "yes" Returns a Token Value for local storage. No additional Sub Fields required. |

| | | | |
|---------------|----|----|--|
| captureOnly | A | 3 | <p>“yes” Conditional – Will update a previous Authorization for Capture.</p> <p>Required fields are:</p> <ul style="list-style-type: none"> - Capture Only Flag - Merchant ID - Terminal ID - Transaction ID - Amount, - Optional - (If different from Auth) - Currency Code - Optional - Tip, (If Different from Auth) |
| merchantID | N | 19 | Will be supplied by GET at Boarding |
| terminalID | N | 19 | Provided by GET as part of Merchant Boarding. Required Field. |
| transactionID | AN | 17 | “CS” for card “AS” for ACH for Generated by EdgePay, used for capture, adjust and refund |
| amount | N | 12 | Optional – Input if Amount is different than the Authorization Amount. With decimal or without. All minor units after decimal. No decimal provided the amount will be processed as major units, (Dollars, Euros, etc) |
| currencyCode | N | 3 | Optional - Multi-Currency Code. Use ISO Currency Code Numbers for Validation |
| tip | N | 12 | Send if Tip added after Auth. Send new amount also. With decimal or without. All minor units after decimal. No decimal provided the amount will be processed as major units, (Dollars, Euros, etc) |
| captureToo | A | 3 | <p>“yes” will Auth, and if successful, will mark the transaction for settlement as well. (SALE). No additional Sub Fields required.</p> |

3.3.1 Refund

URL: <https://api.edgepay-uat.com/refund>

A refund is used to credit back a previously authorized or captured transaction. This object will check the status of the original item using the transactionID if the transactionID is present. Refund can also be sent without a transactionID using the Card information. This must be allowed by GET and configured during the Merchant setup process. Because the refund is not tied to any transaction in this event, a credit will be generated to the card holder's card a no checking is done on previous transactions.

- If transaction is captured, but not settled a void will be sent
- If an amount less than the original amount is sent, then an adjust is made to the original transaction.
- If the transaction was authorized but not captured a reversal will be send
- If the transaction is already settled a return, (Credit) is processed.

***Bold is Required**

| NAME | ENUMERATION | COMMENTS |
|--------------------|-------------|---|
| merchantKey | As defined | As defined - Header |
| merchantID | N/19 | Will be supplied by GET at Boarding |
| terminalID | N/19 | From original transaction |
| transactionID | AN/19 | Conditional – Either the transactionID or the Card data is required. Unique number assigned to each transaction by EdgePay |
| cardNumber | N/19 | Conditional – Either Card data or transactionID must be present. Keyed Card |
| cardExpirationDate | N/4 | Conditional - MMY – Provided for Keyed card only |
| trackData | AN/76 | Conditional - Track 1 or Track 2 unencrypted – Swipe or EMV |
| tokenID | N/16 | Card or Checking Account token |
| amount | N/12 | Conditional – If the Amount to Refund is different than amount in transactionID or if the Card data is supplied in lieu of the transactionID, then amount is required. With decimal or without. All minor units after decimal. No decimal provided the amount will be processed as major units, (Dollars, Euros, etc) |

3.3.2 Token Request

URL: <https://api.edgepay-uat.com/token>

Tokens are unique reference numbers generated upon request and are sent back in the response in place of the credit card account number or checking route/account combinations. Tokens can be used in subsequent transactions in lieu of the credit card number or checking account. You can generate tokens during the payment transaction or individually using this API if a payment is not required.

***Bold** is Required

| Name | Character | Max | Comments |
|---------------------------|-----------|-----|---|
| merchantID | N | 19 | Will be supplied by GET at Boarding |
| terminalID | N | 19 | Required field. Provided by GET. |
| cardnumber | N | 19 | Keyed Card |
| cardExpirationDate | N | 4 | MMYY – Provided for Keyed card only |
| trackData | AN | 76 | Track 1 or Track 2 unencrypted – Swipe or EMV |
| eCheckRoute | N | 9 | Optional – Required for eCheck |
| eCheckAccount | N | 17 | Optional – Required for eCheck |

3.3.3 Recurring Payments

URL: <https://api.edgepay-uat.com/recurring/customer/add>

Recurring transactions allow consumers to authorize merchants to charge their credit card or bank account a specific amount on a regular basis such as weekly, monthly or quarterly for a series of goods or regular services. This transaction supports both add and change functionality.

Please note that the contract number must be setup by the merchant in the EdgePay Portal prior to adding a new card.

Example:

***Bold** is required

| Name | Character | Max | Comments |
|-------------------|-----------|-----|-------------------------------------|
| merchantID | N | 19 | Will be supplied by GET at Boarding |
| terminalID | N | 19 | Required. Provided by GET |

| | | | |
|----------------------------|-----|-----|---|
| actionCode | A | 1 | A = Add, C = Change. If Change, only supply those elements below that need to be updated. |
| cardNumber | N | 19 | Keyed Card |
| cardExpirationDate | N | 4 | MMYY – Provided for Keyed card only |
| trackdata | AN | 76 | Track 1 or Track 2 unencrypted – Swipe or EMV |
| tokenID | N | 16 | Provide in lieu of Card or eCheck data. Generated from a previous transaction. |
| eCheckRoute | N | 9 | Optional – Required for eCheck |
| eCheckAccount | N | 17 | Optional – Required for eCheck |
| eCheckTransType | A | 3 | Optional – Required for echeck. See table |
| eCheckAccountType | AN | 1 | Defines the Receivers account type C – Checking S – Savings G – General Ledger |
| token | A | 3 | “yes” to return a Token Value |
| customerName | AN | 30 | |
| customerNumber | ANC | 25 | The Customer Number you assign |
| billingAddress | AN | 200 | |
| billingCity | AN | 50 | |
| billingState | A | 2 | |
| billingZip | AN | 10 | Space, “-“ allowed |
| shippingAddress | AN | 200 | |
| shippingCity | AN | 50 | |
| shippingState | A | 2 | |
| shippingZip | AN | 10 | Space, “-“ allowed |
| phoneNumber | N | 15 | Customer’s Phone Number |
| eMailAddress | A | 128 | Customer’s email address |
| paymentDataInput | AN | 14 | Define the Point of Sale Card Entry Mode. - Phone (Card Not Present) - Mail (Card Not Present) - Internet (Card Not Present) - Swipe - NFC - EMV - EMVContactless - Fallbackswipe - Barcode - Manualentry |
| customerCurrency | N | 3 | Default to USD, (840) – See Table |
| chargebackShutOff | A | 1 | Y/N If Chargeback received, set Contract Status to Suspend |
| eMailReceipts | A | 1 | Y/N Send Receipt of payment result |
| eMailSchdulePayment | A | 1 | Y/N Send Notice of scheduled payment |
| eMailContractEnd | A | 1 | Y/N Send Notice of End of Contract |
| eMailContractCancel | A | 1 | Y/N Send Notice of Cancelled Contract |
| taxable | A | 1 | Y/N Customer is Taxable |
| taxRate | N | 3 | To Calculate Taxable Amount for Payment |
| contractNumber | N | 20 | Unique Number - Assigned to a Customer by EdgePay. Only send for change requests. |
| contractDescription | A | 30 | Name assigned to Contract |
| status | A | 1 | A to Activate, S to Suspend this contract |

| | | | |
|------------------|---|----|---|
| billingPlan | N | 30 | Billing plan pre-setup in the Processing Center prior to adding or changing a customer/contract |
| startDate | N | 25 | Required |
| endDate | N | 25 | Optional – If no end date, system will continue to bill based on the schedule |

3.3.4 ACH

URL: <https://api.edgepay-uat.com/payment>

ACH payments are electronic payments that are created when the customer gives a merchant authorization to debit directly from the customer's checking or saving account for the purpose of payment.

***Bold** is required

| Name | C h a r a c t e r | Max | Comments |
|--|---|-----|---|
| merchantID | N | 19 | Will be supplied by GET at Boarding |
| terminalID | N | 19 | Provided by GET as part of Merchant Boarding. Required Field. |
| tokenID | N | 16 | Generated by GET at Request |
| eCheckRoutingNumber | N | 9 | Bank Routing number |
| eCheckAccountNumber | N | 19 | Account number issued to customer |
| eCheckTransactionType | A | 3 | See Code Table |
| eCheckSerialNumber | A N | 15 | Conditional – Required for ACH ARC, BOC, POP |
| eCheckAccountType | A N | 1 | Defines the Receivers account type C – Checking S – Savings G – General Ledger |
| eCheckReceivingCompany nyName | A | 50 | Receiving Company Name Required for CCD transactions |
| eCheckReceiverName | A N | 50 | Receiver Name or Customer Name is Required for ARC transactions |
| eCheckPaymentTypeCo de | A | 1 | Defines the Payment Type Code S – Single R – Recurring |

| | | | |
|-------------------------|--------|----|---|
| paymentDataInput | A N | 14 | Define the Point of Sale Card Entry Mode. <ul style="list-style-type: none"> - Phone (Card Not Present) - Mail (Card Not Present) - Internet (Card Not Present) - Swipe - NFC - EMV - EMVContactless - Fallbackswipe - Barcode - Manualentry |
| amount | N | 12 | Total charge amount. With decimal or without. All minor units after decimal. No decimal provided the amount will be processed as major units, (Dollars, Euros, etc) |
| currencyCode | N | 3 | Optional - Multi-Currency Code. Use ISO Currency Code Numbers for Validation |
| customerName | A N | 30 | Conditional – Required for eCheck |
| customerNumber | A N | 17 | Conditional – Customer Order Number, see below for use. Required for eCheck CCD |
| goodsOrService | A N | 10 | Optional Description for ACH processing |

3.4 TRANSACTION RESPONSES

The transaction responses generated by EdgePay contains non-sensitive data that can be used to validate the transaction and confirm your next step in your software work flow, build receipts or build records for internal reporting and storing for subsequent use.

Prepaid cards can receive partial approvals for less than the originally authorized amount. The amount that was authorized will appear in the amount field.

3.4.1.1 EdgePay AP Response Messages

The EdgePay API Response Header contains general HTTP standards that your server will expect for processing. The body contains the information relevant to your processing needs.

The responses contain non-sensitive data that you will use to validate the status of your request and to execute your next steps in your software flow logic. This may be updating the customer with the results, printing a receipt, asking the customer to fix something, saving critical data, updating your CRM order status, General Ledger, Accounts Receivable, etc. Here is an example of our HTTP header and JSON response Object.

Header:

Connection →keep-alive
Content-Encoding →gzip
Content-Type →application/json;charset=UTF-8
Date →Mon, 11 Jun 2018 18:43:10 GMT
Transfer-Encoding →chunked
Vary →Accept-Encoding

JSON Body:

```
{
  "result": "A",
  "responseCode": "A0000",
  "responseMessage": "Success",
  "merchantID": "6320340123456900",
  "terminalID": "88800000282601",
  "sID": "222212345678",
  "timestamp": "2018-06-11-18-43-10",
  "authCode": "TAS632",
  "processor": "TSYS",
  "amount": "1.00",
  "maskedAccount": "401200*****5439",
  "cardExpirationDate": "1128",
  "transactionID": "CS93436593469381642",
  "avsResponse": "Neither the street address or ZIP code match what is
on file.",
}
```

We have implemented a feature called ‘Result.’ This value will assist you in the appropriate UI Work flow. See section 4 ‘Response Codes’ for more details on your “Next Steps.”

More samples of the API can be found in the Appendix.

3.4.2 Payment Response

| Name | Character | Max | Comments |
|----------------|-----------|-----|---|
| result | A | 1 | A = Approved. D=Decline. R=Retry. C=Call for help |
| securityResult | A | 1 | “A” - Approved, “D” - Declined or ‘R’ – Review. When a transaction is under review the merchant is required to go to EdgePay’ s portal to review the transaction for approval or decline. |

| | | | |
|---------------------|----|-----|---|
| authCode | AN | 6 | Conditional – Returned if provided or assigned. |
| processor | AN | 10 | Provides Processor used |
| responseCode | AN | 5 | See table below |
| responseMessage | AN | 225 | See Table below |
| merchantID | N | 19 | Echo back |
| terminalID | N | 19 | Echo back |
| slID | N | 12 | Echo back |
| amount | N | 12 | Response to Request amount. If less than requested amount then utilize Partial approval process. |
| currencyCode | N | 3 | Echo back from Request |
| maskedAccount | N | 19 | Card or Checking Account masked as 1 st 6 and last 4 for card and last 4 for check |
| cardExpiration Date | N | 4 | MMYY – Provided for Keyed card only |
| tokenID | N | 16 | Conditional - Card or Checking Account token number provided if requested |
| customerNumber | AN | 17 | Conditional -Customer ID Number ECHO back if provided |
| transactionID | AN | 19 | Unique number assigned to each transaction by processor. |
| avsResponse | A | 225 | Conditional – Provided for avsOnly Action Code or if Card Not Present and Billing Zip provided. See Table Below |
| cvvResponse | A | 225 | Conditional – Provided if cvvOnly Action Code used or if cvv2 field used. Match or Mismatch |
| commercialCard | A | 1 | 'y' indicates card is a Commercial Card. "n" indicates not a Commercial Card |
| emvArpc | | | Future Use |
| timestamp | N | 19 | Yyyy-mm-dd-hh-mm-ss |
| merchantField1 | AN | 200 | Optional - Returned in Response |
| merchantField2 | AN | 200 | Optional – Returned in Response |

3.4.3 Refund Response

| Name | Character | Max | Comments |
|-------------------|-----------|-----|---|
| result | A | 1 | A = Approved. D=Decline. R=Retry. C=Call for help |
| processor | AN | 10 | Optional - Used for multiple processors |
| responseCode | AN | 5 | See table |
| responseMessage | AN | 225 | See Table |
| merchantID | N | 19 | Echo back |
| terminalID | N | 19 | Echo back |
| origTransactionID | AN | 19 | Unique number assigned to each transaction by processor. |
| transactionID | AN | 19 | Unique number assigned to each transaction by processor. Assigned for Refund and linked to the Original Payment |
| timestamp | N | 19 | Yyyy-mm-dd-hh-mm-ss |

3.4.4 Token Response

| Name | Character | Max | Comments |
|--------------------|-----------|-----|---|
| result | A | 1 | A = Approved. D=Decline. R=Retry. C=Call for help |
| responseCode | AN | 5 | See table |
| responseMessage | AN | 225 | See Table |
| merchantID | N | 19 | Echo back |
| tokenID | N | 16 | Card or Checking Account token |
| cardExpirationDate | N | 4 | Conditional – Echo Back if Present - MMY – Provided for Keyed card only |

3.4.5 Recurring Response

| NAME | ENUMERATION | COMMENTS |
|---------------------|-------------|---|
| result | A/1 | A = Approved. D=Decline. R=Retry. C=Call for help |
| merchantID | N/19 | Echo back |
| terminalID | N/19 | Echo back |
| currencyCode | N/3 | Echo back from Request |
| maskedAccountNumber | N/19 | Card or Checking Account masked as 1 st 6 and last 4 for card and last 4 for check |
| cardExpirationDate | N/4 | MMYY – Provided for Keyed card only |
| tokenID | N/16 | Returned if requested |
| customerNumber | ANC/25 | Echo Back - Customer ID Number |
| contractNumber | N/20 | Unique Number - Assigned to a Customer's contract by EdgePay |
| timestamp | N/Calendar | Yyyy-mm-dd-hh-mm-ss |

4 RESPONSE CODES – WORKING WITH THE RESPONSE

Within each API Responses there are 4 fundamental elements you can use to drive your software flow. They are:

- Result – A Classification that groups all Response Codes into 4 basic types and allows for you to drive software flow logic. This is the most relevant and best to use.
- Response Code – A unique code for each type of response
- Response Message – A unique message, by Response Code, that can be displayed to the Customer or printed on a Receipt. Very helpful in Decline Messages.
- HTTP Status Code – The RESTful code that aligns with the HTTP Status

When processing a Response message 90% percent of the time you will receive the 'A0000' Success Response Code indicating the transaction request was approved. This would indicate to you to move to the Approved Work Flow and notify the Merchant and Customer as such. Within the 'A0000' Response Code. The accompanying Result Code is 'A'

There are many reasons for Declined transactions. These are denoted with a Result Code of 'D' and indicate a Work Flow to support possible Decline. For these cases you will typically present the Decline Message back to the Card Holder with an opportunity to either fix the error or select another form of payment.

When we return a Result Code of 'C', this indicates a setup problem between you and us and we recommend you Decline the transaction from a Work Flow situation and have someone call us, so we can debug the issue and get you going again!

Occasionally you may receive a Result Code of 'R'. This is an indication of a System error on our side such as a "Timeout" or other condition that may have prohibited us from processing your Request. Under these circumstances we recommend you Retry sending the transaction with the same externalReferenceID. If it was processed successfully we will respond with a 'Duplicate Transaction' which is an indication the previously transaction was processed successfully, and the issue was between you and us and the upstream systems, (i.e. Visa, MC, Amex, Discovery) did successfully process the transaction. If the transaction is Approved, no further action is necessary on your side. If the transaction returns 'Retry' again consider moving to a Decline Work Flow and Call us for help.

Finally, if you receive a Response Code not within the table, please use your Decline Transaction flow logic. This is the most conservative pattern and will serve you best.

RESPONSE CODE TABLE:

| Response Code | Result Code | Response Message | HTTP Status Code |
|---------------|-------------|--|------------------|
| A0000 | A | Success | OK |
| A0002 | A | Partial Amount Approved | OK |
| A0014 | A | Void successful | OK |
| D0001 | A | Duplicate Request (Approved previously) | OK |
| EP0081 | A | Successful | OK |
| EP0340 | A | Presentment event successfully created with Transaction id %s | OK |
| EP0409 | A | Notice request processed successfully | OK |
| EP0444 | A | Billing Plan saved successfully | OK |
| EP0445 | A | Success | OK |
| EP0450 | A | Success | OK |
| E0202 | C | System Error (CardTran) | BAD_REQUEST |
| E0302 | C | System Error (CardUTran) | BAD_REQUEST |
| E0911 | C | System Error | BAD_REQUEST |
| E0912 | C | Error on Host | BAD_REQUEST |
| D0050 | C | Inactive terminal (Backend) | BAD_REQUEST |
| D0060 | C | Inactive account (Backend) | BAD_REQUEST |
| D0070 | C | Unique ID Error (Backend) | BAD_REQUEST |
| D0092 | C | Request Format Error (Backend) | BAD_REQUEST |
| D9012 | C | Invalid Merchant_Info/Agent_Chain_Number must be 6 bytes | BAD_REQUEST |
| D9015 | C | Head Quarter merchant not found. Head Quarter merchant not found. | BAD_REQUEST |
| E0010 | C | Inactive Device (Terminal) | BAD_REQUEST |
| E0020 | C | Inactive Merchant (Account) | BAD_REQUEST |
| E0021 | C | Merchant (Account) configuration missing | BAD_REQUEST |
| E7015 | C | Invalid Transaction Key | BAD_REQUEST |
| E9294 | C | terminalData {terminalCapability, terminalOperatingEnvironment, cardholderAuthenticationMethod, terminalAuthenticationCapability, terminalOutputCapability,maxPinLength} group is required | BAD_REQUEST |
| EP0011 | C | Inactive/Invalid Terminal | BAD_REQUEST |
| EP0012 | C | Authentication failed, please use valid credentials. | BAD_REQUEST |
| EP0051 | C | Invalid Merchant Id | BAD_REQUEST |
| EP0302 | C | System Error. Please contact System Admin. | BAD_REQUEST |
| EP0305 | C | System Error. Please contact System Admin. | BAD_REQUEST |
| EP0306 | C | System Error. Please contact System Admin. | BAD_REQUEST |
| EP0321 | C | System Error. Please contact System Admin. | BAD_REQUEST |
| EP0322 | C | System Error. Please contact System Admin. | BAD_REQUEST |

| | | | |
|--------|---|--|-------------|
| EP0323 | C | System Error. Please contact System Admin. | BAD_REQUEST |
| EP0324 | C | System Error. Please contact System Admin. | BAD_REQUEST |
| EP0325 | C | System Error. Please contact System Admin. | BAD_REQUEST |
| EP0402 | C | System error. Please contact administrator | BAD_REQUEST |
| EP0403 | C | System Error. Please contact administrator | BAD_REQUEST |
| EP0407 | C | System error. Please contact administrator | BAD_REQUEST |
| EP0415 | C | System error. Please contact administrator | BAD_REQUEST |
| EP0431 | C | System Error. Please contact administrator | BAD_REQUEST |
| EP0432 | C | System Error. Please contact administrator | BAD_REQUEST |
| EP0443 | C | System error. Please contact administrator | BAD_REQUEST |
| EP0501 | C | Invalid Merchant Access Key | BAD_REQUEST |
| EP0528 | C | Invalid Merchant ID | BAD_REQUEST |
| EP0900 | C | Message format not supported | BAD_REQUEST |
| F9901 | C | Format Error field details | BAD_REQUEST |
| F9908 | C | Sum of elements of group additionalCharges should not be greater than transaction Amount. | BAD_REQUEST |
| F9911 | C | Last chip Read is not Allowed with Fallback Swipe (Empty Candidate List) transaction | BAD_REQUEST |
| F9915 | C | Void Reason is Mandatory for Chip Card transaction | BAD_REQUEST |
| F9917 | C | Invalid emvTags, {9F1F or 9F20 or 57 or 5A} | BAD_REQUEST |
| F9924 | C | Only 3 healthCare amounts are allowed | BAD_REQUEST |
| E4012 | D | Duplicate transaction (Same amount/ Account). | BAD_REQUEST |
| EP0101 | D | The value of element '%s' is missing. | BAD_REQUEST |
| EP0102 | D | The value '%s' of element '%s' is not valid. | BAD_REQUEST |
| D0003 | D | Duplicate Request (Declined previously) | BAD_REQUEST |
| D0004 | D | Reversal Not Allowed | BAD_REQUEST |
| D0005 | D | Return Not Allowed | BAD_REQUEST |
| D0007 | D | Modify Transaction Not Allowed | BAD_REQUEST |
| D0008 | D | Possible Duplicate Request | BAD_REQUEST |
| D0009 | D | Duplicate Request (Reversed previously) | BAD_REQUEST |
| D0080 | D | Duplicate Request (Backend) | BAD_REQUEST |
| D0090 | D | Reversal Not Allowed (Backend) | BAD_REQUEST |
| D0091 | D | Return Not Allowed (Backend) | BAD_REQUEST |
| D0094 | D | Return not allowed, Card number requested does not match with original transaction card number | BAD_REQUEST |
| D0096 | D | Currency code mismatch with original transaction | BAD_REQUEST |

| | | | |
|-------|---|--|-------------|
| D0097 | D | Multiple amount format in single request not supported | BAD_REQUEST |
| D0098 | D | Multiple tax with same tax type is not allowed. | BAD_REQUEST |
| D007 | D | One or more Fields missing or incorrect. | BAD_REQUEST |
| D201 | D | Unable to determine merchant ID | BAD_REQUEST |
| D202 | D | Unable to process your request. | BAD_REQUEST |
| D215 | D | Invalid amount | BAD_REQUEST |
| D001 | D | Refer to Issuer | BAD_REQUEST |
| D002 | D | Suspected Card (pick-up, hot-card) | BAD_REQUEST |
| D003 | D | Honor with identification? | BAD_REQUEST |
| D004 | D | Invalid Amount | BAD_REQUEST |
| D005 | D | Invalid Card | BAD_REQUEST |
| D006 | D | No such issuer | BAD_REQUEST |
| D007 | D | Invalid fee | BAD_REQUEST |
| D008 | D | Incorrect Pin | BAD_REQUEST |
| D009 | D | Pin attempts exceeded | BAD_REQUEST |
| D010 | D | Key synchronization failed from the host | BAD_REQUEST |
| D011 | D | Expired Card | BAD_REQUEST |
| D012 | D | Insufficient Funds | BAD_REQUEST |
| D013 | D | Invalid From Account | BAD_REQUEST |
| D014 | D | Invalid To Account | BAD_REQUEST |
| D015 | D | Withdrawal Limit exceeded | BAD_REQUEST |
| D016 | D | Withdrawal frequency exceeded | BAD_REQUEST |
| D017 | D | Time limit for Pre-Auth reached | BAD_REQUEST |
| D018 | D | AVS FAILED | BAD_REQUEST |
| D019 | D | Billing ZIP Mismatch | BAD_REQUEST |
| D020 | D | CVV2 verification failed | BAD_REQUEST |
| D021 | D | Issuer or Switch inoperative | BAD_REQUEST |
| D022 | D | Duplicate transaction (Same amount / Account) | BAD_REQUEST |
| D023 | D | Balance unavailable for inquiry | BAD_REQUEST |
| D024 | D | Check Digit Err. | BAD_REQUEST |
| D025 | D | Excluded Bin ID for Merchant | BAD_REQUEST |
| D026 | D | Do not honor | BAD_REQUEST |
| D027 | D | AVS and CVV2 failed | BAD_REQUEST |
| D028 | D | Invalid Date | BAD_REQUEST |
| D029 | D | Invalid Service | BAD_REQUEST |
| D030 | D | Host Validation Error | BAD_REQUEST |
| D031 | D | Activity Limit exceeded | BAD_REQUEST |
| D032 | D | Cannot complete because of Violation | BAD_REQUEST |
| D033 | D | Debit Pin Required | BAD_REQUEST |
| D034 | D | Debit Pin Required | BAD_REQUEST |
| D041 | D | Card information is not provided | BAD_REQUEST |
| D216 | D | ReceiverLimitExceeded | BAD_REQUEST |
| D800 | D | Invalid FCS ID | BAD_REQUEST |
| D801 | D | Invalid Voucher Serial Number | BAD_REQUEST |
| D802 | D | Invalid Voucher Approval Code | BAD_REQUEST |
| D803 | D | Electronics Benefit Transactions cannot contain Fee or Tax | BAD_REQUEST |
| D998 | D | PreFraudScout Decline | BAD_REQUEST |
| D999 | D | General Card Auth Decline | BAD_REQUEST |
| D1001 | D | Account Number Invalid | BAD_REQUEST |
| D1002 | D | Valid Account, Cash payments only. | BAD_REQUEST |

| | | | |
|-------|---|---|-------------|
| D1003 | D | Amount invalid. | BAD_REQUEST |
| D1201 | D | Unable to determine merchant ID | BAD_REQUEST |
| D1202 | D | Unable to process your request. | BAD_REQUEST |
| D1214 | D | Invalid store ID | BAD_REQUEST |
| D1215 | D | Invalid amount | BAD_REQUEST |
| D2001 | D | Refer to Issuer | BAD_REQUEST |
| D2002 | D | Suspected Card (pick-up, hot-card) | BAD_REQUEST |
| D2003 | D | Honor with identification? | BAD_REQUEST |
| D2004 | D | Invalid Amount | BAD_REQUEST |
| D2005 | D | Invalid Card | BAD_REQUEST |
| D2006 | D | No such issuer | BAD_REQUEST |
| D2007 | D | Invalid fee | BAD_REQUEST |
| D2008 | D | Incorrect Pin | BAD_REQUEST |
| D2009 | D | Pin attempts exceeded | BAD_REQUEST |
| D2010 | D | Key synchronization failed from the host | BAD_REQUEST |
| D2011 | D | Expired Card | BAD_REQUEST |
| D2012 | D | Insufficient Funds | BAD_REQUEST |
| D2013 | D | Invalid From Account | BAD_REQUEST |
| D2014 | D | Invalid To Account | BAD_REQUEST |
| D2015 | D | Withdrawal Limit exceeded | BAD_REQUEST |
| D2016 | D | Withdrawal frequency exceeded | BAD_REQUEST |
| D2017 | D | Time limit for Pre-Auth reached | BAD_REQUEST |
| D2018 | D | AVS FAILED | BAD_REQUEST |
| D2019 | D | Billing ZIP Mismatch | BAD_REQUEST |
| D2020 | D | CVV2 verification failed | BAD_REQUEST |
| D2021 | D | Issuer or Switch inoperative | BAD_REQUEST |
| D2022 | D | Duplicate transaction (Same amount / Account) | BAD_REQUEST |
| D2023 | D | Balance unavailable for inquiry | BAD_REQUEST |
| D2024 | D | Check Digit Err | BAD_REQUEST |
| D2025 | D | Excluded Bin ID for Merchant | BAD_REQUEST |
| D2026 | D | Do not honor | BAD_REQUEST |
| D2027 | D | AVS and CVV2 failed | BAD_REQUEST |
| D2028 | D | Invalid Date | BAD_REQUEST |
| D2029 | D | Invalid Service | BAD_REQUEST |
| D2030 | D | Host Validation Error | BAD_REQUEST |
| D2031 | D | Activity Limit exceeded | BAD_REQUEST |
| D2032 | D | Cannot complete because of Violation | BAD_REQUEST |
| D2033 | D | Debit Pin Required | BAD_REQUEST |
| D2034 | D | Debit Pin Required | BAD_REQUEST |
| D2035 | D | Check Service authentication failure | BAD_REQUEST |
| D2039 | D | Could Not Retrieve a Valid Card Number for Token | BAD_REQUEST |
| D2041 | D | Card information is not provided | BAD_REQUEST |
| E2042 | D | No Card found for the BIN No Card found for the BIN | BAD_REQUEST |
| D2200 | D | UNKNOWN_ERROR | BAD_REQUEST |
| D2201 | D | CONTENT_TYPE_NOT_SET | BAD_REQUEST |
| D2202 | D | UNKNOWN_CONTENT_TYPE | BAD_REQUEST |
| D2203 | D | CONTENT_LENGTH_NOT_SET | BAD_REQUEST |
| D2204 | D | INCOMING_REQUEST_READ_ERROR | BAD_REQUEST |
| D2205 | D | OUTGOING_RESPONSE_SEND_ERROR | BAD_REQUEST |
| D2206 | D | INPUT_VALIDATION_ERROR | BAD_REQUEST |

| | | | |
|--------|---|--|-------------|
| D2208 | D | OCT_FAILED | BAD_REQUEST |
| D2209 | D | AFT_FAILED | BAD_REQUEST |
| D2210 | D | AFTR_FAILED | BAD_REQUEST |
| D2211 | D | REMOTE_VPP_ERROR | BAD_REQUEST |
| D2212 | D | INVALID_ISSUER_COUNTRY_CODE | BAD_REQUEST |
| D2213 | D | FAST_FUNDS_NOT_ENABLED | BAD_REQUEST |
| D2214 | D | INTERNAL_ERROR | BAD_REQUEST |
| D2215 | D | ACNL_FAILED | BAD_REQUEST |
| D2216 | D | ReceiverLimitExceeded | BAD_REQUEST |
| D2800 | D | Invalid FCS ID | BAD_REQUEST |
| D2801 | D | Invalid Voucher Serial Number | BAD_REQUEST |
| D2802 | D | Invalid Voucher Approval Code | BAD_REQUEST |
| D2803 | D | Electronics Benefit Transactions cannot contain Fee or Tax | BAD_REQUEST |
| EP0012 | D | Invalid Token | BAD_REQUEST |
| EP0013 | D | Merchant is not activated for processing VISA card transactions, please contact system administrator. | BAD_REQUEST |
| EP0014 | D | Merchant is not activated for processing MASTERCARD card transactions, please contact system administrator. | BAD_REQUEST |
| EP0015 | D | Merchant is not activated for processing AMEX card transactions, please contact system administrator. | BAD_REQUEST |
| EP0016 | D | Merchant is not activated for processing DISCOVER card transactions, please contact system administrator. | BAD_REQUEST |
| EP0017 | D | Merchant is not activated for processing JCB card transactions, please contact system administrator. | BAD_REQUEST |
| EP0018 | D | Merchant is not activated for processing DINERS/CHINA UNIONPAY card transactions, please contact system administrator. | BAD_REQUEST |
| EP0019 | D | Invalid Track Data | BAD_REQUEST |
| EP0053 | D | Card Number cannot be blank | BAD_REQUEST |
| EP0054 | D | Card number is invalid | BAD_REQUEST |
| EP0055 | D | Invalid Expiry Date (MMYY) | BAD_REQUEST |
| EP0056 | D | Card expiry date cannot be blank | BAD_REQUEST |
| EP0057 | D | Not supported card type | BAD_REQUEST |
| EP0058 | D | Duplicate Token Request | BAD_REQUEST |
| EP0064 | D | Suspend and card-expiry update are simultaneously not allowed | BAD_REQUEST |
| EP0065 | D | Invalid Account Number | BAD_REQUEST |
| EP0067 | D | Account Number cannot be blank | BAD_REQUEST |
| EP0068 | D | Merchant is not configured for E-Check Tokens | BAD_REQUEST |
| EP0070 | D | Token status is not valid | BAD_REQUEST |
| EP0071 | D | Token is already in active status | BAD_REQUEST |
| EP0072 | D | Token is already suspended | BAD_REQUEST |
| EP0073 | D | Token not associated with the merchant | BAD_REQUEST |
| EP0074 | D | Invalid Usage type, value can be SINGLE/MULTIPLE | BAD_REQUEST |

| | | | |
|--------|---|---|-------------|
| EP0075 | D | Invalid Purpose type, value can be INTERNAL/EXTERNAL | BAD_REQUEST |
| EP0076 | D | Invalid Token Type | BAD_REQUEST |
| EP0077 | D | Token Suspended Permanently, alternative token exists | BAD_REQUEST |
| EP0078 | D | No Token Exists for this MID | BAD_REQUEST |
| EP0079 | D | Token Suspended Permanently | BAD_REQUEST |
| EP0080 | D | Routing Number cannot be blank | BAD_REQUEST |
| EP0101 | D | The value of element '%s' is missing. | BAD_REQUEST |
| EP0102 | D | Not supported card type | BAD_REQUEST |
| EP0102 | D | The value '%s' of element '%s' is not valid. | BAD_REQUEST |
| EP0103 | D | The field '%s' is invalid. | BAD_REQUEST |
| EP0103 | D | The field '%s' is invalid. | BAD_REQUEST |
| EP0104 | D | The value of element '%s' is not valid. | BAD_REQUEST |
| EP0105 | D | Merchant is not allowed for processing '%s' card transactions, please contact system administrator. | BAD_REQUEST |
| EP0106 | D | Level2/Level3 and HealthCare cannot be processed for a single payment request | BAD_REQUEST |
| EP0106 | D | Level2/Level3 and HealthCare cannot be processed for a single payment request | BAD_REQUEST |
| EP0107 | D | Level2/Level3 Auth request should have Level2/Level3 Auth elements | BAD_REQUEST |
| EP0108 | D | Level2/Level3 Auth request should not have Level2/Level3 Capture elements | BAD_REQUEST |
| EP0109 | D | HealthCare transactions are only supported by VISA,MasterCard and Discover | BAD_REQUEST |
| EP0110 | D | Invalid input fields in request | BAD_REQUEST |
| EP0111 | D | Summation of all the sub element amount should be equivalent to total amount | BAD_REQUEST |
| EP0112 | D | Duplicate Transaction | BAD_REQUEST |
| EP0115 | D | The value of element 'amount' is not required. | BAD_REQUEST |
| EP0116 | D | Refund transaction not possible for this transaction. | BAD_REQUEST |
| EP0117 | D | The value of element 'cardDataSource' is not required | BAD_REQUEST |
| EP0118 | D | Invalid Transaction Logging Request | BAD_REQUEST |
| EP0119 | D | Requested operation cannot be performed on this transaction. | BAD_REQUEST |
| EP0120 | D | Level2/Level3 Capture request should not have Level2 or Level3 Auth element | BAD_REQUEST |
| EP0121 | D | Level2/Level3 Capture request should have Level2 or Level3 Capture element | BAD_REQUEST |
| EP0122 | D | Requested operation cannot be performed on this transaction. | BAD_REQUEST |
| EP0123 | D | Partial Refund not allowed for Authorized transactions | BAD_REQUEST |
| EP0124 | D | Token generation is not allowed for requested operation. | BAD_REQUEST |
| EP0301 | D | ACH Transaction already Voided | BAD_REQUEST |
| EP0302 | D | Invalid Routing Number | BAD_REQUEST |
| EP0303 | D | Invalid %s with value %s | BAD_REQUEST |

| | | | |
|--------|---|---|-------------|
| EP0304 | D | Missing %s | BAD_REQUEST |
| EP0310 | D | Invalid Token | BAD_REQUEST |
| EP0316 | D | Either Customer Name or Receiver Name is required | BAD_REQUEST |
| EP0317 | D | Ach Transaction already Refunded | BAD_REQUEST |
| EP0320 | D | Duplicate Transaction | BAD_REQUEST |
| EP0330 | D | ARC transaction is not supported for amount more than 25,000 | BAD_REQUEST |
| EP0344 | D | Requested refund amount is greater than original amount | BAD_REQUEST |
| EP0345 | D | Token not associated with the merchant | BAD_REQUEST |
| EP0346 | D | ACH Service not enabled for merchant | BAD_REQUEST |
| EP0347 | D | Merchant is not allowed to use API | BAD_REQUEST |
| EP0403 | D | No customer found | BAD_REQUEST |
| EP0408 | D | Merchant is not configured for %s Type payment | BAD_REQUEST |
| EP0414 | D | Invalid Token. %s | BAD_REQUEST |
| EP0414 | D | Customer already exists | BAD_REQUEST |
| EP0418 | D | Contract cannot be associated with Suspended Billing Plan | BAD_REQUEST |
| EP0419 | D | Merchant is not configured for E-Check payment | BAD_REQUEST |
| EP0421 | D | Invalid %s with value %s. Only Y or N is accepted | BAD_REQUEST |
| EP0422 | D | At least one contract should be present | BAD_REQUEST |
| EP0422 | D | Invalid start date. Contract Start Date %s cannot be past date. | BAD_REQUEST |
| EP0423 | D | This Merchant does not have selected Billing Plan | BAD_REQUEST |
| EP0423 | D | Invalid end date. Contract End Date %s should be after Contract Start Date %s. | BAD_REQUEST |
| EP0424 | D | Cannot create new contract with suspended status | BAD_REQUEST |
| EP0428 | D | No contract found | BAD_REQUEST |
| EP0429 | D | No results found | BAD_REQUEST |
| EP0446 | D | No record found | BAD_REQUEST |
| EP0447 | D | Duplicate record '%s' | BAD_REQUEST |
| EP0512 | D | Error generating ACH Token for Merchant | BAD_REQUEST |
| EP0765 | D | Level2/Level3 is not supported by '%s' card. | BAD_REQUEST |
| EP901 | D | Expected Fraud Transaction | BAD_REQUEST |
| EP906 | D | Invalid Card Number | BAD_REQUEST |
| F9909 | D | cashTendered must not be less than transactionAmount | BAD_REQUEST |
| F9910 | D | lastChipRead is Mandatory with Fallback Swipe (lcc Terminal Error) transaction | BAD_REQUEST |
| F9912 | D | Invalid content, one of {track1Data, track2Data, track3Data} is required | BAD_REQUEST |
| F9913 | D | Invalid content, {encodedCardData, keyedCardData or swipedCardData} is not Allowed with Chip Card | BAD_REQUEST |
| F9914 | D | emvFallbackCondition is Mandatory with Fallback Swipe transaction | BAD_REQUEST |

| | | | |
|--------|---|---|-----------------------|
| F9916 | D | Fallback Swipe allowed with track2Data only | BAD_REQUEST |
| F9918 | D | Invalid content, {track1Data, track3Data, emulatedTrackData} is not Allowed with Chip Card and encryptionType | BAD_REQUEST |
| R0001 | D | Service Not Enabled | BAD_REQUEST |
| R0003 | D | Invalid Customer Data | BAD_REQUEST |
| R0009 | D | Chargeback Shutoff | BAD_REQUEST |
| T0001 | D | Service Not Enabled | BAD_REQUEST |
| EP0082 | R | Token Service Error Occurred | BAD_REQUEST |
| EP0083 | R | Error in Token Generation | BAD_REQUEST |
| A2207 | R | OUTGOING_REQUEST_FAIL | INTERNAL_SERVER_ERROR |
| D5213 | R | Service is temporarily unavailable. Please try later | INTERNAL_SERVER_ERROR |
| E0015 | R | Unable to process your request. Settlement InProgress. | BAD_REQUEST |
| E0910 | R | Time out | TIMEOUT |
| E5213 | R | Service is temporarily unavailable. Please try later | INTERNAL_SERVER_ERROR |
| E6072 | R | Transaction is in process. Please try again after some time. | INTERNAL_SERVER_ERROR |
| E8900 | R | System Error | INTERNAL_SERVER_ERROR |
| EP0113 | R | Multipass Service Failed. Please retry or contact system administrator. | BAD_REQUEST |
| EP0114 | R | Internal error occurred. Please retry or contact system administrator. | BAD_REQUEST |
| EP0125 | R | Transaction has been declined, please try again | BAD_REQUEST |
| EP0126 | R | System Error!! | BAD_REQUEST |
| EP0326 | R | System Error. Please try again | INTERNAL_SERVER_ERROR |
| EP0327 | R | System Error. Please try again | INTERNAL_SERVER_ERROR |
| EP0328 | R | System Error. Please try again | INTERNAL_SERVER_ERROR |
| EP0329 | R | System Error. Please try again | INTERNAL_SERVER_ERROR |
| EP0331 | R | %s with value %s not supported | INTERNAL_SERVER_ERROR |
| T0003 | R | Service Unsucessful | INTERNAL_SERVER_ERROR |

AVS Reponses

AVS, or Address Verification Service, compares the Card Holders Billing address information to the information on file for the Card Holders at their Issuing bank. The service presents the results based upon the level of information provided. Typically, merchants use just the Billing Zip code to compare due to the variation of street address.

EdgePay has 3 options for supporting AVS. They are:

1. Decline on AVS Billing Zip Code mismatch – If configured for the merchant, if the Zip Code does not match the Zip code on file, EdgePay will decline the payment request and reverse the authorization request back to the card issuer
2. No Action on AVS information – No action will be taken with respect to AVS results and it is up to your system to decide to accept the approved payment or decline the payment. If you decide to decline the payment we recommend you refund, (reverse) the approved payment request authorization.
3. FMS Review on AVS Mismatch – You can setup in our Fraud Management Service to review all AVS mismatches. The Payment request is Approved and will process but, the payment will be displayed in the FMS review screen and you can take action there to accept or refund the transaction.

Here is a table of all possible AVS Responses.

| Code | Authorization Response Message | Response Definition |
|------|--------------------------------|---|
| 0 | Approved | Address verification was not requested. |
| A | Address Match | The street address submitted matches what is on file. |
| B | Address Match | The street address submitted matches what is on file. The ZIP code does not match what was on file. |
| C | Service Unavailable | This service is not supported. |
| D | Exact Match | The street address submitted matches what is on file. |
| F | Exact Match | The street address and ZIP code submitted match what is on file. This applies to UK transactions. |
| G | Version Unavailable | The issuer does not participate in this service. This applies to non-US issuers. |
| I | Version Unavailable | The AVS information was not verified. |
| M | Exact Match | The street address submitted matches what is on file. |
| N | No Match | Neither the street address or ZIP code match what is on file. |

| | | |
|---|--|--|
| P | Zip Match | The ZIP code submitted matches what is on file. The street address does not match what is on file. |
| R | Retry | The issuer system is currently unavailable. |
| S | Serv Unavailable | This service is not supported. |
| U | Ver Unavailable | AVS is unavailable. |
| W | Zip Match | The nine character ZIP code submitted matches what is on file. |
| X | Exact Match | The nine character ZIP code submitted matches what is on file. |
| Y | Exact Match | The five character ZIP code submitted matches what is on file. |
| Z | Zip Match | The nine character ZIP code submitted matches what is on file. |
| 1 | Cardholder name and ZIP match | AMEX only; The cardholder name and ZIP code submitted matches what is on file. |
| 2 | Cardholder name, address, and ZIP match | AMEX only; The cardholder name, address, and ZIP code submitted match what is on file. |
| 3 | Cardholder name and address match | AMEX only; The cardholder name and address submitted match what is on file. |
| 4 | Cardholder name match | AMEX only; The cardholder name submitted match what is on file. |
| 5 | Cardholder name incorrect, ZIP match | AMEX only; The cardholder name does not match what is on file. The ZIP code submitted matches what is on file. |
| 6 | Cardholder name incorrect, address and ZIP match | AMEX only; The cardholder name does not match what is on file. The address and ZIP code submitted matches what is on file. |
| 7 | Cardholder name incorrect, address match | AMEX only; The cardholder name does not match what is on file. The address submitted matches what is on file. |
| 8 | Cardholder, all do not match | AMEX only; Neither the cardholder name, address, or ZIP code match what is on file. |

CVV Response

If the CVV, (Security Code) does not match the Security Code on file for the Card issued, EdgePay will decline the transaction.

Please consider using a Decline flow logic.

| Code | Response Description |
|------|---------------------------------|
| M | CVV2 verification successful |
| P | CVV2 verification not performed |
| U | CVV2 verification not available |
| N | CVV2 verification fail/mismatch |
| S | CVV2 code not present on card |

5 EDGEPAY PIVOT

5.1 DEVELOPMENT AND PRODUCTION URL'S

Development

| Function | URL |
|--------------------|---|
| One Time Pivot Key | https://api.edgepay-uat.com/generatePivotAuthKey |
| Pivot JS script | https://static.portal.edgepay-uat.com/public/js/edgepayPivot.min.js |
| Refund | https://api.edgepay-uat.com/refund |
| Token | https://api.edgepay-uat.com/token |
| Recurring | https://api.edgepay-uat.com/recurring/customer/add |

Production

| Function | URL |
|--------------------|---|
| One Time Pivot key | https://edgepayapi.com/generatePivotAuthKey |
| Pivot JS script | https://static.edgepayportal.com/public/js/edgepayPivot.min.js |
| Refund | https://edgepayapi.com/refund |
| Token | https://edgepayapi.com/token |
| Recurring | https://edgepayapi.com/recurring/customer/add |

5.2 THE IMPLEMENTATION

EdgePay Pivot is a JavaScript Library and Function that you can add to your Payment page and bypass sending sensitive card data to your server, thus eliminating the risks associated with storing, processing, or transmitting credit card data and allows you to operate in a PCI Compliance way.

You would use this Library and Function if do not want to store, process, or transmit card data on your system. If your design already includes server-side card processing, this library and function may not be for you.

For Production processing you will use the Merchant Key, Merchant ID, and Terminal ID we setup for you. For testing purposes please feel free to use the information provided below in the code snippets.

EdgePay Pivot is a 2-step process:

Step 1: At the point of “Checkout” your Server-Side code will execute the following code to retrieve a 1-time Pivot Authentication Key from our Server. This 1-time use key allows you to use the EdgePay Pivot to get a 1-time Token to replace the Card data.

The 1-time Pivot Authentication Key is valid for only 15 minutes, then expires. The key is also not valid after it has been submitted and used. Here is the code you would use to request the 1-Time Pivot Authentication Key from your server.

Java:

```
1. OkHttpClient client = new OkHttpClient();
2.
3. MediaType mediaType = MediaType.parse("application/json");
4.
5. RequestBody body = RequestBody.create(mediaType, "{ \r\n  \"merchantID\": \"6320340123456901\", \r\n  \"terminalID\": \"88800000282601\" \r\n} \r\n ");
6.
7. Request request = new Request.Builder().url("https://api.edgepay-
uat.com/generatePivotAuthKey").post(body).addHeader("Content-
Type", "application/json").addHeader("merchantKey", "C8E05D70D3C8D3D675EAA2FE86674FE
A3BCE1AFBECA4322C
").addHeader("externalReferenceID", "1234561528498479").addHeader("transactionDate", "2018
-06-08T00:01:00").addHeader("Cache-Control", "no-cache").build();
8.
9. Response response = client.newCall(request).execute();
```

Python:

```
1. import http.client
2.
3. conn = http.client.HTTPConnection("api.edgepay-uat.com")
4.
5. payload = "{ \r\n  \"merchantID\": \"6320340123456901\", \r\n  \"terminalID\": \"88800000282601\" \r\n} \r\n "
6.
7. headers = {
8.     'Content-Type': "application/json",
9.     'merchantKey': "C8E05D70D3C8D3D675EAA2FE86674FEA3BCE1AFBECA4322C
",
10.    'externalReferenceID': "1234561528498328",
11.    'transactionDate': "2018-06-08T00:01:00",
12.    'Cache-Control': "no-cache"
13. }
```

```

14.
15. conn.request("POST", "generatePivotAuthKey", payload, headers)
16.
17. res = conn.getResponse()
18. data = res.read()
19. print(data.decode("utf-8"))

```

C#

```

1. var client = new RestClient("https://api.edgepay-uat.com/generatePivotAuthKey");
2.
3. var request = new RestRequest(Method.POST);
4.
5. request.AddHeader("Cache-Control", "no-cache");
6. request.AddHeader("transactionDate", "2018-06-08T00:01:00");
7. request.AddHeader("externalReferenceID", "1234561528498677");
   request.AddHeader("merchantKey", "C8E05D70D3C8D3D675EAA2FE86674FEA3BCE1AFBEC
   A4322C
8. ");
9. request.AddHeader("Content-Type", "application/json");
   request.AddParameter("undefined", "{ \r\n  \"merchantID\": \"6320340123456901
10. \", \r\n  \"terminalID\": \"88800000282601\" \r\n} \r\n ", ParameterType.RequestBody);
11.
12. IRestResponse response = client.Execute(request);

```

This request will return the following response to your Server as an Approved Transaction, providing your Server a 1-Time Pivot Key.

Success:

```

1. {
2.   "result": "A",
3.   "responseCode": "EP0020",
4.   "responseMessage": "Pivot Key Generation Successful",
5.   "timestamp": "2018-06-08 22:35:42.486",
6.   "pivotAuthKey": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ7XCJtZXJjaGFudEtleVwiOlwiTnINCWJPT
   01jT1EwZklvZnVnMXpQZE1WQ05HVU5iZmdyMUt2c2FINmE5NFprY0hqdEE3MkRVZWptTGxK
   cnFVejBuWithVUZoU3ZSZE42amdmdkNRYXN3aE9jMm80bGd1YjV5NmQ2eEJQdDg9XCIsXCJt
   ZXJjaGFudEIEEXCI6XCI2MzlwMzQwMTIzNDU2NzlyXCIsXCJ0ZXJtaW5hbEIEEXCI6XCI4ODgwM
   DAwMDI4MjYwMVwiLFwiZXhwaXJlc0luU2Vjb25kc1wiOjkwMCxclmdlbmVyYXRpb25UaW1ISW5
   NaWxsaXNcljoxNTI4NDk3MzQyNDg0fSIsImhhdCI6MTUyODQ5NzZM0Mn0.1Ytwan7EvC4EqvsQ-
   dsS4iiXAQd_zTWqAFigDqnJvR0"
7. }

```

There can also be a negative response in the request for a 1-Time Pivot Key. This can come in the form of 'Decline, Retry, or Call for Help' within the result field.

Error:

```

1. {
2.   "result": "R",
3.   "responseCode": "EP0900",
4.   "responseMessage": "Message format not supported"
5. }

```

```

1. {
2.   "result": "D",
3.   "responseCode": "EP0102",
4.   "responseMessage": "The value '63203401234567221' of element 'merchantID' is not valid."
5. }

```

Here is a table of all responses and appropriate action to take:

| RESULT | RESPONSE CODE | RESPONSE MESSAGE | RECOMMENDED NEXT ACTION |
|--------|---------------|---|--|
| A | EP0020 | Pivot Key Generation Successful | Use the Pivot Key to generate Payment Page (See below) |
| C | EP0900 | Message format not supported | Call GET for support |
| C | EP0102 | The value 'xxx' of element 'merchantID' is not valid | The Merchant ID in the request packet is not valid. Verify your input |
| C | EP0102 | The value 'xxx' of element 'terminalID' is not valid | The Terminal ID in the request packet is not valid. Verify your input. |
| C | EP0012 | Authentication failed, please use valid credentials | The Merchant Key is not valid. Update Merchant key. |
| D | EP0101 | The value of element 'externalReferenceID' is missing | The externalReferenceID must be included with every request, and be unique |
| D | EP0101 | The value of element 'merchantID' is missing | The merchantID must be included with every request, and be unique |
| D | EP0101 | The value of element 'terminalID' is missing | The terminalID must be included with every request, and be unique |
| R | EP0999 | System Error | Internal Server error, please retry request |

Here are the details of the Response data:

| Field | Attribute | Description |
|------------------------|------------|---|
| result | A/1 | A = Approved D = Decline R = Retry C = Call for Help |
| responseCode | AN/6 | The 6-digit code and the table below may be used for determining next step in your code |
| responseMessage | AN String | Text describing the condition |
| timestamp | N/Calendar | Yyyy-mm-dd-hh-mm-ss |
| pivotAuthKey | 256 bits | JSON Web Key |

Once you have your 1-Time Pivot Authentication Key, you are ready to load the Payment Page. As part of that load you include the following code as part of generating the HTML page.

```

1. <script type = "text/javascript"
2. src = "https://static.portal.edgepay-uat.com/public/js/edgepayPivot.min.js"></script>

1. <script type = "text/javascript">
2.
3. EdgePay.init({
4.     pivotAuthKey: "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ7XCJtZXJjaGFudEtleVwiOiwiTGdSWERva21mbGc1Y3F4anVVUG5aRlpLVHphGZ1hFZGt0Vlo0YkNybnllVHdleXJ0YW53UDhOQW5kbmFwNm9xU3o0SnNlaDVNTm5SNWFFWTBmUThCdHEzVmRibEcwL3YvK21JMWJlYm92WjA9XCIsXCJtZXJjaGFudEiEXCI6XCi2MzlwMzQwMTIzNDU2NzlyXCIsXCJ0ZXJtaW5hbEiEXCI6XCi4ODgwMDAwMDI4MjYwMVwiLFwiZXhwaXJlc0luU2Vjb25kc1wiOjkwMCxldmVvYyYXRpb25UaW1ISW5NaWxsaXNcljoxNTI4NDk5NTU3NDQzfSIsImhhdCI6MTUyODQ5OTU1N30.hyP_leBSjd9bLb-cXfAik-Yf9gcDnWjE6Fe29EPpG0c" (This is the key from the 1-time pivot key request)
5. });
6.
7. EdgePay.getToken({
8.     formFieldIdSelector: {
9.         submitButton: "btnSubmit",
10.        cardNumber: "inputCreditCard",
11.        cardExpirationDate: "inputExpirationDate",

```

```

12.     cvv2: "inputCVV2"
13. },
14. onSuccess: function(success) {
15.     /*success call back to applied business logic*/
16.     /*tokenId can be found in success.body.tokenID */
17.     document.getElementById("token").value= success.body.tokenID;
18.     document.getElementById("paymentForm").submit();
19. },
20. onError: function(error) {
21.     /*error call back to applied business logic*/
22.
23. }
24. });
25.
26. </script>

```

If your page have the card expiration date in two different fields (month and year), you can replace the cardExpirationDate field with the cardExpirationMonth and cardExpirationYear fields in the getToken function:

```

27. EdgePay.getToken({
28.     formFieldIdSelector: {
29.         submitButton: "btnSubmit",
30.         cardNumber: "inputCreditCard",
31.         cardExpirationMonth: "inputExpiryMonth",
32.         cardExpirationYear: "inputExpiryYear",
33.         cvv2: "inputCVV2"
34.     },

```

This function will download the EdgePay JavaScript library and add to the user's browser.

Within the code you added are the variables that holds the 1-time Pivot Authentication Key. Please insert the key you retrieved from the previous request in the pivotAuthKey variable when you build the page.

Also, within the code is the definitions required to download the library to process the Card information and return the 1-Time token. The code controls the input variables and submit button as well as handling an Ajax Method that will get fired from executing the Submit button. This method will send the Card Information to EdgePay using the 1-Time Pivot Key for Authentication.

For your Payment form most of the variables are fixed but we recognize there are several approaches to Card Expiration Date. Thus, we have enabled the following formats within the EdgePay Pivot Library:

CARD EXPIRATION DATE FORMATS SUPPORTED:

MMYY

MMYYYY

MM/YY

MM YY (2 Fields)

MM YYYY (2 Fields)

Feel free to create the Card Expiration format that best fits your page and needs.

The Java script library will perform the following edits as part of entry on the form:

| ERROR TYPE | FIELD NAME | ERROR TEXT | RAW |
|--------------------------|---------------------|-----------------------------------|--|
| VALIDATION_FAILED | cardNumber | This is required. | {"errorType":"VALIDATION_FAILED","body":[{"fieldName":"cardNumber","errorText":"This is required"}]} |
| VALIDATION_FAILED | cardNumber | Must be a number | {"errorType":"VALIDATION_FAILED","body":[{"fieldName":"cardNumber","errorText":"Must be a number"}]} |
| VALIDATION_FAILED | cardExpirationField | Card Expiration Field is required | {"errorType":"VALIDATION_FAILED","body":[{"fieldName":"cardExpirationField","errorText":"Card Expiration Field is required"}]} |

| | | | |
|--------------------------|--|--|---|
| VALIDATION_FAILED | cardExpirationDate | Card Expiration format not matched | {"errorType":"VALIDATION_FAILED","body":[{"fieldName":"cardExpirationDate","errorText":"Card Expiration format not matched"}]} |
| VALIDATION_FAILED | cardExpirationMonth/ cardExpirationYear | Card Expiration format not matched | {"errorType":"VALIDATION_FAILED","body":[{"fieldName":"cardExpirationMonth/cardExpirationYear","errorText":"Card Expiration format not matched"}]} |
| VALIDATION_FAILED | cardExpirationField | Please input cardExpirationField field or combination of cardExpirationMonth and cardExpirationYear fields | {"errorType":"VALIDATION_FAILED","body":[{"fieldName":"cardExpirationField","errorText":"Please input cardExpirationField field or combination of cardExpirationMonth and cardExpirationYear fields"}]} |
| VALIDATION_FAILED | cvv2 | Must be a number | {"errorType":"VALIDATION_FAILED","body":[{"fieldName":"cvv2","errorText":"Must be a number"}]} |
| VALIDATION_FAILED | cvv2 | Length must be xxxx or xxx | {"errorType":"VALIDATION_FAILED","body":[{"fieldName":"cvv2","errorText":"Length must be xxxx or xxx"}]} |
| VALIDATION_FAILED | cardNumber | It is not a proper card number | {"errorType":"VALIDATION_FAILED","body":[{"fieldName":"cardNumber","errorText":"It is not a proper card number"}]} |

| | | | |
|--------------------------|---------|---------|--|
| VALIDATION_FAILED | See Raw | See Raw | { "errorType": "VALIDATION_FAILED", "body": [{ "fieldName": "cardNumber", "errorText": "It is not a proper card number", }, { "fieldName": "cvv2", "errorText": "Length must be xxxx or xxx"}]} |
|--------------------------|---------|---------|--|

These edits will be corrected by the User before the code will allow the user to proceed.

Upon the user selecting the Submit button the JavaScript library that is downloaded executes a request to EdgePay and returns a 1-Time token to replace the Card information, (Credit Card Number, Expiration Date, and CVV Value). This token is time sensitive as well and will expire in 15 minutes.

You will use the “OnSuccess” Call back method to generate your POST back to your server with your order information and the 1-Time token which is now ready to make the payment from your server using the EdgePay Payment API and inputting the Token instead of the Credit Card information.

Here is the “OnSuccess” response message:

```
{body:{"result":"A","responseCode":"EP0081","responseMessage":"Successful","merchantID":"6320340123456722","tokenID":"5839914216406751","cardExpirationDate":"0220"},"status":200}
```

After processing the Payment request use the Payment Response Result to determine your next action. If the Payment Response is an “OnError” response and you wish to ask the Customer for another form of payment or re-enter your card information you must first request another 1-Time Authentication Key and load asynchronously to the current Payment page, then display the work flow instructions to the user.

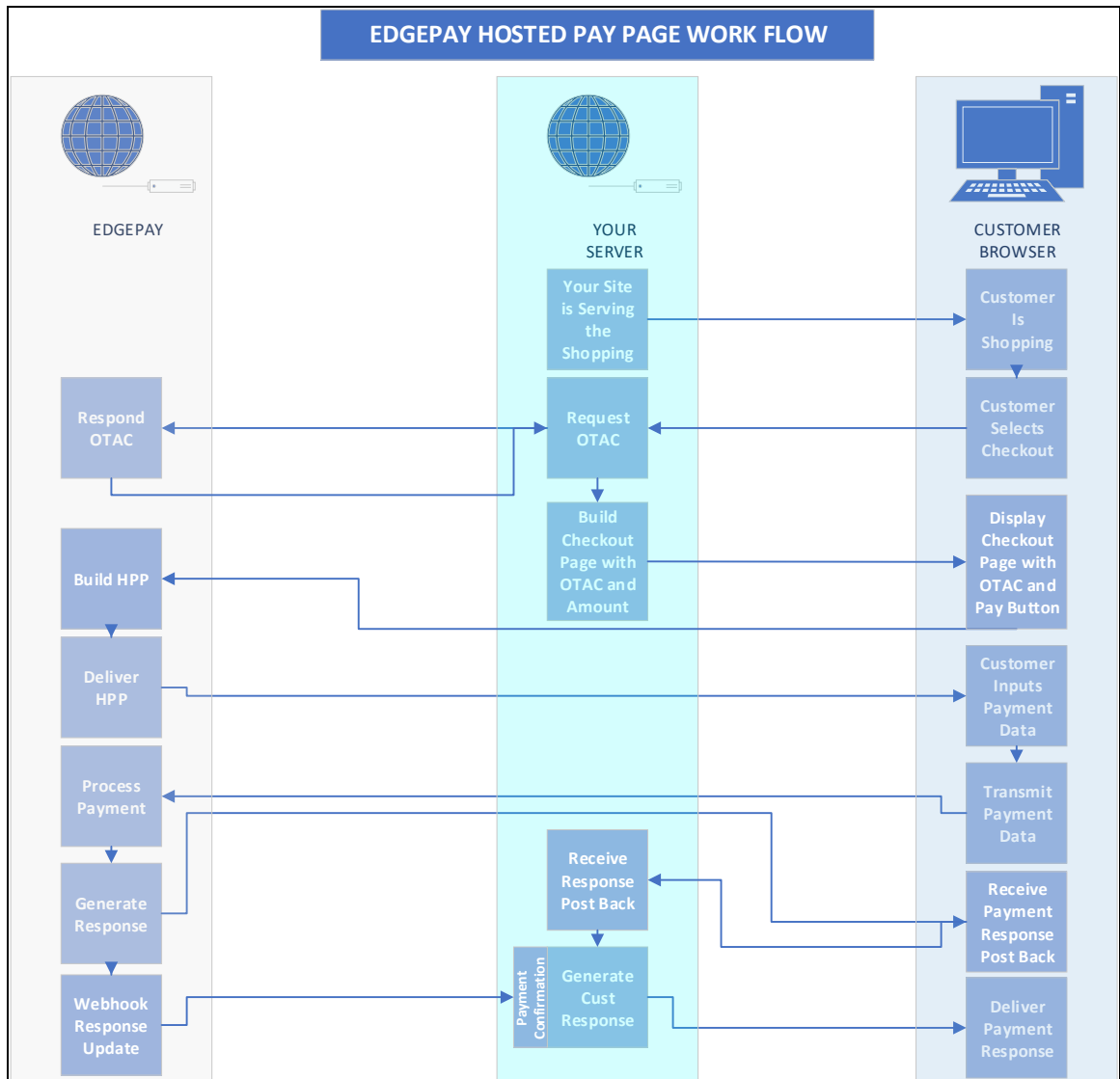
If you receive a “OnError” result use the workflow below to facilitate next steps for your development. These can be performed on form via JavaScript or back at your server.

| Result | RESPONSE CODE | RESPONSE MESSAGE | RAW | RECOMMENDED NEXT ACTION |
|--------|---------------|--|--|--|
| R | EP0999 | System error | { "errorType": "SERVICE_INVOCATION_ERROR", "body": { "result": "R", "responseCode": "EP0999", "responseMessage": "System error", "timestamp": "2018-06-12-14-43-55", "status": 500 } } | Request a new 1-Time Pivot Key and resend request |
| D | EP0055 | Invalid Expiry Date | { "errorType": "SERVICE_INVOCATION_ERROR", "body": { "result": "D", "responseCode": "EP0055", "responseMessage": "Invalid Expiry Date (MMYY)", "merchantID": "6320340123456722", "cardExpirationDate": "1111", "status": 400 } } | Display result to user and request corrected Date. Note; Requires a new Pivot Key |
| D | EP0012 | Authentication failed, please use valid credentials | { "errorType": "SERVICE_INVOCATION_ERROR", "body": { "result": "D", "responseCode": "EP0012", "responseMessage": "Authentication failed, please use valid credentials.", "status": 400 } } | The 1-time Key was invalid. Request another 1-Time key and resubmit |
| C | EP0900 | Message format not supported | { "errorType": "SERVICE_INVOCATION_ERROR", "body": { "result": "C", "responseCode": "EP0900", "responseMessage": "Message format not supported", "status": 400 } } | This is an internal Key error. Call for help! |
| D | EP0054 | Card number is invalid | { "errorType": "SERVICE_INVOCATION_ERROR", "body": { "result": "D", "responseCode": "EP0054", "responseMessage": "Card number is invalid", "merchantID": "6320340123456722", "cardExpirationDate": "0220", "status": 400 } } | The customer input an invalid card. Get a new key and prompt to retry card input |
| D | EP0102 | The value 'XXXX' of element 'cardExpirationDate' is not valid. | { "errorType": "SERVICE_INVOCATION_ERROR", "body": { "result": "D", "responseCode": "EP0102", "responseMessage": "The value '0200' of element 'cardExpirationDate' is not valid.", "status": 400 } } | The customer input an invalid Expiration date. Get a new key and prompt to retry Card Expiration input |

This concludes the EdgePay Pivot work flow. If you have additional questions, please contact us at 888.775.1500.

6 HOSTED PAYMENT PAGE

The EdgePay hosted payment page is a service that uses our server to process the payment. By using this service, you avoid storing or processing Credit Card data on your server and the challenges of PCI audit or certification. Below is a flowchart to help understand the interfaces and process.



Prior to using this process, you must use our Hosted Pay Page function in our “Processing Center” to construct your Hosted Pay Page, “Pay Button” and define your Post Back URL.

The Webhook Confirmation is a future deliverable for Q1 2019.

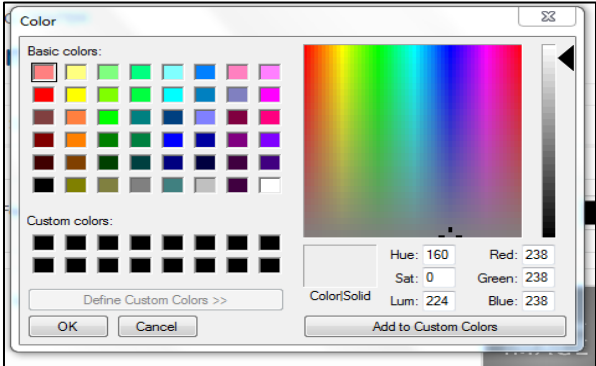
This payment interface requires minimal web development and maintenance because it is managed by EdgePay. It also reduces the merchant liability for PCI compliance because the liability resides on our platform, out of the merchant's environment.

6.1 STEP 1: SETTING UP A HOSTED PAYMENT PAGE

The Hosted Payment Page under Settings menu allows the Merchant to set up their payment page. Once a user successfully login to the EdgePay Application UI, user specific Dashboard page will be displayed to the user with a left navigation menu based on the user privileges provided.

6.1.1 To setup the hosted payment page the merchant will

1. log on to www.edgepayportal.com
2. Under settings select **Hosted Payment Page**
3. Select **Add Details**. The **HPP Information** screen will appear
4. By editing the following fields set up various merchant-specific fields which will result in the look and feel of the merchant company
 - a.

| Field | Description |
|-----------------------|--|
| Style Settings | |
| Font Style | <p>List of Font Styles</p> <ul style="list-style-type: none"> • normal • italic • oblique • initial • inherit |
| Font Color | <p>Users can Select Font Color from a Color Panel.</p>  |

| Field | Description |
|---------------------------------|--|
| Background Color | Users can Select Background Color from a Color Panel. |
| Upload Image | User can upload Image which will be used as Logo. |
| Header/Footer Option | User can select Header or Footer placement for the Logo. |
| Header Text | User can specify Header Text |
| Footer Text | User can specify Footer Text |
| Processing Options | |
| Response URL | The response URL for the HPP. |
| Payment Options | It can be eCheck or Credit or Both. |
| Maximum Attempts | Maximum Attempts to be entered |
| Return Token | Return a Token if Transaction is Approved |
| Default Currency | This will be USD |
| Require CVV2 | This will provide an option whether CVV2 is required or not. |
| Checkout Button Settings | |
| Button Selection | Button selection from options in library |
| Button Color | Set Button color. |
| Button Type | Set Button type (Raised, Flat). |
| Button Text Font | Set Button Font. |
| Button Text | Set Button Text. |
| Button Text Color | Set Button Text Color. |

5. Select **Save**. Selecting **Save** shows the button code for 'Payment' that will appear on the merchant page
6. Copy the button code by clicking on **Copy Button Code**. A sample button code is provided below.

A sample button code

```

<button id="egdepay-pay-button" style="background-color: rgb(192, 228, 250); color:
rgb(0, 0, 0); padding: 0px;">
<span style="padding: 5px 10px; display: block; float: left; border-radius: inherit; font-
weight: 500;">Check Out</span>
</button><input id="egdepay-amount" type = "hidden" value="0.00"><input id="egdepay-
otac" type = "hidden" value="">
<script type="text/javascript">(function(){document.getElementById("egdepay-
pay-button").addEventListener("click", function(){var edgepayAmount =
document.getElementById("egdepay-amount").value;var edgepayOtac =
document.getElementById("egdepay-
otac").value;window.location.href="https://portal.edgepay-
uat.com/processTransaction?mid=6320340123456902
&amount="+edgepayAmount+"&otac="+edgepayOtac;});})();</script>

```

6.2 STEP 2: PAYMENT BUTTON INTEGRATION

In this step the merchant can embed a **Payment** button on their Checkout webpage. Selecting this button will redirect to the EdgePay Gateway to serve the payment processing page to the customer you created in step 6.1.1.

6.2.1 To integrate the Payment button

1. Generate JavaScript code of button by saving HPP setting in edgepay portal.
2. In the page where the payment button is to be integrated, implement One Time Authentication Code (OTAC) generation by using OTAC API on page load (see 6.2.2). You also need to assign the OTAC from the API to the hidden element with ID: **edgepay-otac**.
3. Populate or map the amount in the *hidden* amount field with ID: **edgepay-amount**.
4. Embed the JavaScript code generated in *Step 1* in the page.

Prior to serving this page to the Customer you will request a ‘One Time Authentication Code’ from EdgePay and include both the OTAC and Amount in hidden fields on the page. NOTE: Some times you may want or need to update the Amount prior to the Customer selecting the “Pay Button.” You can do this with Javascript on the page.

6.2.2 Requesting the OTAC

The One Time Authentication Code is requested from your server prior to serving up your Checkout page with the EdgePay Pay Button. The request/Response is a RESTful message using JSON format. The request will include your Merchant Key from your server to authenticate the request. The response will include the OTAC that will be used to authenticate the request. The OTAC expires in 15 minutes from the request.

```
POST /otac HTTP/1.1
Host: api.edgepay-uat.com
Content-Type: application/json
merchantKey: B9B51FC2490BEC32B6D3B9F4C13ED35D6E753EC4133BDC6E
externalReferencelD: 1234561527119391
transactionDate: 2018-05-23T02:00:00
```

```
{
  "merchantID": "6320340123456902",
  "terminalID": "88800000282601"
}
```

As always, you must update the externalReferncelD to a unique value, (recommend your phone number and increment and set the transactionDate to 'Today UTC'.

The OTAC Service will respond with the following RESTful message:

```
{
  "result": "A",
  "responseCode": "A0000",
  "responseMessage": "Success",
  "merchantID": "6320340123456902",
  "terminalID": "88800000282601",
  "timestamp": "2018-05-23-23-53-45",
  "transactionID": "CS11876175370499715",
  "otac": "6f31eb30-e30a-4c9a-a361-f9ec9391c556"
}
```

If there is an error in the OTAC request, you will see the following response:

```
{
  "result": "C",
  "responseCode": "EP0012",
  "responseMessage": "Authentication failed, please use valid credentials.",
  "merchantID": "6320340123456902",
  "terminalID": "88800000282601",
  "timestamp": "2018-07-16-19-39-04",
}
```

If this condition occurs, there is an issue with your merchantKey that you provided for the request. You may also receive these error conditions as well:

```
{
  "result": "D",
  "responseCode": "EP0102",
  "responseMessage": "The Value '%s' of element '%s' is not valid",
  "merchantID": "6320340123456902",
  "terminalID": "88800000282601",
  "timestamp": "2018-07-16-19-39-04",
}
```

If this occurs, the merchantID or terminalID may be in error.

```
{
  "result": "D",
  "responseCode": "EP0326",
  "responseMessage": "System Error. Please Try Again",
  "merchantID": "6320340123456902",
  "terminalID": "88800000282601",
  "timestamp": "2018-07-16-19-39-04",
}
```

This condition would occur when we have an internal server error. We suggest a retry on this response.

6.3 HOW HPP WORKS

1. With above work complete, when a customer clicks on the *EdgePay Payment* button you created in the Processing Center and added to your Checkout Page, the merchant specific Hosted Payment form is served to your customer.
2. The customer fills out the payment information form and submits.
3. The transaction is submitted to Edgepay for Payment processing. Included in the Hosted Payment Page setup is the Redirect URL our system will use to redirect the response to your server.
4. Once we process the payment request we will generate a HTTP 300 Redirect message that includes the origin information. We will post the Response message to the Card Holder Browser including your Response URL saved in the setup. The Browser will detect the HTTP 300 message and redirect the response message via the

GET method to your Server as a HTTP 200 message. Query the response url string and generate a response back to your customer. This method allows you to control any final messaging. Below is a sample Response Message for an Approved Transaction.

Here is a sample URL string:

Request URL: http://ec2-36-220-202-7.us-west-2.compute.amazonaws.com/home/hpp?result=A&responseCode=A0000&responseMessage=Success&amount=10.01&maskedAccount=401200*****5439&transactionID=CS49700897428090290×tamp=2018-09-28T20:16:34

Request Method: GET

Status Code: 200 OK

Remote Address: 36.22-.202.7:80

Referrer Policy: no-referrer-when-downgrade

Response Headers

Cache-Control: private

Content-Length: 2301

Content-Type: text/html; charset=utf-8

Date: Fri, 28 Sep 2018 20:16:36 GMT

Server: Microsoft-IIS/10.0

Request Headers

Accept:

text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8

Accept-Encoding: gzip, deflate

Accept-Language: en-US,en;q=0.9

Cache-Control: no-cache

Connection: keep-alive

Host: ec2-36-220-202-7.us-west-2.compute.amazonaws.com

Pragma: no-cache

Upgrade-Insecure-Requests: 1

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.3497.100 Safari/537.36

Query String Parameters

result: A

responseCode: A0000

responseMessage: Success

amount: 10.01

maskedAccount: 401200*****5439

transactionID: CS49700897428090290

timestamp: 2018-09-28T20:16:34

We recommend using the Result Code to interpret the correct Work Flow for your pages. Please reference the Response Code Table and AVS Response in Section 4 to see possible responses and recommended actions on generating and delivering Responses to your customer.

7 SHOPPING CART INTEGRATION

EdgePay is currently integrated in Direct Mode to WooCommerce.

8 EDGEPAY BATCH PAY

The EdgePay Batch Upload service extends you the ability to send us a file of transaction to process. For some business situations it is most convenient to send transactions in a file. BatchPay supports that need. A common example is taking shipping records off a CRM to generate Capture transactions.

BatchPay is accessed from the Processing Center. You will need a .csv file to upload transactions for processing.

Selecting BatchPay link from EdgePay menu (Figure: 8-1) allows user to perform following:

- **MID**—To select Merchant ID.
- **Upload Batch File**—For uploading file of transactions in pre-defined format.
- **Report**—Report page shows detailed report of all the items in the batch and status.

Figure: 8-1: Batch Pay Menu

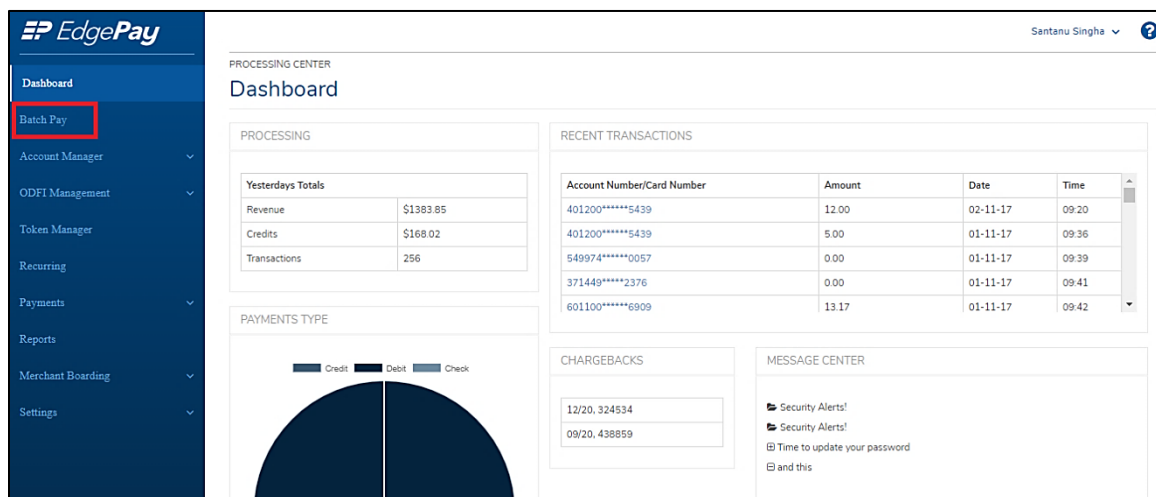


Figure: 8-2: Batch Pay Menu

PROCESSING CENTER

Batch Pay

MID 7991234560720181234

UPLOAD BATCH FILE

Drop your file here!

OR

Select a file

Upload

REPORT

start date

end date

Submit

18 results found

| DATE:TIME | FILENAME | TOTAL TRANS | ANY ERROR |
|---------------------|--|-------------|-----------|
| 2017-10-30 11:10:02 | BatchPayCard_AS_24thOct1.csv | 12 | 12 |
| 2017-10-30 11:10:01 | BatchPayCard_AS_17thOct.csv | 12 | 12 |
| 2017-10-25 07:10:04 | BatchPayCard_AS_25thOctMultipleTimeUploadFrmSelectFile.csv | 12 | 4 |
| 2017-10-25 07:10:04 | BatchPayCard_AS_25thOctMultipleTimeUploadFrmSelectFile.csv | 12 | 4 |
| 2017-10-25 07:10:03 | BatchPayCard_AS_25thOctMultipleTimeUpload.csv | 12 | 4 |
| 2017-10-25 07:10:03 | BatchPayCard_AS_25thOctMultipleTimeUpload.csv | 12 | 4 |
| 2017-10-25 07:10:02 | BatchPayCard_AS_25thOctMultipleTimeUpload.csv | 12 | 4 |
| 2017-10-25 07:10:02 | BatchPayCard_AS_25thOctMultipleTimeUpload.csv | 12 | 4 |
| 2017-10-24 11:10:02 | BatchPayCard_AS_24thOctWithTID.csv | 12 | 4 |
| 2017-10-24 10:10:12 | BatchPayCard_AS_24thOct2.csv | 12 | 7 |

Records per page: 10

8.1 UPLOADING THE BATCH

Upload Batch files allows user to upload file of transactions in pre-defined format (.csv format) as shown in

. User can upload files in 2 ways:

- **Drag and drop**—drop a file onto the page for upload.
- **Browse**—Browse files using the ‘Windows Open’ dialog box.

The upload service will check the file extension and attributes of the file to insure the file conforms to the format specified.

Batch Pay Upload

| DATE:TIME | FILENAME | TOTAL TRANS | ANY ERROR |
|---------------------|--|-------------|-----------|
| 2017-10-30 11:10:02 | BatchPayCard_AS_24thOct1.csv | 12 | 12 |
| 2017-10-30 11:10:01 | BatchPayCard_AS_17thOct.csv | 12 | 12 |
| 2017-10-25 07:10:04 | BatchPayCard_AS_25thOctMultipleTimeUploadFrmSelectFile.csv | 12 | 4 |
| 2017-10-25 07:10:04 | BatchPayCard_AS_25thOctMultipleTimeUploadFrmSelectFile.csv | 12 | 4 |
| 2017-10-25 07:10:03 | BatchPayCard_AS_25thOctMultipleTimeUpload.csv | 12 | 4 |
| 2017-10-25 07:10:03 | BatchPayCard_AS_25thOctMultipleTimeUpload.csv | 12 | 4 |
| 2017-10-25 07:10:02 | BatchPayCard_AS_25thOctMultipleTimeUpload.csv | 12 | 4 |
| 2017-10-25 07:10:02 | BatchPayCard_AS_25thOctMultipleTimeUpload.csv | 12 | 4 |
| 2017-10-24 11:10:02 | BatchPayCard_AS_24thOctWithTID.csv | 12 | 4 |
| 2017-10-24 10:10:12 | BatchPayCard_AS_24thOct2.csv | 12 | 7 |

8.1.1 Drag and drop

To drag and drop batch file for uploading, the file must be in .csv format as shown below.

Batch Pay Drag-and-Drop File

PROCESSING CENTER

Batch Pay

MID 7991234560720181234

UPLOAD BATCH FILE

Drop your file here

BatchPayFile AS_27thOctCaptureOnly - Copy

OR

Select a file

Upload

REPORT

start date end date Submit

18 results found

| DATE:TIME | FILENAME | TOTAL TRANS | ANY ERROR |
|---------------------|--|-------------|-----------|
| 2017-10-30 11:10:02 | BatchPayCard_AS_24thOct1.csv | 12 | 12 |
| 2017-10-30 11:10:01 | BatchPayCard_AS_17thOct.csv | 12 | 12 |
| 2017-10-25 07:10:04 | BatchPayCard_AS_25thOctMultipleTimeUploadFrmSelectFile.csv | 12 | 4 |
| 2017-10-25 07:10:04 | BatchPayCard_AS_25thOctMultipleTimeUploadFrmSelectFile.csv | 12 | 4 |
| 2017-10-25 07:10:03 | BatchPayCard_AS_25thOctMultipleTimeUpload.csv | 12 | 4 |
| 2017-10-25 07:10:03 | BatchPayCard_AS_25thOctMultipleTimeUpload.csv | 12 | 4 |
| 2017-10-25 07:10:02 | BatchPayCard_AS_25thOctMultipleTimeUpload.csv | 12 | 4 |
| 2017-10-25 07:10:02 | BatchPayCard_AS_25thOctMultipleTimeUpload.csv | 12 | 4 |
| 2017-10-24 11:10:02 | BatchPayCard_AS_24thOctWithTID.csv | 12 | 4 |
| 2017-10-24 10:10:12 | BatchPayCard_AS_24thOct2.csv | 12 | 7 |

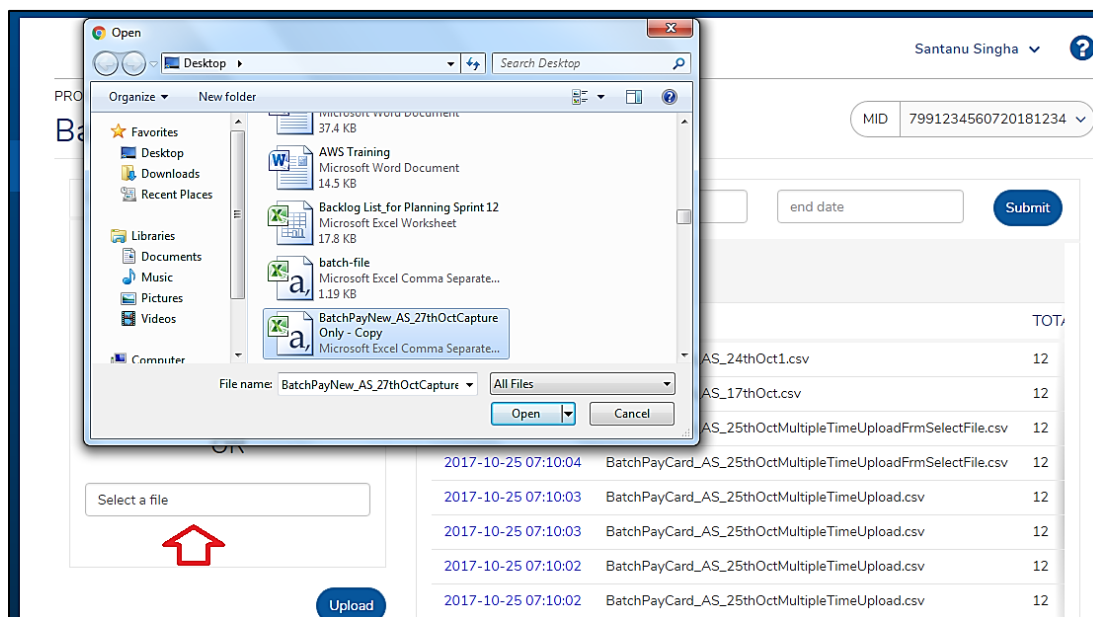
1 2 > >>

Records per page: 10

8.1.2 Browse

From Upload Batch file menu, choose select a file menu, browse, and select the file from your system as shown above. The file must be in .csv format. After selecting the file, click on the upload button.

Browse Batch File for Upload



Selecting the “Upload” button will result in a “Success Message” notification as shown below.

PROCESSING CENTER

Batch Pay

MID 7991234560720181234

UPLOAD BATCH FILE

BatchPayNew_AS_27thOctCaptureOnly - C

OR

Select a file

Upload

REPORT

start date

end date

Submit

18 results found

| DATE/TIME | FILENAME | TOTAL TRANS | ANY ERROR |
|---------------------|---|-------------|-----------|
| | d_AS_24thOct1.csv | 12 | 12 |
| | d_AS_17thOct.csv | 12 | 12 |
| | d_AS_25thOctMultipleTimeUploadFrmSelectFile.csv | 12 | 4 |
| | d_AS_25thOctMultipleTimeUploadFrmSelectFile.csv | 12 | 4 |
| 2017-10-25 07:10:03 | BatchPayCard_AS_25thOctMultipleTimeUpload.csv | 12 | 4 |
| 2017-10-25 07:10:03 | BatchPayCard_AS_25thOctMultipleTimeUpload.csv | 12 | 4 |
| 2017-10-25 07:10:02 | BatchPayCard_AS_25thOctMultipleTimeUpload.csv | 12 | 4 |
| 2017-10-25 07:10:02 | BatchPayCard_AS_25thOctMultipleTimeUpload.csv | 12 | 4 |
| 2017-10-24 11:10:02 | BatchPayCard_AS_24thOctWithTID.csv | 12 | 4 |
| 2017-10-24 10:10:12 | BatchPayCard_AS_24thOct2.csv | 12 | 7 |

1

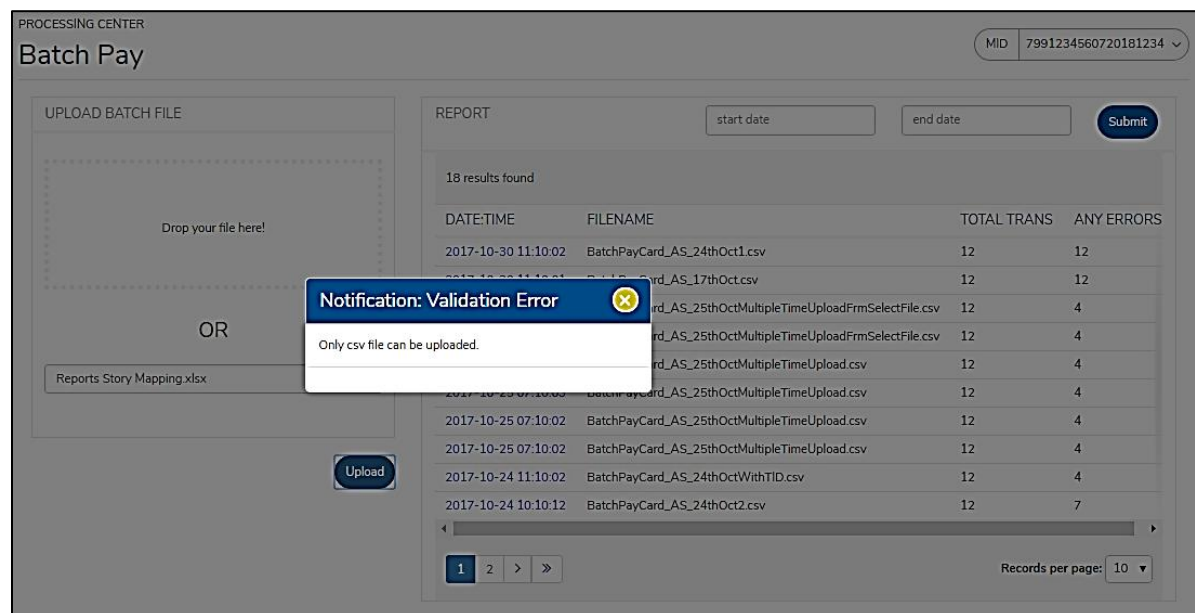
2

>

>>

Records per page: 10

Batch File Upload Failure (File Format Validation)



8.2 BATCHPAY REPORTS

The BatchPay Reports provides a list of the last *ten* days batch report that is generated after the batch is processed as shown in 2. The BatchPay Report allows user to generate following:

- A link to specific reports.
- Selecting Submit button by choosing a start date and end date, a report range can be viewed.
- Report can be filtered by selecting MID from drop-down.

Figure: 8-2: BatchPay Report

| DATE:TIME | FILENAME | TOTAL TRANS | ANY ERROR |
|---------------------|--|-------------|-----------|
| 2017-10-30 11:10:02 | BatchPayCard_AS_24thOct1.csv | 12 | 12 |
| 2017-10-30 11:10:01 | BatchPayCard_AS_17thOct.csv | 12 | 12 |
| 2017-10-25 07:10:04 | BatchPayCard_AS_25thOctMultipleTimeUploadFrmSelectFile.csv | 12 | 4 |
| 2017-10-25 07:10:04 | BatchPayCard_AS_25thOctMultipleTimeUploadFrmSelectFile.csv | 12 | 4 |
| 2017-10-25 07:10:03 | BatchPayCard_AS_25thOctMultipleTimeUpload.csv | 12 | 4 |
| 2017-10-25 07:10:03 | BatchPayCard_AS_25thOctMultipleTimeUpload.csv | 12 | 4 |
| 2017-10-25 07:10:02 | BatchPayCard_AS_25thOctMultipleTimeUpload.csv | 12 | 4 |
| 2017-10-25 07:10:02 | BatchPayCard_AS_25thOctMultipleTimeUpload.csv | 12 | 4 |
| 2017-10-24 11:10:02 | BatchPayCard_AS_24thOctWithTID.csv | 12 | 4 |
| 2017-10-24 10:10:12 | BatchPayCard_AS_24thOct2.csv | 12 | 7 |

The report section displays the fields shown in 8-3.

Table: 8-3: BatchPay Report

| Fields | Description |
|-------------|--|
| DATE:TIME | Displays the date and time of the batch file when it was uploaded to the system. |
| FILENAME | Shows the Batch File name. |
| TOTAL TRANS | Shows the number of total transaction in the Batch File. |
| ANY ERRORS | Shows the number of Error if any. |

Note : The **DATE:TIME** field is a hyperlink. Selecting the hyper link the Batch Summary Report opens up in a new page.

The Batch Summary Report is shown below.

Batch Summary Report

| | | | | | | | | | |
|--|--------|-----------------|-------------------|---------------------|----------------|---------------|-----------|----------|---------|
| PROCESSING CENTER | | | | | | | | | |
| Batch Summary Report | | | | | | | | | |
| MID: 7991234560720181234 | | | | | | | | | |
| <div>DOWNLOAD</div> <div>BACK</div> | | | | | | | | | |
| TOTAL AMOUNT: | | TOTAL CREDITS: | | TOTAL TRANSACTIONS: | | TOTAL DEBITS: | | | |
| \$570.67 | | \$0.00 | | 12 | | \$0.00 | | | |
| DATE:TIME | AMOUNT | CUSTOMER NUMBER | CARD/CHECK NUMBER | TYPE | TRANSACTION ID | CUSTOMER NAME | RECURRING | CURRENCY | STATUS |
| 2017-10-30 11:00:02 | 70.00 | | | Unknown | | JohnK | No | USD | DECLINE |
| 2017-10-30 11:00:02 | 90.09 | | 401200*****5439 | Visa | | JohnK | No | USD | DECLINE |
| 2017-10-30 11:00:02 | 0.00 | | | Unknown | | | No | USD | DECLINE |
| 2017-10-30 11:00:02 | 81.00 | | | Unknown | | JohnK | No | USD | DECLINE |
| 2017-10-30 11:00:02 | 77.11 | | 371449*****8431 | AmericanExpress | | JohnK | No | USD | DECLINE |
| 2017-10-30 11:00:02 | 97.09 | | 401200*****5439 | Visa | | JohnK | No | USD | DECLINE |
| 2017-10-30 11:00:02 | 0.00 | | | Unknown | | | No | USD | DECLINE |
| 2017-10-30 11:00:02 | 101.20 | | 601100*****6909 | Discover | | JohnK | No | USD | DECLINE |
| 2017-10-30 11:00:02 | 43.08 | | 401200*****5439 | Visa | | JohnK | No | USD | DECLINE |
| 2017-10-30 11:00:01 | 11.10 | | 401200*****5439 | Visa | | JohnK | No | USD | DECLINE |
| <div>1 2 > >></div> <div>Records per page: 10</div> | | | | | | | | | |

The batch summary reports header displays the fields shown in the table below :

Table: 8-1: Batch Summary Report

| Fields | Description |
|--------------------|---|
| Total Amount | Displays total amount of the transactions in the batch file. |
| Total Credits | Displays total credits of the transactions in the batch file. |
| Total Transactions | Displays the total number of transactions in the Batch file. |
| Total Debits | Displays total debits of the transactions in the batch file. |

The batch summary report provides the following:


- Information about the individual transactions in the batch file.
- It shows information about the following fields for transactions present in the batch file:

- DATE:TIME
- Amount
- Customer Number
- Card/ Check Number
- Type
- Transaction ID
- Customer Name
- Recurring
- Currency
- Status

8.2.1 Report Download

Selecting Download button user can download the Batch Summary Report in .CSV format as shown below

Batch Summary Report

| PROCESSING CENTER | | | | | | | | |
|--|----------------|---------------------|-------------------|-----------------|----------------|---------------|-----------|----------|
| Batch Summary Report | | | | | | | | |
| MID: 7991234560720181234 | | | | | | | | |
| <div>  DOWNLOAD BACK </div> | | | | | | | | |
| TOTAL AMOUNT: | TOTAL CREDITS: | TOTAL TRANSACTIONS: | TOTAL DEBITS: | | | | | |
| \$570.67 | \$0.00 | 12 | \$0.00 | | | | | |
| DATE:TIME | AMOUNT | CUSTOMER NUMBER | CARD/CHECK NUMBER | TYPE | TRANSACTION ID | CUSTOMER NAME | RECURRING | CURRENCY |
| 2017-10-30 11:00:02 | 70.00 | | | Unknown | | JohnK | No | USD |
| 2017-10-30 11:00:02 | 90.09 | | 401200*****5439 | Visa | | JohnK | No | USD |
| 2017-10-30 11:00:02 | 0.00 | | | Unknown | | | No | USD |
| 2017-10-30 11:00:02 | 81.00 | | | Unknown | | JohnK | No | USD |
| 2017-10-30 11:00:02 | 77.11 | | 371449*****8431 | AmericanExpress | | JohnK | No | USD |
| 2017-10-30 11:00:02 | 97.09 | | 401200*****5439 | Visa | | JohnK | No | USD |
| 2017-10-30 11:00:02 | 0.00 | | | Unknown | | | No | USD |
| 2017-10-30 11:00:02 | 101.20 | | 601100*****6909 | Discover | | JohnK | No | USD |
| 2017-10-30 11:00:02 | 43.08 | | 401200*****5439 | Visa | | JohnK | No | USD |
| 2017-10-30 11:00:01 | 11.10 | | 401200*****5439 | Visa | | JohnK | No | USD |
| <div> 1 2 > » </div> <div>Records per page: 10</div> | | | | | | | | |

Selecting Back button will take user back to Batch Pay landing page.

8.2.2 Transaction Detail Report

The DATE:TIME field is hyperlinked. Once user clicks on the DATE:TIME of a transaction, the hyperlinks directs the user to Transaction Detail Report (Described in Report Section) page. Transaction Detail Report shows the details of the transaction as shown below.

Batch Summary Report–Transaction Detail

PROCESSING CENTER

Transaction Detail Report

Back

TRANSACTION DETAIL REPORT

| | |
|------------------|-------------------------|
| Status | DECLINED |
| Date:Time | 2017-10-30 11:00:02 |
| Customer Number | |
| Card Number | |
| Expiry Date | |
| Check Account | |
| Check Route | |
| Transaction Type | Unknown |
| Transaction ID | |
| Customer Name | JohnK |
| Billing Address | billingAddress Milpitas |
| Billing Zip | 94404 |
| POS Entry | |
| Terminal ID | 866688889999000000 |

PAYMENT RESULTS

Refund

| | |
|--------------------|-----------------------|
| Result | |
| Transaction ID | |
| Auth Code | |
| Token ID | 5819690668261220 |
| Processor | |
| Response Code | CS012 |
| Response Message | Duplicate Transaction |
| AVS Response | |
| CVV Response | |
| Commercial Card | |
| Transaction Amount | 70.00 |
| Tax Amount | 0.00 |
| Tip Amount | 0.00 |

ACTION CODES USED

CVV Only

CONDITIONAL DATA

8.3 BATCHPAY FILE FORMAT

BatchPay file will be of CSV format with fields as stated below

Table: 8-2: BatchPay File Specification

| Fields | Description |
|----------------------|-------------------------|
| externalReferenceID | External Reference ID |
| cardNumber | Card Number |
| cardExpirationDate | Card Expiration Date |
| tokenID | Token ID |
| token | Token |
| eCheckRoute | eCheck Route |
| eCheckAccount | eCheck Account |
| eCheckTransType | eCheck Transaction Type |
| eCheckSerialNumber | eCheck Serial Number |
| eCheckAccountType | eCheck Account Type |
| goodsOrService | Goods Or Service |
| receivingCompanyName | Receiving Company Name |

| Fields | Description |
|-----------------------|--------------------------|
| paymentTypeCode | Payment Type Code |
| paymentDataInput | Payment Data Input |
| amount | Amount |
| currencyCode | Currency Code |
| customerName | Customer Name |
| customerNumber | Customer Number |
| captureOnly | Capture Only |
| captureToo | Capture Too |
| processor | Processor |
| billingAddress | Billing Address |
| billingCity | Billing City |
| billingState | Billing State |
| billingZip | Billing Zip |
| merchantField1 | Merchant Field 1 |
| merchantField2 | Merchant Field 2 |
| posData | Pos Data |
| cvv2 | Cvv2 |
| adjust | Adjust |
| refund | Refund |
| tip | Tip |
| tax | Tax |
| healthCare | Health Care |
| healthCareAccountType | Health Care Account Type |
| prescriptionAmt | Prescription Amount |
| visionAmt | Vision Amount |
| dentalAmt | Dental Amount |
| clinicAmt | Clinic Amount |
| posQualified | POS Qualified |
| rxNumber | Rx Number |
| level2 | Level2 |
| level2Auth | Level2 Auth |
| level2Capture | Level2 Capture |
| taxAmount | Tax Amount |
| taxCategory | Tax Category |
| taxRate | Tax Rate |








| Fields | Description |
|-------------------------|---------------------------|
| transactionDate | Transaction Date |
| level3 | Level3 |
| level3Auth | Level3 Auth |
| level3Capture | Level3 Capture |
| shipZipCode | Ship Zip Code |
| destinationZipCode | Destination Zip Code |
| invoiceNumber | Invoice Number |
| itemCode | Item Code |
| itemCommunityCode | Item Community Code |
| ItemDescription | Item Description |
| itemQuantity | Item Quantity |
| itemUnitOfMeasure | Item Unit Of Measure |
| itemExtendedAmount | Item Extended Amount |
| freightAmount | Freight Amount |
| dutyAmount | Duty Amount |
| purchaseOrder | Purchase Order |
| customerVATNumber | Customer VAT Number |
| summaryCommodityCode | Summary Commodity Code |
| vatInvoice | Vat Invoice |
| itemDiscountName | Item Discount Name |
| itemDiscountAmount | Item Discount Amount |
| itemDiscountPercentage | Item Discount Percentage |
| itemDiscountType | Item Discount Type |
| destinationCountryCode | Destination Country Code |
| supplierReferenceNumber | Supplier Reference Number |
| transactionId | Transaction Id |
| cardOnFile | Card On File |
| recurring | Recurring |
| billingType | Billing Type |
| paymentCount | Payment Count |
| currentPaymentCount | Current Payment Count |
| chargeDescriptor | Charge Descriptor |
| transDiscountName | Trans Discount Name |
| transDiscountAmount | Trans Discount Amount |
| authCode | Auth Code |













A sample Batch File is attached for reference.















Batchpay.csv

8.4 LIST OF SAMPLE BATCH FILES FOR DIFFERENT ACH OR CARD PAYMENT TYPES:

| Payment Type | Sample Batch Files | Fields value need to be changed before execute |
|---------------------|--|---|
| AuthOnly-Level1 |  AuthOnly-Level1.csv | transactionDate externalReferenceID |
| AuthOnly-Level2 |  AuthOnly-Level2.csv | transactionDate externalReferenceID |
| AuthOnly-Level3 |  AuthOnly-Level3.csv | transactionDate externalReferenceID |
| Capture-Level1 |  Capture-Level1.csv | transactionId transactionDate externalReferenceID |
| ForceCapture-Level1 |  ForceCapture-Level1.csv | transactionDate externalReferenceID |
| ForceCapture-Level2 |  ForceCapture-Level2.csv | transactionDate externalReferenceID |
| ForceCapture-Level3 |  ForceCapture-Level3.csv | transactionDate externalReferenceID |

| Payment Type | Sample Batch Files | Fields value need to be changed before execute |
|-------------------------|---|---|
| Refund |  Refund.csv | transactionDate externalReferenceID transactionId |
| Sale-HealthCare |  Sale-HealthCare.csv | transactionDate externalReferenceID |
| Sale-Level1 |  Sale-Level1.csv | transactionDate externalReferenceID |
| Sale-Level2 |  Sale-Level2.csv | transactionDate externalReferenceID |
| Sale-Level3 |  Sale-Level3.csv | transactionDate externalReferenceID |
| Adjust |  Adjust.csv | transactionDate externalReferenceID transactionId |
| ACH-ARC-Checking |  ACH-ARC-Checking.csv | transactionDate externalReferenceID |
| ACH-ARC- General Ledger |  ACH-ARC-GL.csv | transactionDate externalReferenceID |
| ACH-ARC-Savings |  ACH-ARC-Savings.csv | transactionDate externalReferenceID |
| ACH-CCD-Checking |  ACH-CCD-Checking.csv | transactionDate externalReferenceID |
| ACH-CCD- General Ledger |  ACH-CCD-GL.csv | transactionDate externalReferenceID |
| ACH-CCD-Savings |  ACH-CCD-Savings.csv | transactionDate externalReferenceID |

| Payment Type | Sample Batch Files | Fields value need to be changed before execute |
|-------------------------|---|--|
| ACH-POP-Checking |  ACH-POP-Checking.csv | transactionDate externalReferenceID |
| ACH-POP- General Ledger |  ACH-POP-GL.csv | transactionDate externalReferenceID |
| ACH-POP-Savings |  ACH-POP-Savings.csv | transactionDate externalReferenceID |
| ACH-PPD-Checking |  ACH-PPD-Checking.csv | transactionDate externalReferenceID |
| ACH-PPD- General Ledger |  ACH-PPD-GL.csv | transactionDate externalReferenceID |
| ACH-PPD-Savings |  ACH-PPD-Savings.csv | transactionDate externalReferenceID |
| ACH-TEL-Checking |  ACH-TEL-Checking.csv | transactionDate externalReferenceID |
| ACH-TEL- General Ledger |  ACH-TEL-GL.csv | transactionDate externalReferenceID |
| ACH-TEL-Savings |  ACH-TEL-Savings.csv | transactionDate externalReferenceID |
| ACH-WEB-Checking |  ACH-WEB-Checking.csv | transactionDate externalReferenceID |
| ACH-WEB- General Ledger |  ACH-WEB-GL.csv | transactionDate externalReferenceID |
| ACH-WEB-Savings |  ACH-WEB-Savings.csv | transactionDate externalReferenceID |

9 EDGEPAY RECURRING PAYMENT MANAGER

The EdgePay Recurring Payment Manager is a set of services that is extended through the Processing Center. Additionally, the Application Protocol Interface allows for the addition of Customers to existing Billing Plans.

For developers please contact Developer Support to setup access to our UI for development and testing services.

By using the RPM, you can set up Products or Services under a Billing Plan and then add Customers to the Billing Plan.

The system will take those Customers and Automatically Bill them on the schedules you have pre-defined in the Billing Plan.

You can setup as many Billing Plans as you need to accommodate the variability of your business or its products or services.

9.1 RECURRING PAYMENT FEATURES:

1. A Customer may have more than 1 contract
2. Retry Options to establish the number of Retries and their timing
3. Backup Payment if a Customer wishes to provide an alternative to the primary payment method
4. Account Updater to update cards on file
5. Reports for Declined Transactions, Account Updater failures, and Expiring Contracts
6. Notice system of Recurring Events via email

9.2 BENEFITS OF RECURRING BILLING:

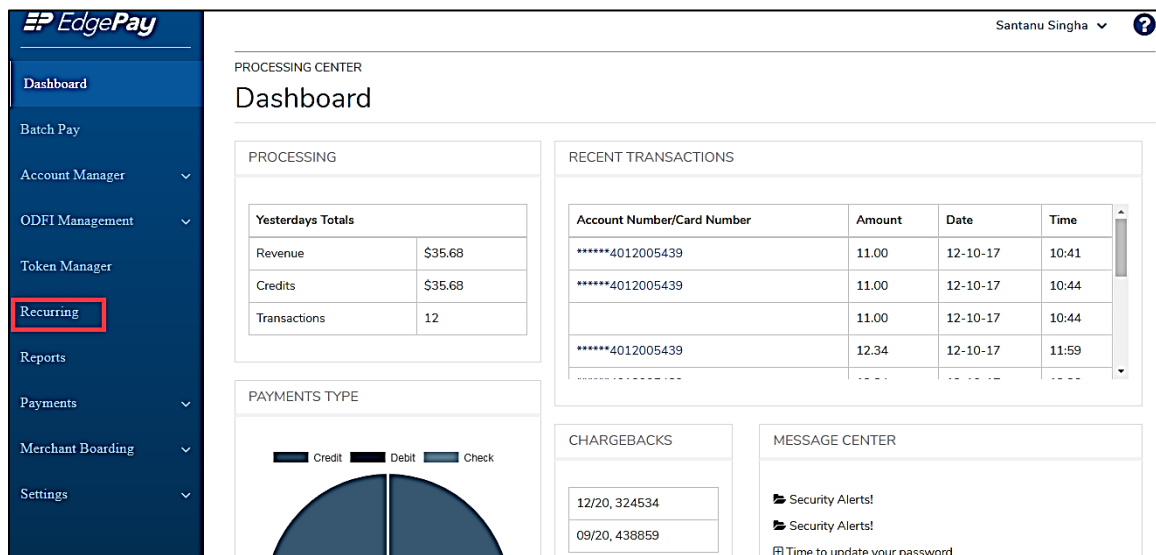
1. **Enhance customer relationships:** Recurring billing improves the relationship between the business and their customers. Automated recurring payments offers customer convenience and operational efficiency. Likewise, customers who register for recurring billing are more apt to stay with a business long term, as it is easier for them to maintain their future payments.
2. **Increase Administrative Efficiency:** Recurring payments automates manual business tasks. With automatic billing, set up a payment schedule, and then easily maintain infrequent changes that may occur with the payment method and schedule.

3. **Reduce Declines:** the automated recurring billing program can significantly reduce declined payments with the Account Updater. A card on file may expire or be replaced as a result of a new card being issued or stolen card. Account Updater services will reduce declines due to expired or replaced cards. This service maintains expiring and replacement card numbers for cards on file.
4. **Encrypted Security:** Recurring billing also gives a business the peace of mind to securely manage their customer's confidential information. Electronic billing reduces the distribution and replication of secure information through online encryption, token processing and assures PCI Compliance

9.3 ADDING BILLING PLANS

To access Recurring click the **Recurring** link from left navigation menu, the Recurring Payments landing page will open as shown in Figure 9-1.

Figure 9-1: Recurring Menu



A Billing Plan is added from **Add Billing Plan** page.

To access Add billing plan page

Click on the **Add billing plan** button in the Recurring Payments Landing Page (

Figure 9-2.)

Figure 9-2: Add Billing Plan Button

PROCESSING CENTER

MID 9830433771

Search Customer

FIND A CUSTOMER/CONTRACT

Phone Number

Customer Name

Customer Number

Billing Zip

Email Address

Token

Search Customer

REPORTS

Account Updater **Declined Transaction** **Expiring Contract**

FIND A BILLING PLAN

Billing Plan **Search**

| Billing Plan Name | Status | Action |
|-------------------|--------|-------------|
| 3days Plan | A | View |
| 2days Plan | A | View |
| D.Plan | A | View |
| Weekly Plan | A | View |
| AnkurDaily | A | View |

Notice Maintenance **Add Billing Plan** **Add Customer**

A new page for Add Billing Plan will be shown (Figure 9-3.)

Figure 9-3: Add Billing Plan Page

PROCESSING CENTER

MID 7991234560720181234

Add Billing Plan

ENTER BILLING PLAN

Billing Plan

Status **Select Status** ▾

Fee Desc 1

Fee Amount 1 **+**

Frequency 1 **Select Frequency** ▾

Save

To add a billing plan the user needs to assign a contract. Billing amount, frequency can be added from the Add Billing Plan page. The page has these fields for entry:

Table 9-3: Add Billing Plan

| Convention | Description |
|--------------|---|
| Billing Plan | Enter description for billing plan. |
| Status | Select status of the plan: Active or Suspend . |
| Fee Desc 1 | Enter fee description. A contract can have more than one fee. |
| Fee Amount 1 | Enter Fee amount. |
| Frequency | <p>Select Frequency. Frequency can be selected from any of the following:</p> <ul style="list-style-type: none"> • Weekly • Semi-Monthly • Monthly • Quarterly, • Semi Yearly • Annual • Specific days - Up to 60 days specified |

If user enters invalid data or no data, the system will not allow submission of the form and will show erroneous fields in red as shown in (Figure 2-4.)

Figure 9-4: Add Billing Plan Page – Validation Error

PROCESSING CENTER

Add Billing Plan

You have entered invalid data. Please try again. MID 7991234560720181234

ENTER BILLING PLAN

Billing Plan Enter Billing Plan

Status Select Status

Fee Desc 1 Enter Fee Desc

Fee Amount 1 Enter Fee Amount

Frequency 1 Select Frequency

Save

Once a billing plan is successfully added, the user will view success notification.

Figure 9-5: Add Billing Plan Page – Success

PROCESSING CENTER

Santanu Singha

MID 7991234560720181234

Add Billing Plan

ENTER BILLING PLAN

Billing Plan: 1234bill

Fee Desc 1: 1234bill

Frequency 1: Weekly

Status: Active

Fee Amount 1: 0.01

Notification: Success

Successful

Save

9.4 ADDING CUSTOMERS

Before adding a new customer, you need to add billing plans (if no billing plans has added before; refer to Add Billing Plan), so that it is displayed at billing plan drop-down section from add customer page.

Add Customer allows user to add a customer and their associated contracts.

To add a new customer

Click on the **Add Customer** button in the recurring payments landing page.

Figure 9-6: Add Customer Button

PROCESSING CENTER

MID 7991234560720181234

Search Customer

FIND A CUSTOMER/CONTRACT

Phone Number: Enter Phone Number

Customer Name: Enter Customer Name

Customer Number: Enter Customer Number

Billing Zip: Enter Billing Zip

Email Address: Enter Email

Token: Enter Token Number

Search Customer

REPORTS

Account Updater Declined Transaction Expiring Contract

FIND A BILLING PLAN

Billing Plan: Enter Billing Plan Search

| Billing Plan Name | Status | Action |
|-------------------|--------|--------|
| RPM_TESTPlan | A | View |
| auto1 | A | View |
| auto | A | View |
| auto123 | A | View |
| qut02 | A | View |

Notice Maintenance Add Billing Plan Add Customer

Clicking the Add Customer button opens a new page for Add Customer as shown in Figure 9-7:

Figure 9-7: Add Customer Button

Enter the customer details and customer contact fields in this page. A customer may have multiple contracts. These contact details can also be added here.

After entering all customer and contracts related information, click **Add Customer** button. This will allow user to register the customer to EdgePay system.

Clicking the **Cancel** button will take user back to the Recurring payments landing page as shown in Figure 9-1.

Table 9-7: Add Customer

| Convention | Description |
|---------------|---|
| Customer Name | Enter your customer name up to 30 alphanumeric character |

| | |
|------------------|---|
| Customer Number | Enter customer number up to 25 alphanumeric character assigned by the merchant. |
| Billing Address | Enter billing address up to 200 alphanumeric character |
| Billing City | Enter billing city up to 50 alphanumeric character |
| Billing State | Select billing state from drop-down. |
| Billing Zip | Enter billing ZIP code up to 9 numeric characters. |
| Shipping Address | Enter shipping address up to 200 alphanumeric characters. |
| Shipping City | Enter shipping city. |
| Shipping State | Select shipping state from drop-down. |
| Shipping Zip | Enter shipping zip up to 9 numeric characters. |
| Phone Number | Enter your phone number. |
| Email | Enter your Email ID. |
| Type | <p>Select type, either Card, ACH or Token from drop-down</p> <ul style="list-style-type: none"> • If card is selected, the form expands and shows additional fields for: <ul style="list-style-type: none"> – Card Number—Enter Card Number. – Card Expiry Date—Enter card expiry date. • If ACH is selected, the form expands and shows additional fields for: <ul style="list-style-type: none"> – Account Number—Enter Account Number. – Routing Number—Enter Routing Number. – Account Type—Select Account type, either saving or checking or General Ledger. • If Token is selected, the form expands and shows additional field for: <ul style="list-style-type: none"> – Token—Enter Token Number. |
| Currency | Select Currency, USD. For now the system only allows USD. |

| | |
|------------------------|--|
| Status | Select customer status: Active or Suspend from drop-down |
| Payment Data Input | Select Payment Data input from drop-down: <ul style="list-style-type: none"> • Phone • Mail • Internet • Swipe • Manual |
| Member Username | Enter Username |
| Member Password | Enter Password |
| Url | Enter Url |
| Affiliate | Enter Affiliate |
| Contract No 1 | Shows contract number. |
| Contract Description 1 | Enter Contract Description. |
| Status 1 | Select Contract Status: Active or Suspend from drop-down. |
| Billing Plan 1 | Select billing plan form drop-down. |
| Start Date 1 | Click on this field to show the calendar and the select start date from calendar. |

If the user enters invalid data, the system will not allow the submission of the form. The system will show error for invalid data as shown in Figure 9-2.

Figure 9-2: Add Customer Page – Validation Error

PROCESSING CENTER

Add Customer

You have entered invalid data. Please try again.

ENTER CUSTOMER DETAILS

| | | | |
|--------------------|--|-----------------|--|
| Customer Name | <input type="text" value="Enter Customer Name"/> | Customer Number | <input type="text" value="Enter Customer Number"/> |
| Billing Address | <input type="text" value="Enter Billing Address"/> | Billing City | <input type="text" value="Enter Billing City"/> |
| Billing State | <input type="text" value="Select Billing State"/> | Billing Zip | <input type="text" value="Enter Billing Zip"/> |
| Shipping Address | <input type="text" value="Enter Shipping Address"/> | Shipping City | <input type="text" value="Enter Shipping City"/> |
| Shipping State | <input type="text" value="Select Shipping State"/> | Shipping Zip | <input type="text" value="Enter Shipping Zip"/> |
| Phone Number | <input type="text" value="Enter Phone Number"/> | Email | <input type="text" value="Enter Email"/> |
| Type | <input type="text" value="Select Type"/> | Status | <input type="text" value="Select Status"/> |
| Currency | <input type="text" value="Select Currency"/> | | |
| Payment Date Input | <input type="text" value="Select Payment Date Input"/> | | |

ADD CONTRACT DETAILS

| | | | |
|----------------------|--|--------------------------|--|
| Contract No 1 | <input type="text" value=""/> | Contract Description 1 | <input type="text" value="Enter Contract Description"/> |
| Status 1 | <input type="text" value="Select Status"/> | Billing Plan 1 | <input type="text" value="Select Billing Plan"/> |
| Start Date 1 | <input type="text" value="Enter Start Date"/> | End Date 1 | <input type="text" value="Enter End Date"/> |
| Email Receipt 1 | <input type="text" value="Select Email Receipt"/> | Email Schedule Payment 1 | <input type="text" value="Select Email Schedule Payment"/> |
| Email Contract End 1 | <input type="text" value="Select Email Contract End"/> | Email Contract Cancel 1 | <input type="text" value="Select Email Contract Cancel"/> |
| Taxable 1 | <input type="text" value="Select Taxable"/> | Tax Rate 1 | <input type="text" value="Enter Tax Rate"/> |

Cancel Add Customer

If the customer already exists in the system, the system will show a notification:

Figure 9-3: Add Customer Page – Duplicate Information Error

ADD CONTRACT DETAILS

Notification: Validation Error

Customer already exists

| | | | |
|----------------------|---|--------------------------|---|
| Contract No 1 | <input type="text" value=""/> | Contract Description 1 | <input type="text" value="Santanu Contract"/> |
| Status 1 | <input type="text" value="Active"/> | Billing Plan 1 | <input type="text" value="auto1"/> |
| Start Date 1 | <input type="text" value="2017-11-03"/> | End Date 1 | <input type="text" value="2017-12-31"/> |
| Email Receipt 1 | <input type="text" value="Yes"/> | Email Schedule Payment 1 | <input type="text" value="Yes"/> |
| Email Contract End 1 | <input type="text" value="Yes"/> | Email Contract Cancel 1 | <input type="text" value="Yes"/> |
| Taxable 1 | <input type="text" value="Yes"/> | Tax Rate 1 | <input type="text" value="0.01"/> |

Cancel Add Customer

Once customer is successfully added, the system will show the following notification:

Figure 9-4: Add Customer Page – Success

The screenshot shows a web form titled "ADD CONTRACT DETAILS". A blue notification box with a green 'X' icon and the text "Notification: Success" and "Customer has been saved successfully." is overlaid on the form. The form contains several input fields and dropdown menus: "Contract No 1" (text input), "Status 1" (dropdown menu with "Active" selected), "Start Date 1" (date input with "2017-11-03"), "Email Receipt 1" (dropdown menu with "Yes" selected), "Email Contract End 1" (dropdown menu with "Yes" selected), "Taxable 1" (dropdown menu with "Yes" selected), "Contract Description 1" (text input with "2433333333333"), "Billing Plan 1" (dropdown menu with "auto" selected), "End Date 1" (date input with "2017-12-30"), "Email Schedule Payment 1" (dropdown menu with "Yes" selected), "Email Contract Cancel 1" (dropdown menu with "Yes" selected), and "Tax Rate 1" (text input with "0.1"). At the bottom right, there are "Cancel" and "Add Customer" buttons.

9.5 RECURRING CUSTOMER ADD API

Below is the layout for the Payment Request Header

9.5.1.1 Recurring Customer Add Header

| Header | Character | Max | Notes |
|--|-----------|-----|--|
| POST /payment HTTP/ 1.1 | AN | | Current HTTP version |
| HOST: /api.edgepay- uat.com/recurring/customer/add | AN | | The API Request location |
| Content-Type: application/json | AN | | |
| merchantKey: 67168CA2B96231F4BFF26E38EFFBA2FDF E4001DFCBABA8EF | AN | 48 | This is a TEST KEY. Please use for testing purposes only |
| externalRefernceID: 1234561523473261 | N | 16 | Unique ID that must be enforced. Replace each time with a new Unique Key. And please don't use "1" as you will most likely get a Duplicate record response! |
| transactionDate: 2018-05-23T02:00:00 | Calendar | 20 | Set to today, UTC |

9.5.2 The Recurring Customer Add JSON Body

The Body of the Recurring Customer Add request will vary according to your specific business needs. Some items marked as **BOLD** are required but the others are Conditional or Optional based on your specific business or transaction needs. Let start with a few simple examples:

Simple Customer Add Transaction with cURL.

Feel free to drop this into a Terminal to test it out and get a feel for the results. You must update the externalReferenceID to a unique value and reset the transactionDate to today, UTC

```
1.  curl -X POST \
    https://api.edgepay-uat.com/recurring/customer/add \ -H 'Cache-Control: no-cache' -
    H 'externalReferenceID: 1234561528752675' -
    H 'merchantKey: 67168CA2B96231F4BFF26E38EFFBA2FD4001DFCBABA8EF
2.  \ -H 'transactionDate: 2018-06-11T02:00:00' -d '{
    "merchantID":"6320340123456903
3.  ", "terminalID":"88800000282601", "actionCode":"A", "cardNumber":"4761739001010119", "cardExpiration
    Date":"0622", "customerNumber":"8000465765646", "customerName":"James
    Morgan", "billingAddress":"25, Spilburg lane", "billingCity":"California", "billingState":"AL",
    "billingZip":"56431", "shippingAddress":"Miller street", "shippingCity":"New York",
    "shippingState":"AR", "shippingZip":"65401", "phoneNumber":"897654546654", "eMailAddress":"james@g
    mail.com", "paymentDataInput":"Mail", "customerCurrency":"840", "contractNbr":[
4.  {
5.  "status":"A", "billingPlan":"Gym Plan", "startDate":"2018-07-18", "endDate":"2019-07-
    18", "eMailReceipt":"Y", "eMailSchedulePayment":"N", "eMailContractEnd":"Y", "eMailContractCancel":"Y",
    "taxable":"Y", "taxRate":"4.0", "contractDescription":"Monthly Subscription"
6.  },
    ]
    }
    "
```

Here is a Java Example:

```
1.  OkHttpClient client = new OkHttpClient();
2.  MediaType mediaType = MediaType.parse("application/json");
    RequestBody body = RequestBody.create(mediaType, "{\r\n  \"merchantID\":\"6320340123456903
3.  \",\r\n  \"terminalID\":\"88800000282601\", \r\n  \"actionCode\":\"A\", \r\n
    \"cardNumber\":\"4761739001010119\", \r\n  \"cardExpirationDate\":\"0622\", \r\n
    \"customerNumber\":\"8000465765648\", \r\n  \"customerName\":\"John Smith\", \r\n
    \"billingAddress\":\"1234 ABC Street\", \r\n  \"billingCity\":\"Brentwood\", \r\n  \"billingState\":\"CA\", \r\n
    \"billingZip\":\"96431\", \r\n  \"shippingAddress\":\"1234 ABC Street\", \r\n
    \"shippingCity\":\"Brentwood\", \r\n  \"shippingState\":\"CA\", \r\n  \"shippingZip\":\"96431\", \r\n
    \"phoneNumber\":\"8976545468895\", \r\n  \"eMailAddress\":\"JSmith@gmail.com\", \r\n
    \"paymentDataInput\":\"Mail\", \r\n  \"customerCurrency\":\"840\", \r\n  \"contractNbr\":[ \r\n\r\n  {
    \r\n    \"status\":\"A\", \r\n    \"billingPlan\":\"Personal Trainer\", \r\n    \"startDate\":\"2018-07-
    20\", \r\n    \"endDate\":\"2019-07-18\", \r\n    \r\n    \"eMailReceipt\":\"Y\", \r\n
    \"eMailSchedulePayment\":\"N\", \r\n    \"eMailContractEnd\":\"N\", \r\n
    \"eMailContractCancel\":\"Y\", \r\n    \"taxable\":\"Y\", \r\n    \"taxRate\":\"2.00\", \r\n
    \"contractDescription\":\"Trainer Fee\" \r\n  } \r\n  ] \r\n}\r\n");
4.  Request request = new Request.Builder()
5.  .url("https://api.edgepay-uat.com/recurring/customer/add")
6.  .post(body)
7.  .addHeader("Content-Type", "application/json")
    .addHeader("merchantKey",
    "67168CA2B96231F4BFF26E38EFFBA2FD4001DFCBABA8EF")
8.  .addHeader("externalReferenceID", "1234561532106278")
9.  .addHeader("transactionDate", "2018-07-19T02:00:00")
10. .addHeader("Cache-Control", "no-cache")
11. .build();
12.
13. Response response = client.newCall(request).execute();
```

Here is a Python Example

```

import http.client conn = "{\r\n  \"merchantID\": \"6320340123456903\r\n\r\n  \"terminalID\": \"88800000282601\", \r\n  \"actionCode\": \"A\", \r\n  \"cardNumber\": \"4761739001010119\", \r\n  \"cardExpirationDate\": \"0622\", \r\n  \"customerNumber\": \"8000465765646\", \r\n  \"customerName\": \"James Morgan\", \r\n  \"billingAddress\": \"25, Spilburg lane\", \r\n  \"billingCity\": \"California\", \r\n  \"billingState\": \"AL\", \r\n  \"billingZip\": \"56431\", \r\n  \"shippingAddress\": \"Miller street\", \r\n  \"shippingCity\": \"New York\", \r\n  \"shippingState\": \"AR\", \r\n  \"shippingZip\": \"65401\", \r\n  \"phoneNumber\": \"897654546654\", \r\n  \"eMailAddress\": \"james@gmail.com\", \r\n  \"paymentDataInput\": \"Mail\", \r\n  \"customerCurrency\": \"840\", \r\n  \"contractNbr\": [\r\n    {\r\n      \"status\": \"A\", \r\n      \"billingPlan\": \"Personal Trainer\", \r\n      \"startDate\": \"2018-07-20\", \r\n      \"endDate\": \"2019-07-18\", \r\n      \"eMailReceipt\": \"Y\", \r\n      \"eMailSchedulePayment\": \"N\", \r\n      \"eMailContractEnd\": \"N\", \r\n      \"eMailContractCancel\": \"Y\", \r\n      \"taxable\": \"Y\", \r\n      \"taxRate\": \"2.00\", \r\n      \"contractDescription\": \"Trainer Fee\" \r\n    } \r\n  ] \r\n}\r\n\r\n"
1. headers = {
2.   'merchantKey': '67168CA2B96231F4BFF26E38EFFBA2FD4001DFCBABA8EF',
3.   'externalReferenceID': '1234561528754878',
4.   'transactionDate': '2018-06-11T02:00:00',
5.   'Cache-Control': 'no-cache',
6. }
7. conn.request("POST", "payment", payload, headers) res = conn.getresponse() data = res.read() print(data.decode("utf-8"))

```

9.5.2.1 Customer Add Response:

The EdgePay API Response Header has no relevant information, so you can bypass the header and go to the body:

Header:

Connection →keep-alive
 Content-Encoding →gzip
 Content-Type →application/json;charset=UTF-8
 Date →Mon, 11 Jun 2018 18:43:10 GMT
 Transfer-Encoding →chunked
 Vary →Accept-Encoding
 X-Application-Context →switch-gateway:443

JSON Body:

```

{
  "result": "A",
  "responseCode": "A0000",
  "responseMessage": "Successful",
  "merchantID": "6320340123456903",
  "terminalID": "88800000282601",
  "timestamp": "2018-07-18 20:40:07.389",
  "currencyCode": "840",
  "maskedAccountNumber": "476173*****0119",
  "cardExpirationDate": "0622",
  "tokenID": "4480956803450469",
  "customerNumber": "8000465765648"
}

```

More samples of the API can be found in the Appendix.

10 **EDGEPAY FRAUD MANAGEMENT SERVICE**

The EdgePay Fraud Management Service is an independent service that is enabled at the Merchant level. The system utilizes Static, Dynamic, and Machine Learning capabilities to identify and tag suspicious transactions as they occur.

While focused on Digital Fraud, the service also works for Card Present transactions. The Multi-Dimensional approach to Fraud management allows you to tailor your Fraud management services to meet your business needs without any IT work.

Our goal is to maximize revenue. This is achieved by squeezing out the bad guys while saying yes to the good guys, every time. Our approach does not remove you from the decision. Rather it allows you to set the rules, filters, and processes and update them as you need to while providing you the tools to look at those suspicious transactions and make informed decisions.

Within FMS, you set the rules. Along with the “Entities” which are facts you define to identify fraud, you set to either Decline or Review any transaction that is tagged by the fact. If you set it to Decline, the service will decline transaction even before we send out to the Card Brands. But, if you set to Review, then any transaction tagged with this fact will be displayed in the Review bin for you review and decision. NOTE: Review does not stop the payment but, you can easily reverse the transaction in this view if you decide not to deliver the product or service.

For integrated solutions we supply the Security Result from FMS in the Payment Response if you elect to use FMS. This can be used to denote Review and Decline transactions on your host system as well. Please refer to section 3.4.1 for Payment Response and the Security Result.

GET provides Security analysts to assist you in the initial setup of Fraud Management Services.

11 CERTIFICATION

11.1 TESTING AND CERTIFICATION INFORMATION

Certification Credentials

Contact Developer Support to get Certification Credentials

| |
|--|
| developersupport@gettrx.com |
| |
| |
| |
| |

Test Card Numbers

| Card Type | Card Number | Exp Date | CVV | Street AVS | Zip AVS |
|-----------------------------------|------------------|----------|------|---------------|------------|
| Visa | 4012000098765439 | 12/20 | 999 | 8320 | 85284 |
| Mastercard | 5499740000000057 | 12/20 | 998 | 8320 | 85284 |
| Mastercard 2 nd Bin | 2223000048400011 | 12/25 | 998 | 8320 | 85284 |
| American Express | 3714496353920376 | 12/20 | 9997 | 8320 | 85284 |
| Discover | 3055155515160018 | 12/20 | 996 | 8320 | 85284 |
| JCB | 3530142019945859 | 1220 | 996 | 8320 | 85284 |

11.2 TEST TRANSACTIONS

Once ready to certify please use the following test scripts to verify specific transaction behaviour's.

| Input type | Amount | AVS | CVV | Result | Result Code |
|------------|--------|-------|-----|------------------|-------------|
| Sale | 1.00 | | | Approved | A |
| Sale | 2.22 | | | Decline | D |
| Sale | 35.35 | | | Partial Approval | A |
| Refund | 5.55 | | | Decline | D |
| Sale | 7.77 | | | Error | R |
| Sale | 8.88 | | | Call | C |
| Sale | 1.11 | 11111 | | AVS Mismatch | A |
| Sale | 6.66 | | 111 | CVV Mismatch | A |

12 APPENDIX

12.1 SAMPLE TRANSACTIONS

Sale -Swiped Transaction

POST /payment HTTP/1.1
Host: api.edgepay-uat.com
Content-Type: application/json
merchantKey: E3DFE3D0150EB516F4E351CE7FE5A2FB799B320BDC5D523B
externalReferenceID: 1234561527119337
transactionDate: 2018-05-23T02:00:00

```
{
  "captureToo": "Yes",
  "merchantID": "6320340123456900",
  "terminalID": "88800000282601",
  "paymentDataInput": "SWIPE",
  "amount": "11.10",
  "currencyCode": "840",
  "trackData": ";4012000098765439=20121011796251900000?"
}

{
  "result": "A",
  "responseCode": "A0002",
  "responseMessage": "Partial Amount Approved",
  "merchantID": "6320340123456900",
  "terminalID": "88800000282601",
  "timestamp": "2018-05-23-23-48-12",
  "authCode": "TAS701",
  "processor": "TSYS",
  "amount": "11.10",
  "currencyCode": "840",
  "transactionID": "CS47745840597246312",
  "avsResponse": "Address verification was not requested.",
  "commercialCard": "n"
}
```

Sale – Manual Transaction

POST /payment HTTP/1.1
Host: api.edgepay-uat.com
Content-Type: application/json
merchantKey: E3DFE3D0150EB516F4E351CE7FE5A2FB799B320BDC5D523B
externalReferenceID: 1234561527119391
transactionDate: 2018-05-23T02:00:00

```
{
  "captureToo": "Yes",
  "merchantID": "6320340123456900",
  "terminalID": "88800000282601",
  "paymentDataInput": "Manualentry",
  "amount": "101.00",
```

```

    "currencyCode": "840",
    "cardNumber": "4012000098765439",
    "cardExpirationDate": "1220",
    "billingZip": "85284"
  }

  {
    "result": "A",
    "responseCode": "A0000",
    "responseMessage": "Success",
    "merchantID": "6320340123456900",
    "terminalID": "88800000282601",
    "timestamp": "2018-05-23-21-40-46",
    "authCode": "TAS889",
    "processor": "TSYS",
    "amount": "101.00",
    "currencyCode": "840",
    "maskedAccount": "401200*****5439",
    "cardExpirationDate": "1220",
    "transactionID": "CS13730432886403704",
    "avsResponse": "Neither the street address or ZIP code match what is on file.",
    "commercialCard": "n"
  }

```

Sale – Transaction with Recurring Installments (Total Amount)

POST /recurring/customer/add HTTP/1.1
 Host: edgepayapi.com
 Content-Type: application/json
 merchantKey: F2D0B30DE03831236D51156E0170D6B576514FC2C86D94E0
 externalReferenceID: 1234561570454900
 transactionDate: 2019-10-04T00:00:01

```

  {
    "captureToo": "Yes",
    "merchantID": "6320340123456900",
    "terminalID": "88800000282601",
    "paymentDataInput": "INTERNET",
    "billingAddress": "Address",
    "billingCity": "City",
    "billingState": "CA",
    "billingZip": "85284",
    "shippingAddress": "NewYork",
    "shippingCity": "NewYork",
    "shippingState": "AR",
    "shippingZip": "65401",
    "amount": "10.01",
    "currencyCode": "840",
    "cardNumber": "4716308261910750",
    "cardExpirationDate": "1122",
    "processor": "TSYS",
    "customerName": "Installment-092319",
    "customerNumber": "10042019",
    "phoneNumber": "9038000439",
    "eMailAddress": "james@gmail.com",
    "url": "http://www.edgepay.com",
  }

```

```

        "memberUsername": "Username1",
        "memberPassword": "Password1234",
        "affiliate": "12365ID",
        "makeRecurring": "Yes",
        "status": "A",
        "merchantField1": "Field1",
        "merchantField2": "Field2",
        "recurringPaymentRequest":
    [
    {
        "schedulePaymentType": "Installment",
        "totalAmount": "40.91",
        "frequency": "4",
        "paidIn": "4",
        "startDate": "2019-10-06",
        "description": "Installment 10042019",
        "emailReceipt": "Y"
    }
    ]
}

{
    "result": "A",
    "responseCode": "A0000",
    "responseMessage": "Successful",
    "merchantID": "6320340123456900",
    "terminalID": "88800000282601",
    "timestamp": "2019-10-04 17:42:29.143",
    "maskedAccount": "401200*****5439",
    "cardExpirationDate": "1220",
    "amount": "10.01",
    "tokenID": "5424391763607600",
    "customerNumber": "10042019",
    "securityResult": "A",
    "securityMessage": "APPROVED",
    "processor": "TSYS",
    "authCode": "TAS625",
    "avsResponse": "Zip Match(Z) : The nine character ZIP code submitted matches what is on file.",
    "commercialCard": "y",
    "avsCode": "Z",
    "transactionId": "CS70565957796520677",
    "recurringIDs": [
        "9201691219991561"
    ],
}

```

Sale – Transaction with Recurring Installments (Recurring Amount)

POST /recurring/customer/add HTTP/1.1
 Host: edgepayapi.com
 Content-Type: application/json
 merchantKey: F2D0B30DE03831236D51156E0170D6B576514FC2C86D94E0
 externalReferenceID: 1234561570454900
 transactionDate: 2019-10-04T00:00:01

```

{
    "captureToo": "Yes",

```

```

"merchantID":"6320340123456900",
"terminalID":"88800000282601",
"paymentDataInput":"INTERNET",
"billingAddress":"Address",
"billingCity":"City",
"billingState":"CA",
"billingZip":"85284",
"shippingAddress":"NewYork",
"shippingCity":"NewYork",
"shippingState":"AR",
"shippingZip":"65401",
"amount":"10.00",
"currencyCode":"840",
"cardNumber":"4111111111111111",
"cardExpirationDate":"1122",
"processor":"TSYS",
"customerName":"Recurring Installment",
"customerNumber":"903800043952",
"phoneNumber":"9038000439",
"eMailAddress":"test@gettrx.com",
"url":"http://www.edgepay.com",
"memberUsername":"Username1",
"memberPassword":"Password1234",
"affiliate":"12365ID",
"makeRecurring":"Yes",
"status":"A",
"recurringPaymentRequest":
[
{
"schedulePaymentType":"Installment",
"recurringAmount":"43.91",
"frequency":"4",
"paidIn":"4",
"startDate":"2019-10-05",
"description":"Recurring Installment",
"emailReceipt":"Y"
}
]
}

{
"result": "A",
"responseCode": "A0000",
"responseMessage": "Successful",
"merchantID":"6320340123456900",
"terminalID":"88800000282601",
"timestamp": "2019-10-04 20:15:38.276",
"maskedAccountNumber": "411111*****1111",
"cardExpirationDate": "1122",
"tokenID": "4619016931237921",
"customerNumber": "903800043952",
"securityResult": "A",
"securityMessage": "APPROVED",
"processor": "TSYS",
"authCode": "TAS750",
"avsResponse": "Zip Match(Z) : The nine character ZIP code submitted matches
what is on file.",
"commercialCard": "y",
"avsCode": "Z",

```

```

        "transactionId": "CS77084354804935084",
        "recurringIDs": [
            "1400822326746785"
        ],
        "amount": "10.00"
    }

```

Sale – Transaction with Recurring Subscription (Recurring Amount)

POST /recurring/customer/add HTTP/1.1

Host: edgepayapi.com

Content-Type: application/json

merchantKey: F2D0B30DE03831236D51156E0170D6B576514FC2C86D94E0

externalReferenceID: 1234561570454900

transactionDate: 2019-10-04T00:00:01

```

{
    "captureToo": "Yes",
    "merchantID": "6320340123456900",
    "terminalID": "88800000282601",
    "paymentDataInput": "INTERNET",
    "billingAddress": "Address",
    "billingCity": "City",
    "billingState": "CA",
    "billingZip": "85284",
    "shippingAddress": "NewYork",
    "shippingCity": "NewYork",
    "shippingState": "AR",
    "shippingZip": "65401",
    "amount": "13.03",
    "currencyCode": "840",
    "cardNumber": "5499740000000057",
    "cardExpirationDate": "1122",
    "processor": "TSYS",
    "customerName": "Recurring Subscription",
    "customerNumber": "903800043953",
    "phoneNumber": "9038000439",
    "eMailAddress": "wwatley@gettrx.com",
    "url": "http://www.edgepay.com",
    "memberUsername": "Username1",
    "memberPassword": "Password1234",
    "affiliate": "12365ID",
    "makeRecurring": "Yes",
    "status": "A",
    "recurringPaymentRequest":
    [
        {
            "schedulePaymentType": "Subscription",
            "recurringAmount": "43.91",
            "frequency": "4",
            "startDate": "2019-10-20",
            "description": "Recurring Subscription",
            "emailReceipt": "Y"
        }
    ]
}

```

```

}
]

{
  "result": "A",
  "responseCode": "A0000",
  "responseMessage": "Successful",
  "merchantID": "6320340123456900",
  "terminalID": "88800000282601",
  "timestamp": "2019-10-04 20:19:54.03",
  "maskedAccountNumber": "549974*****0057",
  "cardExpirationDate": "1122",
  "tokenID": "4134804882045891",
  "customerNumber": "903800043953",
  "securityResult": "A",
  "securityMessage": "APPROVED",
  "processor": "TSYS",
  "authCode": "VTLMC1",
  "avsResponse": "Exact Match(Y) : The five character ZIP code submitted matches
what is on file.",
  "commercialCard": "y",
  "avsCode": "Y",
  "transactionId": "CS91278938575504158",
  "recurringIDs": [
    "2359256122420744"
  ],
  "amount": "13.03"
}

```

Sale (Token) – Transaction with Recurring Installments (Recurring Amount)

POST /recurring/customer/add HTTP/1.1
 Host: edgepayapi.com
 Content-Type: application/json
 merchantKey: F2D0B30DE03831236D51156E0170D6B576514FC2C86D94E0
 externalReferenceID: 1234561570454900
 transactionDate: 2019-10-04T00:00:01

```

{
  "captureToo": "Yes",
  "merchantID": "6320340123456900",
  "terminalID": "88800000282601",
  "paymentDataInput": "INTERNET",
  "billingAddress": "Addresstwo",
  "billingCity": "citytwo",
  "billingState": "CA",
  "billingZip": "85284",
  "shippingAddress": "city",
  "shippingCity": "NewYorktwo",
  "shippingState": "AR",
  "shippingZip": "65401",
  "amount": "11.99",
  "currencyCode": "840",
  "tokenID": "4619016931237921",
  "processor": "TSYS",
  "customerName": "sb five",
  "customerNumber": "903800043955",

```

```

"phoneNumber":"9038000439",
"eMailAddress":"james@gmail.com",
"url":"http://www.edgepaycom",
"memberUsername":"Username1",
"memberPassword":"Password1234",
"affiliate":"12365ID",
"makeRecurring":"Yes",
"status":"A",
"recurringPaymentRequest":
[
{
"schedulePaymentType":"Installment",
"recurringAmount":"11.12",
"frequency":"M",
"startDate":"2019-10-20",
"paidIn":"3",
"description":"Token-BillingPlan03",
"emailReceipt":"Y"
}

{
"result": "A",
"responseCode": "A0000",
"responseMessage": "Success",
"merchantID": "6320340123456900",
"terminalID": "88800000282601",
"timestamp": "2019-10-04-13-40-20",
"securityResult": "A",
"securityMessage": "APPROVED",
"authCode": "TAS009",
"processor": "TSYS",
"amount": "11.99",
"currencyCode": "840",
"maskedAccount": "411111*****1111",
"cardExpirationDate": "1122",
"tokenID": "4619016931237921",
"customerNumber": "903800043955",
"transactionID": "CS11264563162462083",
"avsResponse": "Zip Match(Z) : The nine character ZIP code submitted matches
what is on file.",
"commercialCard": "n",
"avsCode": "Z"
}
}

```

Sale (Token) – Transaction with Recurring Installments (Total Amount)

POST /recurring/customer/add HTTP/1.1
 Host: edgepayapi.com
 Content-Type: application/json
 merchantKey: F2D0B30DE03831236D51156E0170D6B576514FC2C86D94E0
 externalReferenceID: 1234561570454900
 transactionDate: 2019-10-04T00:00:01

```

{
"captureToo":"Yes",
"merchantID":"6320340123456900",
"terminalID":"88800000282601",
"paymentDataInput":"INTERNET",

```

```

"billingAddress": "Address two",
"billingCity": "city two",
"billingState": "CA",
"billingZip": "85284",
"shippingAddress": "city",
"shippingCity": "New York two",
"shippingState": "AR",
"shippingZip": "65401",
"amount": "11.99",
"currencyCode": "840",
"tokenID": "5970803989190028",
"processor": "TSYS",
"customerName": "SB four",
"customerNumber": "903800043",
"phoneNumber": "9038000439",
"emailAddress": "james@gmail.com",
"url": "http://www.edgepay.com",
"memberUsername": "Username1",
"memberPassword": "Password1234",
"affiliate": "12365ID",
"makeRecurring": "Yes",
"status": "A",
"recurringPaymentRequest":

[
{
"schedulePaymentType": "Installment",
"totalAmount": "11.12",
"frequency": "M",
"startDate": "2019-10-05",
"paidIn": "3",
"description": "Token-BillingPlan03",
"emailReceipt": "Y"
}
]

{
"result": "A",
"responseCode": "A0000",
"responseMessage": "Successful",
"merchantID": "6320340123456900",
"terminalID": "88800000282601",
"timestamp": "2019-10-04 20:58:52.39",
"maskedAccountNumber": "411111*****1111",
"cardExpirationDate": "0621",
"tokenID": "5970803989190028",
"customerNumber": "903800043",
"securityResult": "A",
"securityMessage": "APPROVED",
"processor": "TSYS",
"authCode": "TAS182",
"avsResponse": "Zip Match(Z) : The nine character ZIP code submitted matches
what is on file.",
"commercialCard": "y",
"avsCode": "Z",
"transactionId": "CS62157876691734660",
"recurringIDs": [
"2906167474020226"
],

```

```
        "amount": "11.99"
    }
}
```

Sale (Token) – Transaction with Recurring Subscription (Recurring Amount)

POST /recurring/customer/add HTTP/1.1

Host: edgepayapi.com

Content-Type: application/json

merchantKey: F2D0B30DE03831236D51156E0170D6B576514FC2C86D94E0

externalReferenceID: 1234561570454900

transactionDate: 2019-10-04T00:00:01

```
{
    "captureToo": "Yes",
    "merchantID": "888000002826",
    "terminalID": "88800000282601",
    "paymentDataInput": "INTERNET",
    "billingAddress": "Addresstwo",
    "billingCity": "citytwo",
    "billingState": "CA",
    "billingZip": "85284",
    "shippingAddress": "city",
    "shippingCity": "NewYorktwo",
    "shippingState": "AR",
    "shippingZip": "65401",
    "amount": "11.99",
    "currencyCode": "840",
    "tokenID": "5970803989190028",
    "processor": "TSYS",
    "customerName": "SB six",
    "customerNumber": "903800043956",
    "phoneNumber": "9038000439",
    "eMailAddress": "james@gmail.com",
    "url": "http://www.edgepaycom",
    "memberUsername": "Username1",
    "memberPassword": "Password1234",
    "affiliate": "12365ID",
    "makeRecurring": "Yes",
    "status": "A",
    "recurringPaymentRequest":
    [
        {
            "schedulePaymentType": "Subscription",
            "recurringAmount": "11.12",
            "frequency": "M",
            "startDate": "2019-10-20",

            "description": "Token-BillingPlan03",
            "emailReceipt": "Y"
        }
    ]
}
```

```
{
    "result": "A",
}
```

```

"responseCode": "A0000",
"responseMessage": "Success",
"merchantID": "888000002826",
"terminalID": "88800000282601",
"timestamp": "2019-10-04-14-03-39",
"securityResult": "A",
"securityMessage": "APPROVED",
"authCode": "TAS212",
"processor": "TSYS",
"amount": "11.99",
"currencyCode": "840",
"maskedAccount": "411111*****1111",
"cardExpirationDate": "0621",
"tokenID": "5970803989190028",
"customerNumber": "903800043956",
"transactionID": "CS63389217481298380",
"avsResponse": "Zip Match(Z) : The nine character ZIP code submitted matches what is on
file.",
"commercialCard": "n",
"avsCode": "Z"
}

```

Refund Transaction

POST /refund HTTP/1.1
 Host: api.edgepay-uat.com
 Content-Type: application/json
 merchantKey: E3DFE3D0150EB516F4E351CE7FE5A2FB799B320BDC5D523B
 externalReferenceID: 1234561527119566
 transactionDate: 2018-05-23T02:00:00

```

{
  "merchantID": "6320340123456900",
  "terminalID": "88800000282601",
  "transactionID": "CS75302302660447689"
}

{
  "result": "A",
  "responseCode": "A0000",
  "responseMessage": "Success",
  "merchantID": "6320340123456900",
  "terminalID": "88800000282601",
  "timestamp": "2018-05-23-23-53-45",
  "processor": "TSYS",
  "transactionID": "CS11876175370499715",
  "origTransactionID": "CS50463158851818869"
}

```

Refund Any Card Manual Transaction

POST /refund HTTP/1.1
 Host: api.edgepay-uat.com
 Content-Type: application/json
 merchantKey: E3DFE3D0150EB516F4E351CE7FE5A2FB799B320BDC5D523B
 externalReferenceID: 1234561532110897
 transactionDate: 2018-07-20T02:00:00

```
{
  "merchantID":"6320340123456900",
  "terminalID":"88800000282601",
  "amount":"2.11",
  "cardNumber":"4012000098765439",
  "cardExpirationDate":"1220",
  "paymentDataInput":"MANUALEENTRY"
}

{
  "result": "A",
  "responseCode": "A0000",
  "responseMessage": "Success",
  "merchantID": "6320340123456900",
  "terminalID": "88800000282601",
  "timestamp": "2018-07-20-18-01-17",
  "processor": "TSYS",
  "transactionID": "CS70910267410995714"
}
```

Refund Any Card Swipe Transaction

POST /refund HTTP/1.1
 Host: api.edgepay-uat.com
 Content-Type: application/json
 merchantKey: E3DFE3D0150EB516F4E351CE7FE5A2FB799B320BDC5D523B
 externalReferenceID: 1234561532111002
 transactionDate: 2018-07-20T02:00:00

```
{
  "merchantID":"6320340123456900",
  "terminalID":"88800000282601",
  "paymentDataInput":"SWIPE",
  "amount":"3.00",
  "trackData".";4012000098765439=20121011796251900000?"
}

{
  "result": "A",
  "responseCode": "A0000",
  "responseMessage": "Success",
  "merchantID": "6320340123456900",
  "terminalID": "88800000282601",
  "timestamp": "2018-07-20-18-08-59",
  "processor": "TSYS",
  "transactionID": "CS79546938710933612"
}
```

Refund Any Card (Token) Manual Transaction

POST /refund HTTP/1.1
 Host: api.edgepay-uat.com
 Content-Type: application/json
 merchantKey: E3DFE3D0150EB516F4E351CE7FE5A2FB799B320BDC5D523B
 externalReferenceID: 1234561532110897
 transactionDate: 2018-07-20T02:00:00

```
{
  "merchantID":"6320340123456900",
  "terminalID":"88800000282601",
  "amount":"2.11",
  "tokenID": "4784294325501111",
  "paymentDataInput":"MANUALENTY"
}

{
  "result": "A",
  "responseCode": "A0000",
  "responseMessage": "Success",
  "merchantID": "6320340123456900",
  "terminalID": "88800000282601",
  "timestamp": "2018-07-20-18-01-17",
  "processor": "TSYS",
  "transactionID": "CS70910267410995714"
}
```

Token Transaction

POST /token HTTP/1.1
 Host: api.edgepay-uat.com
 Content-Type: application/json
 merchantKey: E3DFE3D0150EB516F4E351CE7FE5A2FB799B320BDC5D523B
 externalReferenceID: 1234561527119506

```
{
  "merchantID":"6320340123456900",
  "terminalID":"88800000282601",
  "cardNumber":"6011000993026909",
  "cardExpirationDate":"1221"
}

{
  "result": "A",
  "responseCode": "EP0081",
  "responseMessage": "Successful",
  "merchantID": "6320340123456900",
  "tokenID": "4460741062478699",
  "cardExpirationDate": "1221"
}
```

Recurring – Add Customer and Contract Token

POST /recurring/customer/add HTTP/1.1
 Host: api.edgepay-uat.com
 Content-Type: application/json
 merchantKey: 67168CA2B96231F4BFF26E38EFFBA2FD4001DFCBABA8EF
 externalReferenceID: 1234561532111259
 transactionDate: 2018-07-20T02:00:00

```
{
  "merchantID":"6320340123456903",
  "terminalID":"88800000282601",
  "actionCode":"A",
  "tokenID": "4784294325501111",

```

```

"customerNumber":"8000465765647",
"customerName":"Jane Morgan",
"billingAddress":"25, Spilburg lane",
"billingCity":"California",
"billingState":"AL",
"billingZip":"56431",
"shippingAddress":"Miller street",
"shippingCity":"New York",
"shippingState":"AR",
"shippingZip":"65401",
"phoneNumber":"897654546654",
"eMailAddress":"jan@gmail.com",
"paymentDataInput":"Mail",
"customerCurrency":"840",

"contractNbr":[
  {
    "status":"A",
    "billingPlan":"Gym Plan",
    "startDate":"2018-09-18",
    "endDate":"2019-07-18",
    "eMailReceipt":"Y",
    "eMailSchedulePayment":"N",
    "eMailContractEnd":"Y",
    "eMailContractCancel":"Y",
    "taxable":"Y",
    "taxRate":"2.0",

    "contractDescription":"Monthly Subscription"
  },
]

{
  "result": "A",
  "responseCode": "A0000",
  "responseMessage": "Successful",
  "merchantID": "6320340123456903",
  "terminalID": "88800000282601",
  "timestamp": "2018-07-20 18:30:59.59",
  "currencyCode": "840",
  "maskedAccountNumber": "476173*****0119",
  "cardExpirationDate": "1220",
  "tokenID": "4784294325501111",
  "customerNumber": "8000465765649"
}

```

Recurring - Customer and Contract Update Card

```

POST /recurring/customer/add HTTP/1.1
Host: api.edgepay-uat.com
Content-Type: application/json
merchantKey: 67168CA2B96231F4BFF26E38EFFBA2FD4001DFCBABA8EF
externalReferenceID: 1234561532111929
transactionDate: 2018-07-20T02:00:00

```

```

{
  "merchantID":"6320340123456903",
  "terminalID":"88800000282601",
  "actionCode":"C",
  "cardNumber":"4761739001010119",

```

```

"cardExpirationDate": "0626",
"customerNumber": "8000465765646",
"customerName": "James Morgan",
"billingAddress": "28, Spilburg lane",
"billingCity": "California",
"billingState": "AL",
"billingZip": "56431",
"shippingAddress": "Miller street",
"shippingCity": "New York",
"shippingState": "AR",
"shippingZip": "65401",
"phoneNumber": "897654546654",
"emailAddress": "jamesm@gmail.com",
"paymentDataInput": "Mail",
"customerCurrency": "840",

"contractNbr": [
  {
    "contractNumber": "482761723871089",
    "status": "S",
    "billingPlan": "Personal Trainer",
    "startDate": "2018-07-20",
    "endDate": "2019-07-18",
    "emailReceipt": "Y",
    "emailSchedulePayment": "N",
    "emailContractEnd": "Y",
    "emailContractCancel": "Y",
    "taxable": "Y",
    "taxRate": "4.0",
    "contractDescription": "Trainer Fee"
  }
]
}

{
  "result": "A",
  "responseCode": "A0000",
  "responseMessage": "Successful",
  "merchantID": "6320340123456903",
  "terminalID": "88800000282601",
  "timestamp": "2018-07-20 18:38:01.023",
  "currencyCode": "840",
  "maskedAccountNumber": "476173*****0119",
  "cardExpirationDate": "0626",
  "tokenID": "4480956803450469",
  "customerNumber": "8000465765646"
}

```

Recurring - Customer and Contract Update -Token

POST /recurring/customer/add HTTP/1.1
 Host: api.edgepay-uat.com
 Content-Type: application/json
 merchantKey: 67168CA2B96231F4BFF26E38EFFBA2FD4001DFCBABA8EF
 externalReferenceID: 1234561532112415
 transactionDate: 2018-07-20T02:00:00

```

{
  "merchantID": "6320340123456903",

```

```

"terminalID": "88800000282601",
"actionCode": "C",
"tokenID": "4784294325501111",
"customerNumber": "8000465765649",
"customerName": "Bill Jones",
"billingAddress": "25, Iceburg lane",
"billingCity": "California",
"billingState": "CA",
"billingZip": "93035",
"shippingAddress": "Miller street",
"shippingCity": "New York",
"shippingState": "AR",
"shippingZip": "65401",
"phoneNumber": "897654546654",
"emailAddress": "jan@gmail.com",
"paymentDataInput": "Mail",
"customerCurrency": "840",

"contractNbr": [
  {
    "contractNumber": "750861822875329",
    "status": "A",
    "billingPlan": "Personal Trainer",
    "startDate": "2018-07-20",
    "endDate": "2019-07-18",
    "emailReceipt": "Y",
    "emailSchedulePayment": "N",
    "emailContractEnd": "Y",
    "emailContractCancel": "Y",
    "taxable": "Y",
    "taxRate": "4.0",
    "contractDescription": "Trainer Fee"
  }
]
}

{
  "result": "A",
  "responseCode": "A0000",
  "responseMessage": "Successful",
  "merchantID": "6320340123456903",
  "terminalID": "88800000282601",
  "timestamp": "2018-07-20 18:46:05.07",
  "currencyCode": "840",
  "maskedAccountNumber": "476173*****0119",
  "cardExpirationDate": "1220",
  "tokenID": "4784294325501111",
  "customerNumber": "8000465765649"
}

```

PayFac Sale -Swiped Transaction

```

POST /payment HTTP/1.1
Host: api.edgepay-uat.com
Content-Type: application/json
merchantKey: E3DFE3D0150EB516F4E351CE7FE5A2FB799B320BDC5D523B
externalReferenceID: 1234561527119337
transactionDate: 2018-05-23T02:00:00

```

```
{
  "captureToo": "Yes",
  "merchantID": "6320340123456900",
  "terminalID": "88800000282601",
  "paymentDataInput": "SWIPE",
  "amount": "11.10",
  "currencyCode": "840",
  "trackData": ";4012000098765439=20121011796251900000?",

  "payFac": "Yes",
  "sID": "222212345678",
  "acctSplits": [

    {"accountID": 1, "amount": 1.00},
    {"accountID": 1, "amount": 1.00}

  ]
}
```

```
{
  "result": "A",
  "responseCode": "A0002",
  "responseMessage": "Partial Amount Approved",
  "merchantID": "6320340123456900",
  "terminalID": "88800000282601",
  "timestamp": "2018-05-23-23-48-12",
  "authCode": "TAS701",
  "processor": "TSYS",
  "amount": "11.10",
  "currencyCode": "840",
  "transactionID": "CS47745840597246312",
  "avsResponse": "Address verification was not requested.",
  "commercialCard": "n"
}
```

PayFac Sale – Manual Transaction

POST /payment HTTP/1.1
 Host: api.edgepay-uat.com
 Content-Type: application/json
 merchantKey: E3DFE3D0150EB516F4E351CE7FE5A2FB799B320BDC5D523B
 externalReferenceID: 1234561527119391
 transactionDate: 2018-05-23T02:00:00

```
{
  "captureToo": "Yes",
  "merchantID": "6320340123456900",
  "terminalID": "88800000282601",
  "paymentDataInput": "Manualentry",
  "amount": "101.00",
  "currencyCode": "840",
  "cardNumber": "4012000098765439",
  "cardExpirationDate": "1220",
  "billingZip": "85284",

  "payFac": "Yes",
  "sID": "222212345678",
```

```

"acctSplits":[

{"accountID": 1, "amount": 1.00},
{"accountID": 1, "amount": 1.00}

]
}

{
  "result": "A",
  "responseCode": "A0000",
  "responseMessage": "Success",
  "merchantID": "6320340123456900",
  "terminalID": "88800000282601",
  "timestamp": "2018-05-23-21-40-46",
  "authCode": "TAS889",
  "processor": "TSYS",
  "amount": "101.00",
  "currencyCode": "840",
  "maskedAccount": "401200*****5439",
  "cardExpirationDate": "1220",
  "transactionID": "CS13730432886403704",
  "avsResponse": "Neither the street address or ZIP code match what is on file.",
  "commercialCard": "n"
}

```

Auth Only Request Packet (PayFac)

POST /payment HTTP/1.1
 Host: api.edgepay-uat.com
 Content-Type: application/json
 merchantKey: E3DFE3D0150EB516F4E351CE7FE5A2FB799B320BDC5D523B
 externalReferenceID: 1234561527119391
 transactionDate: 2018-05-23T02:00:00

```

{
  "merchantID":"88000002826",
  "terminalID":"88800000282601",
  "paymentDataInput":"Manualentry",
  "amount":"2.05",
  "currencyCode":"840",
  "cardNumber":"371449635392376",
  "cardExpirationDate":"0919",
  "cvv2":"9997",
  "billingAddress":"8320",
  "billingCity":"Milpitas",
  "billingState":"CA",
  "billingZip":"85284-9999",
  "token":"yes"
  "payFac":"Yes",
    "sID": "2222",
    "acctSplits":[
      {"accountID": 1, "amount": 1.00},
      {"accountID": 1, "amount": 1.00}
    ]
}

```

Auth Only Response packet

```

{
  "result": "A",

```

```

"responseCode": "A0000",
"responseMessage": "Success",
"merchantID": "88000002826",
"terminalID": "88800000282601",
"sID": "2222",
"timestamp": "2019-03-20-08-44-28",
"securityResult": "A",
"securityMessage": "APPROVED",
"authCode": "AXS631",
"processor": "TSYS",
"amount": "2.05",
"currencyCode": "840",
"maskedAccount": "371449****2376",
"cardExpirationDate": "0919",
"tokenID": "4122589763319236",
"transactionID": "CS42347608157031738",
"avsResponse": "Zip Match(Z) : The nine character ZIP code submitted matches what is on
file.",

"cvvResponse": "CVV2 verification successful",
"avsCode": "Z"
}

```

Auth Only Request Packet for SubMerchant Direct

```

POST /payment HTTP/1.1
Host: api.edgepay-uat.com
Content-Type: application/json
merchantKey: E3DFE3D0150EB516F4E351CE7FE5A2FB799B320BDC5D523B
externalReferenceID: 1234561527119391
transactionDate: 2018-05-23T02:00:00

```

```

{
  "merchantID": "2222",
  "terminalID": "88800000282601",
  "paymentDataInput": "Manualentry",
  "amount": "2.05",
  "currencyCode": "840",
  "cardNumber": "371449635392376",
  "cardExpirationDate": "0919",
  "cvv2": "9997",
  "billingAddress": "8320",
  "billingCity": "Milpitas",
  "billingState": "CA",
  "billingZip": "85284-9999",
  "token": "yes",
  "payFac": "Yes",
  "sID": "2222",
  "acctSplits": [
    {"accountID": 1, "amount": 1.00},
    {"accountID": 1, "amount": 1.00}
  ]
}

```

Auth Only Response packet

```

{
  "result": "A",
  "responseCode": "A0000",
  "responseMessage": "Success",

```

"merchantID": "2222",
"terminalID": "88800000282601",
"sID": "2222",
"timestamp": "2019-03-20-08-44-28",
"securityResult": "A",
"securityMessage": "APPROVED",
"authCode": "AXS631",
"processor": "TSYS",
"amount": "2.05",