First Data.

FirstAPI Recurring Batch Solution

Implementation Guide

Version 2017-11a

First Data

Implementation Guide Recurring Batch Solution

Version 2017-11

Contents

Getting Support

There are different manuals available for First Data's eCommerce solutions. This Implementation Guide will be the most helpful for integrating the Recurring Batch solution for usage with our distribution channels.

If you have read the documentation and cannot find the answer to your question, please contact your local support team.

First Data

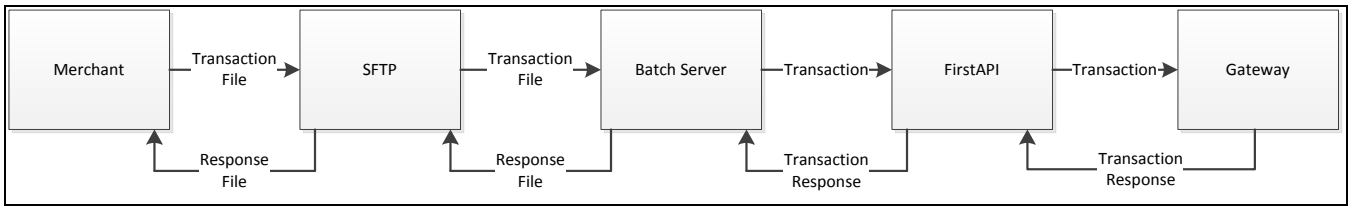
1. Overview and Supported APIs

FirstAPI offers merchants the ability to send transactions and receive responses in batch files based on the APIs defined on the FirstAPI Developer Portal. Currently, the batch transaction process supports the following APIs:

- Credit Card Payments
  - Authorize
  - Purchase
  - Recurring
- Token-Based Payments
  - Authorize
  - Purchase
  - Recurring

The batch files are .CSV files that are transferred between the merchant and FirstAPI using the First Data File Gateway (FDFG) secure transmission servers. FDFG supports multiple transmission protocols, including SFTP and HTTPS. The specific procedures in this document refer to the SFTP protocol, but others may be supported. Merchants should consult with their First Data implementation team for more details.

The general data transfer flow is shown below:



1.1 Prerequisites

Before sending and receiving files, merchants must first:

- Onboard to the appropriate gateway.
- Enroll and be certified on FirstAPI (the FirstAPI Developer Portal).
- Onboard to the FDFG and receive a FDFG Merchant ID and password. Additionally, this process will require the merchant to establish authentication information appropriate to the transmission protocol.

1.2 About This Document

The target audience for this document is a First Data client that wants to implement the transaction features provided by FirstAPI in a batch environment.

This document provides the following information:

- File naming standards
- Input and output file layouts and formats
- Inbound and outbound file transfer processes and procedures using SFTP through the First Data File Gateway

1.3 Related Documents and Resources

The following documents and resources provide supporting information for this document:

- FirstAPI Developer Portal (https://developer.payeezy.com/) - This portal is where merchants register and are boarded to FirstAPI. It also contains specifications for all supported APIs, sample requests and responses, and development guides.

The following documents are provided by the First Data File Gateway team. Merchants and developers should contact their First Data implementation team for access to these documents.

- *First Data File Gateway External Client Boarding Guide* - Includes the requirements and processes for transferring files.
- *First Data File Gateway Boarding Questionaire* - Includes the information a merchant needs to provide in order to set up an account to transfer batch files.
- *First Data File Gateway MyFileGateway Implementation Guide* – Explains how to work with the MyFileGateway user interface for HTTPS transfers.

2. File Naming Standards

For proper processing, all files must be named according to the standards described in this section.

Note: If the file is not named according to the following naming standards, it will be ignored. No errors will be sent.

2.1 Input Files

Files sent to FirstAPI (input files) must be named using the following standard:

`<SFG-Merchant-Id>.<targetSystem>.YYMMDDnn.<filename>.<extension>`

The following table describes the different components of this standard.

Component	Description
SFG-Merchant-Id	The ID assigned to the merchant when boarding to FDFG. **Note:** This is the FDFG ID, not the FirstAPI Developer Portal Merchant ID.
targetSystem	The Target System ID (3-8 characters). Currently, **FAPIT** (FirstAPI transactions) is supported.
YYMMDDnn	Date and serial number of the file upload. The serial number is used to differentiate multiple files on a single day and begins with 00.
Filename	Any string of letters that identifies the file to the merchant. The string should be limited to no more than 16 characters.
Extension	Currently, .**CSV** is the supported value.

2.2 Output Files

Files sent from First Data to the merchant in response to input files follow the same naming standard with an `'-out'` string appended to the date component.

2.3 Examples

Request file - `MERCHANT1.FAPIT.17031500.transactions.csv`
Response file - `MERCHANT1.FAPIT.17031500-out.transactions.csv`

First Data

3. Input File Format and Layout

The transaction input file is a CSV file where the first two lines describe the configuration of the file and each additional line describes a single transaction

3.1 Configuration Lines

The first two lines of the transaction input files describe identification and configuration information, as follows:

- Line 1 – Merchant Identification
- Line 2 – Fields

3.1.1 Merchant Identification Line

The first line of the transaction input file must be the Merchant Identification line. This line identifies the merchant submitting the file. The line contains the following key-value pairs, separated by commas.

Key	Description
Email	The unique email address of the developer as registered on the FirstAPI Developer Portal.
Token	The token that identifies the merchant in the developer's account on the FirstAPI Developer Portal.

An example of the Merchant Identification line is shown below:

```
email=emailname@email.com,token=fdoa-a480ce8951daa73262734cf102641994c1e55e7cdf4c02b6
```

3.1.2 Fields Line

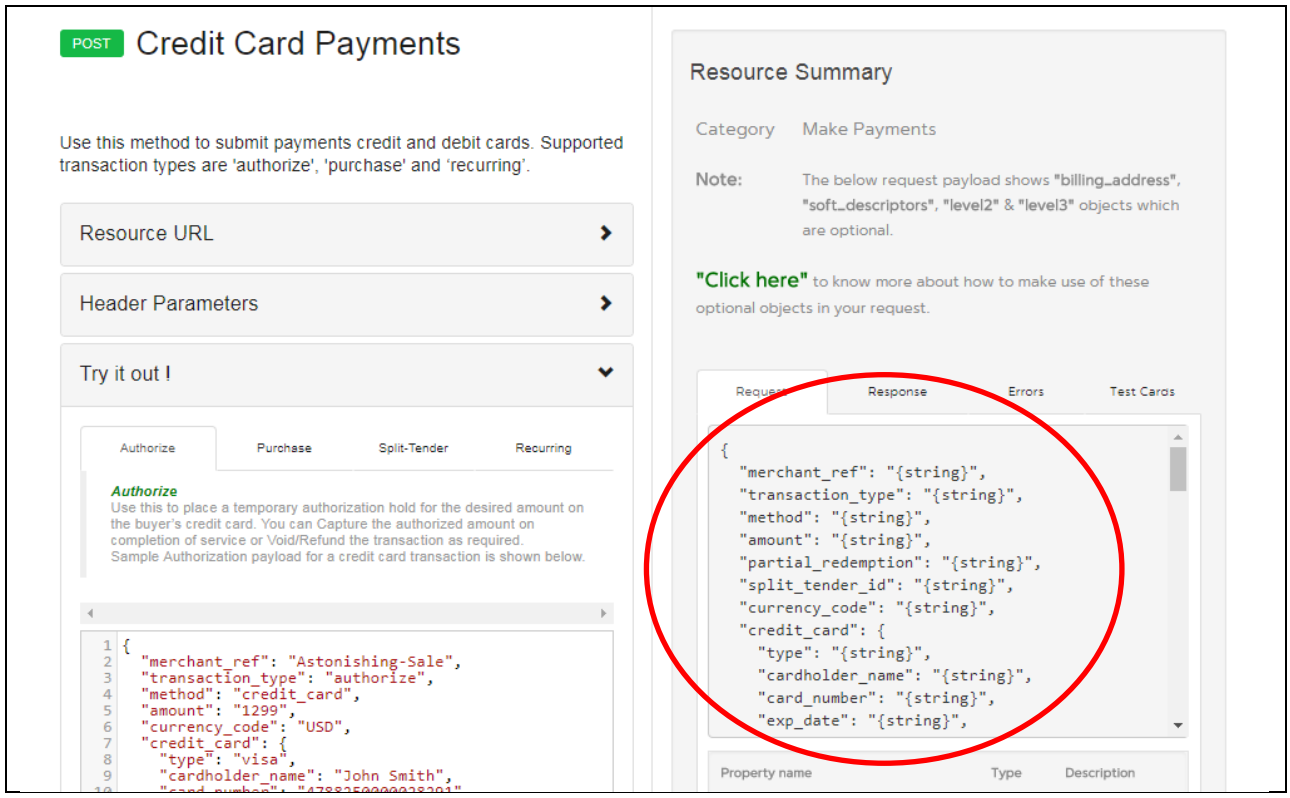
The second line of the transaction lists the fields that will be included in the transaction lines. The required format lists each field separated by a comma. The order of the fields in this line is the order that must be followed in each transaction line.

For each batch file, the appropriate fields are listed in the FirstAPI Developer Portal under the equivalent API. In the API, the fields are sent as a JSON object; in the batch file, each field must be listed on the second line of the input file.

Note: The CVV field is not supported in the batch environment. If the field is sent, it will be ignored. FirstAPI will not store the CVV field and will not include it in any responses.

3.1.2.1 API to Batch Fields Example

The following figure shows the Credit Card Payments - Authorize API on the FirstAPI Developer Portal, with the Request payload circled.



The fields in the non-optional object for this transaction are shown as:

```
{
  "merchant_ref": "{string}",
  "transaction_type": "{string}",
  "method": "{string}",
  "amount": "{string}",
  "partial_redemption": "{string}",
  "split_tender_id": "{string}",
  "currency_code": "{string}",
  "credit_card": {
    "type": "{string}",
    "cardholder_name": "{string}",
    "card_number": "{string}",
    "exp_date": "{string}",
    "cvv": "{string}"
  },
```

The corresponding Fields line for a Credit Card Payment transaction batch file is shown below:

```
merchant_ref,transaction_type,method,amount,currency_code,credit_card.type,credit_card
.cardholder_name,credit_card.card_number,credit_card.exp_date
```

Note that API objects containing multiple name/value pairs are represented in the batch Fields line as dot-separated names. For example, the credit_card object on the API contains the following name/value pairs:

```
"credit_card": {
  "type": "{string}",
  "cardholder_name": "{string}",
  "card_number": "{string}",
  "exp_date": "{string}",
  "cvv": "{string}"
```

On the Fields line, the object name/value pairs are defined as:

```
credit_card.type,credit_card.cardholder_name,credit_card.card_number,credit_card.exp_date
```

Note that the `credit_card.cvv` field is not included in the Fields line for the batch file as it is not supported in batch.

Additional fields from the optional objects defined in the API may be added to the batch Fields definition as necessary.

3.1.2.2 API Specification Location

To access the FirstAPI specifications on the FirstAPI Developer Portal, click here. (https://developer.payeezy.com/apis)

3.2 Transaction Line(s)

A transaction line contains the data for a single transaction. The line consists of a comma separated list of fields. The fields in the transaction line must correspond to the list of field names in the Fields Line. Missing fields are specified using an empty string. The number of commas in the Transaction Line must be N-1 where N is the number of fields in the Fields Line.

The following example shows a Transaction line corresponding to the Fields line defined above.

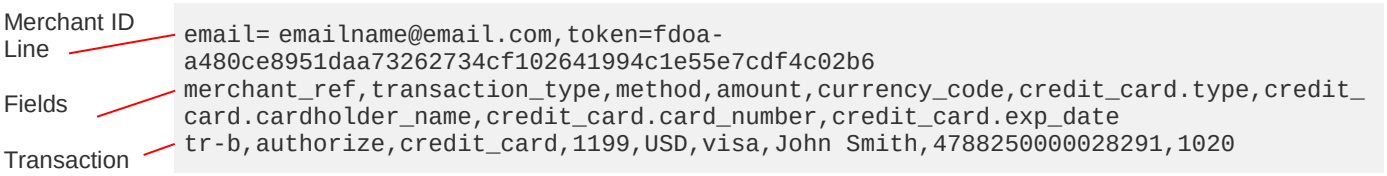
```
tr-b,authorize,credit_card,1199,USD,visa,John Smith,4788250000028291,1020
```

3.3 Input File Examples

The following examples show various input files.

3.3.1 Credit Card Payments - Authorize

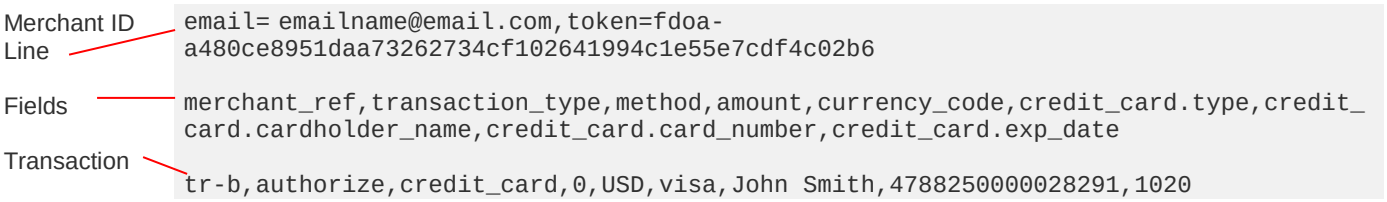
The following example shows a Credit Card Payments transaction batch input file with the Merchant Identification Line, the Configuration line, and a single Authorize transaction line.



3.3.2 Credit Card Payments – Authorize (with Zero Amount to Receive Tokenized Card Number)

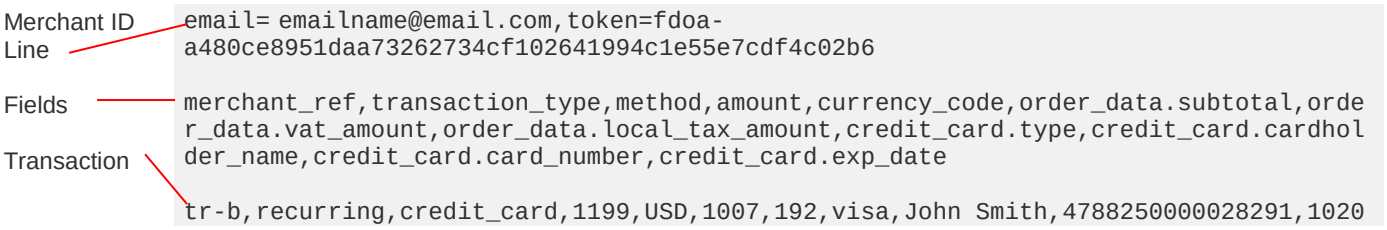
Currently, the batch processing system does not support the Tokenize Credit Card API. Merchants can tokenize a credit card numbers by sending an Authorize transaction with the amount equal to zero (0). The response file will contain a tokenized credit card number that can be stored by the merchant and used in subsequent Token-Based Payment transactions.

The following example shows a Credit Card Payments transaction batch input file with the Merchant Identification Line, the Configuration line, and a single zero amount Authorize transaction line.



3.3.1 Credit Card Payments – Recurring (with Local Tax fields)

The following example shows a Credit Card Payments transaction batch input file with the Merchant Identification Line, the Configuration line, and a single Recurring transaction line. This example also shows the optional Local Tax fields. For complete definitions of the tax fields, refer to the API definition on the Developer Portal.




```
email= emailname@email.com,token=fdoa-
a480ce8951daa73262734cf102641994c1e55e7cdf4c02b6

merchant_ref,transaction_type,method,amount,currency_code,order_data.subtotal,orde
r_data.vat_amount,order_data.local_tax_amount,credit_card.type,credit_card.cardhol
der_name,credit_card.card_number,credit_card.exp_date

tr-b,recurring,credit_card,1199,USD,1007,192,visa,John Smith,4788250000028291,1020
```

4. Output File Format and Layout

The transaction output file is a CSV file where the first line describes the configuration of the file and each additional line describes a single transaction.

4.1 Configuration Line

The first lines of the transaction output file lists the fields that will appear in the transaction lines of the file. The fields correspond to the Response of the corresponding API shown on the FirstAPI Developer Portal.

The following example shows an output file configuration line that corresponds to the sample input file shown in the previous section.

```
merchant_ref,correlation_id,transaction_status,validation_status,transaction_type,tran
saction_id,transaction_tag,method,amount,currency,avs,bank_resp_code,bank_message,gate
way_resp_code,gateway_message,token.token_type,token.token_data.value,card.type,card.c
ard_number,card.exp_date,card.cardholder_name,error_message
```

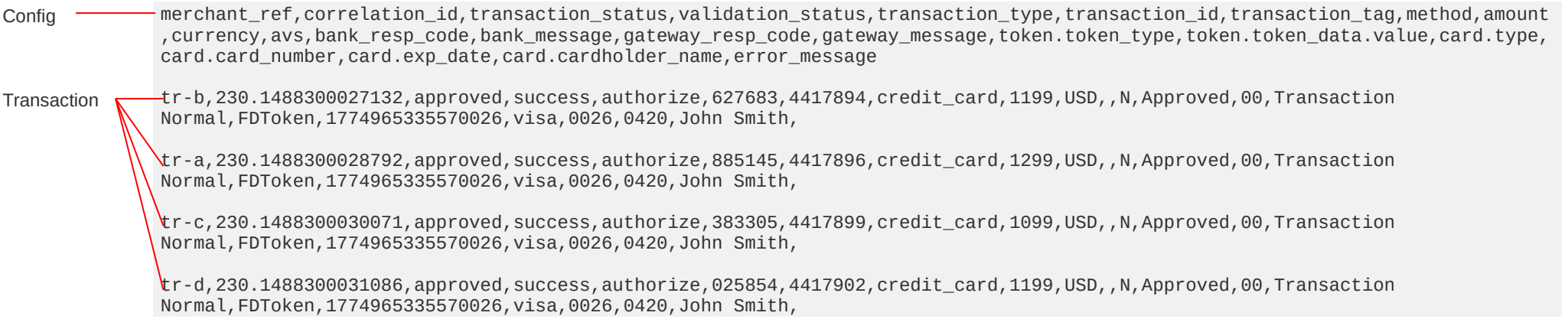
4.2 Transaction Line(s)

A transaction line contains the data for a single transaction. The line consists of a comma separated list of fields. The fields in the batch configuration line correspond to the fields shown for the equivalent Response as shown in the API specification on the Developer Portal.

The fields in the transaction line correspond to the list of field names in the Configuration line. Missing fields are specified using an empty string. The number of commas in the Transaction Line will be N-1 where N is the number of fields in the Configuration line.

4.3 Output File Example

The following example shows an output file with a configuration line and corresponding Transaction lines.




```
merchant_ref,correlation_id,transaction_status,validation_status,transaction_type,transaction_id,transaction_tag,method,amount
,currency,avs,bank_resp_code,bank_message,gateway_resp_code,gateway_message,token.token_type,token.token_data.value,card.type,
card.card_number,card.exp_date,card.cardholder_name,error_message

tr-b,230.1488300027132,approved,success,authorize,627683,4417894,credit_card,1199,USD,,N,Approved,00,Transaction
Normal,FDToken,1774965335570026,visa,0026,0420,John Smith,

tr-a,230.1488300028792,approved,success,authorize,885145,4417896,credit_card,1299,USD,,N,Approved,00,Transaction
Normal,FDToken,1774965335570026,visa,0026,0420,John Smith,

tr-c,230.1488300030071,approved,success,authorize,383305,4417899,credit_card,1099,USD,,N,Approved,00,Transaction
Normal,FDToken,1774965335570026,visa,0026,0420,John Smith,

tr-d,230.1488300031086,approved,success,authorize,025854,4417902,credit_card,1199,USD,,N,Approved,00,Transaction
Normal,FDToken,1774965335570026,visa,0026,0420,John Smith,
```

First Data

5. File Transfer Process

Both input and output files are transferred between the client and First Data using the First Data File Gateway (FDFG). FDFG supports multiple secure transmission protocols, including SFTP and HTTPS. The procedures in this guide refer to file transfer using SFTP through FDFG's Sterling File Gateway (SFG).

Note: While developers and merchants can board to FirstAPI through the Developer Portal, boarding to the First Data File Gateway is best accomplished through working with First Data's Support Team. Please contact First Data for more information.

5.1 Prerequisites

Before beginning the batch transfer process, merchants will need to complete the following tasks:

- Board to the appropriate gateway.
- Register on the Developer Portal and complete the certification requirements. Refer to the FirstAPI Developer Portal (https://developer.payeezy.com) for details.
- Work with First Data L2 Support team to complete the *FDFG Boarding Questionnaire* and receive their SFG ID and password. Refer to the *First Data File Gateway Boarding Questionnaire*. The specific file naming standards needed for the batch process are described earlier in this document. Clients may also find the *First Data File Gateway External Client Boarding Guide* helpful during this step.
    - This step also includes the creation of appropriate authentication artifacts, such as a public/private key pair for SFTP transfers or an SSL certificate for HTTPS transfers.

FDFG supports both a Client Acceptance Testing (CAT) and Production environment, as detailed in the *First Data File Gateway External Client Boarding Guide.* User account IDs and passwords are the same for both environments.

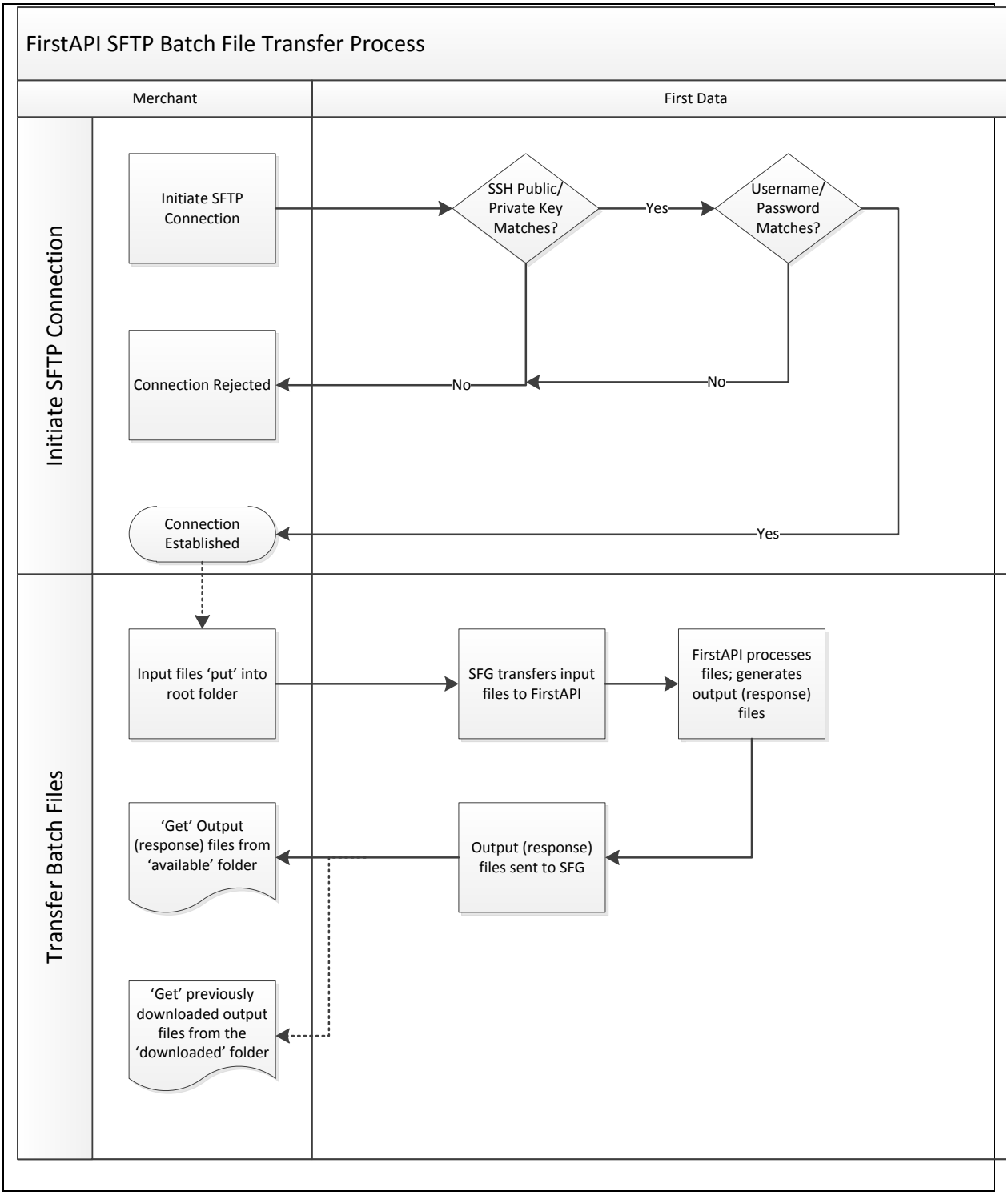
Currently, the connection data is as follows:

	CAT (NA)	Production (NA)
Domain Name	test-gw-na.firstdataclients.com	prod-gw-na.firstdataclients.com
SFTP Port	6522	6522

Any discrepancies between this guide and the documentation provided by FDFG should be resolved in favor of the requirements published by FDFG.

5.1 File Transfer Flow

The following diagram shows the input and output flows for a batch file submitted to FirstAPI using SFTP.



5.1.1 Input File Transfer Flow

1. Log-in to the FDFG site using the credentials provided when boarded (user name, password, SSH key).

2. Use the 'put' command to place the transaction file into the root folder. The file must named according the standards described earlier in this document and must have the correct format.

5.1.2 Output (Response) File Retrieval Flow

1. Log-in to the FDFG site using the credentials provided when boarded.

2. Change the current folder:
   - The `'available'` folder holds any currently processed files.
   - The `'downloaded'` folder holds any files previously downloaded by the client.

3. Use the 'get' command to retrieve the response files in the folder. Note that SFG appends a timestamp to the end of the filename.

5.2 SFTP Examples

Files sent using SFTP are transferred with commands issued on the command line. This section describes the input and output file transfer processes in a Windows environment and in a Linux environment.

Note: The environment used in the example commands is the CAT environment.

5.2.1 Windows

Note that Windows does not include native support for file transfer, so an external file transfer application such ast PuTTy must be downloaded and installed. The following process assumes that PuTTy has been downloaded and installed.

To transfer files in a Windows environment:

1. From the Windows command line, type the following command:
   ```
   psftp -i <public key file> -l <username> -P 6522 test-gw-na.firstdataclients.com
   ```

2. When prompted, enter the client password.

3. Use the 'put' command to upload the transaction input file to the SFTP server.
   ```
   psftp put <file path>\<file prefix>.<file name>.csv
   ```

4. First Data places the output (response) file in the `'available'` folder once the input file has been processed.

5. To retrieve the output file, use the `'get'` command to download the response file. Once you download the file, SFG automatically moves it to the `'downloaded'` folder, where it remains accessible.

5.2.2 Unix / macOS

To transfer files in a Unix or macOS environment:

1. From the command line, type the following command:
   `sftp -i <public key file> -P 6522 <username>@test-gw-na.firstdataclients.com`

2. When prompted, enter the client password.

3. Use the `'put'` command to upload the transaction input file to the SFTP server.

4. First Data places the output (response) file in the `'available'` folder once the input file has been processed.

5. To retrieve the output file, use the `'get'` command to download the response file. Once you download the file, SFG automatically moves it to the `'downloaded'` folder, where it remains accessible.

First Data

6. Revision History

Revision Number / Date	Description
V2017-11 – 9 Nov 2017	Initial version of document
V2017-11a – 27 Nov 2017	Clarified supported APIs

First Data.