

1 Document Control

Title	API External Release Notes Sprint 17 – 1 Mar 2019
Document Owner	Integrated Solutions IT Program Manager APAC
Author	Amanda Emerson
Date Created	15 Feb 2019
Document Purpose	To advise partner developers of API changes in the upcoming AFEX production release and enable them to make any code changes to their software accordingly.
Target Audience	Partner developers

Revision History

Version	Date	Author	Change Details
0.1	26 Feb 2019	Amanda Emerson	Draft for review
1.0	27 Feb 2019	Amanda Emerson	Version approved 27 Feb 2019 by Brad Anderson– Product Director API

2 Release Implementation Dates

Environment	Implementation Date	Current Status
Production	1 Mar 2019 PST	Pending Implementation
Demo	To be confirmed	Pending Implementation

3 AFEX API Documentation

A new version of the AFEX API documentation (v5.1 2019) will be published to reflect any changes in this document.

<https://doc.api.afex.com/>

4 Release Notes / Features

Note, the end point URLs shown throughout this document are the AFEX Demo Environment, and AFEX UAT environment. These are shown for testing / illustration purposes and are not production end points.

4.1 Feature Summary

The table below is a summary of the new features / changes described in this document.

AFEX Ref#	Feature
43704	Credit limit exceeded - return a meaningful error message when booking a trade where the client's trade credit limit is exceeded
43813	Provide more accurate rate and settlement amount for trades
43691	Display complete customer rate on trade confirmations
43922	Add "Note" field to api/payments/create and api/payments?id=

4.2 43704 – Credit Limit Exceeded Response

This change applies to the following API methods:

- [POST api/trades/create](#)
- [POST api/forwards/create](#)

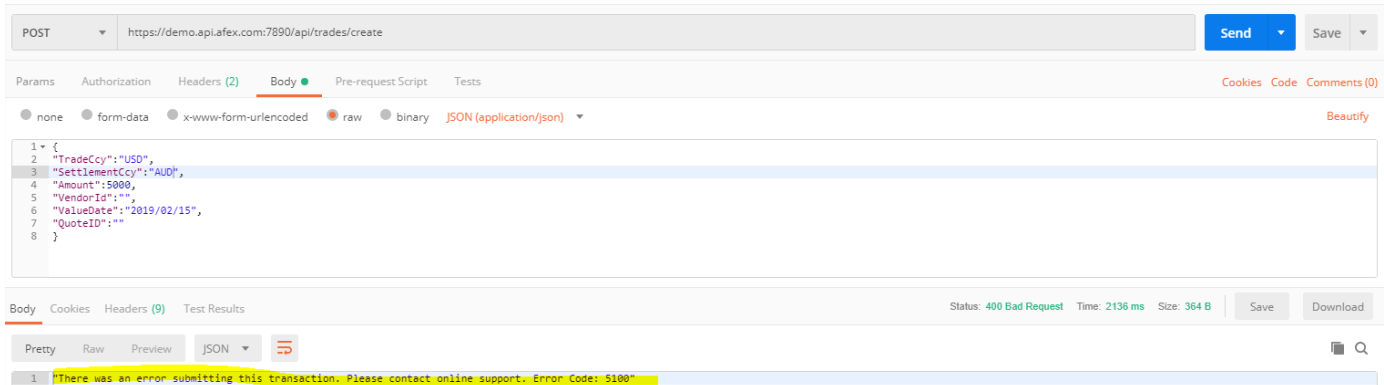
Currently, the error response returned when the client exceeds their spot or forward limit is "error 5100". This is not meaningful to the end user. This change will introduce the more meaningful response "This transaction exceeds the allowed limit on this account".

Example of current / AS IS functionality

The screen shot below shows the current response to [POST api/trades/create](#) when the credit limit is exceeded:

Note the following issues with the current response:

- The error message is not meaningful

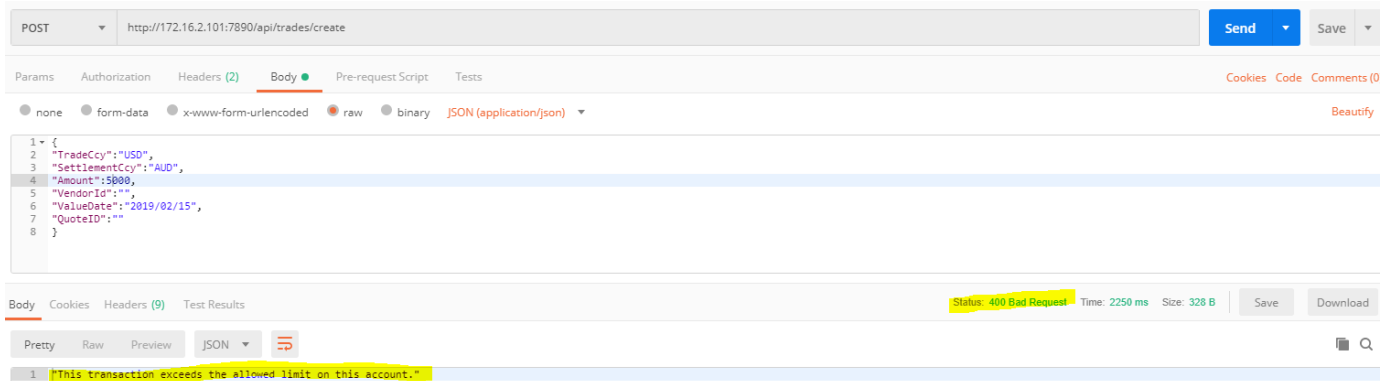


Example of new / TO BE functionality

Using [POST api/trades/create](#), the same trade data as per above is used to demonstrate the new functionality to be implemented.

Note the following with respect to the screen shot of new functionality below:

- The status code returned remains 400 Bad Request as per the AS IS response
- The error message has been changed to be more meaningful i.e. "This transaction exceeds the allowed limit on this account".



4.3 43813 – Provide more accurate rates and settlement amounts / 43691 – Display complete rate on trade confirmations

This change applies to the following API methods:

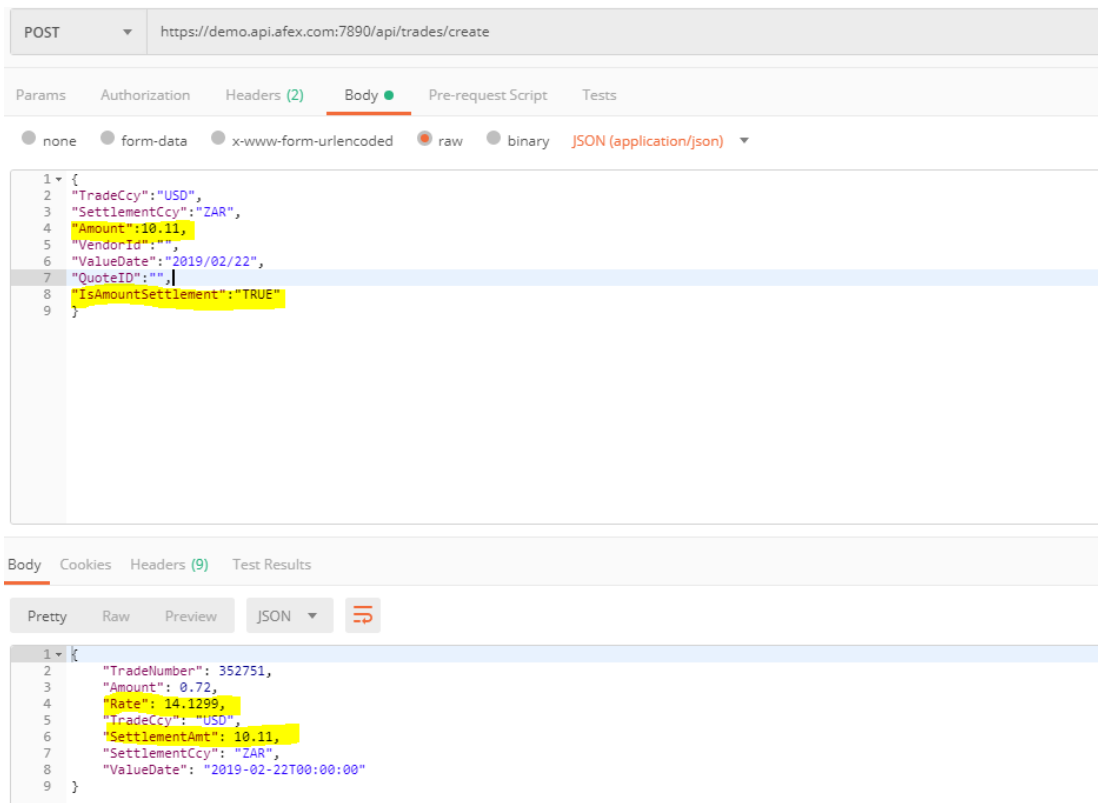
- [POST api/trades/create](#)
- [POST api/forwards/create](#)

Currently, when sending a trade where the “Amount” is the Settlement Amount i.e. `IsSettlementAmount=TRUE`, it is possible that the settlement amount on the trade history and PDF trade confirmation can vary from the settlement amount specified when the trade was originally created. To correct this, the rate calculation has been adjusted to add more decimal places to ensure the settlement amount accuracy. The settlement amount shown in the response at the time the trade is booked will match the trade history, and the PDF trade confirmation emailed to the client. Further to this, the rate returned by the API and displayed on the trade confirmation will include all decimal places.

Example of current / AS IS functionality

Book a trade:

The screen shot below shows the current response to [POST api/trades/create](#) where the Amount sent in the trade is the Settlement Amount. Note, the response shows the Amount is 10.11 and the rate is returned is to 4 decimal places.



The screenshot displays a REST client interface for a POST request to `https://demo.api.afex.com:7890/api/trades/create`. The request body is a JSON object with the following fields: `TradeCcy` (USD), `SettlementCcy` (ZAR), `Amount` (10.11), `VendorId` (empty), `ValueDate` (2019/02/22), `QuoteID` (empty), and `IsAmountSettlement` (TRUE). The response body is a JSON object with the following fields: `TradeNumber` (352751), `Amount` (0.72), `Rate` (14.1299), `TradeCcy` (USD), `SettlementAmt` (10.11), `SettlementCcy` (ZAR), and `ValueDate` (2019-02-22T00:00:00).

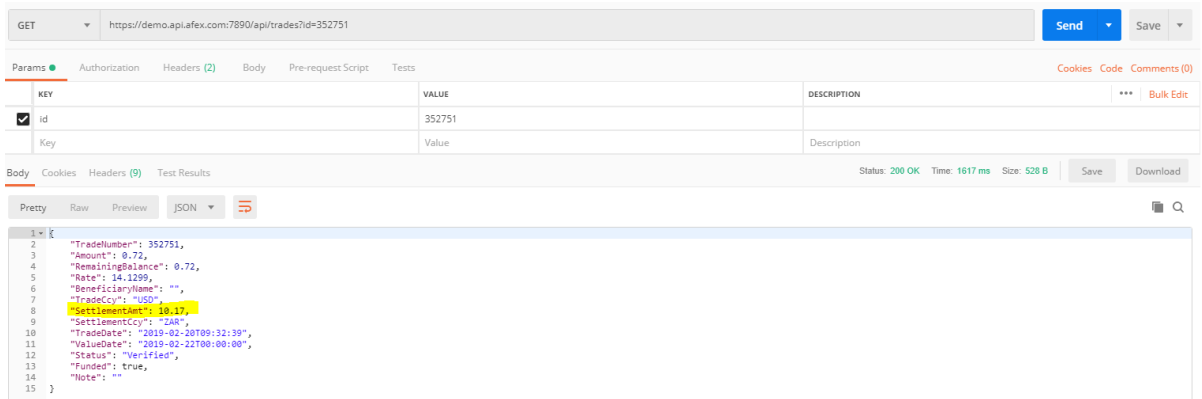
```
POST https://demo.api.afex.com:7890/api/trades/create

Body
  none form-data x-www-form-urlencoded raw binary JSON (application/json)
  1 {
  2   "TradeCcy": "USD",
  3   "SettlementCcy": "ZAR",
  4   "Amount": 10.11,
  5   "VendorId": "",
  6   "ValueDate": "2019/02/22",
  7   "QuoteID": "",
  8   "IsAmountSettlement": "TRUE"
  9 }

Body Cookies Headers (9) Test Results
  Pretty Raw Preview JSON
  1 {
  2   "TradeNumber": 352751,
  3   "Amount": 0.72,
  4   "Rate": 14.1299,
  5   "TradeCcy": "USD",
  6   "SettlementAmt": 10.11,
  7   "SettlementCcy": "ZAR",
  8   "ValueDate": "2019-02-22T00:00:00"
  9 }
```

Get trade history:

Now using [GET api/trades?id=352751](https://demo.api.afex.com/7890/api/trades?id=352751), the trade created above is retrieved and it can be observed that the settlement amount shown is now 10.17 which varies from the original settlement amount submitted which was 10.11.



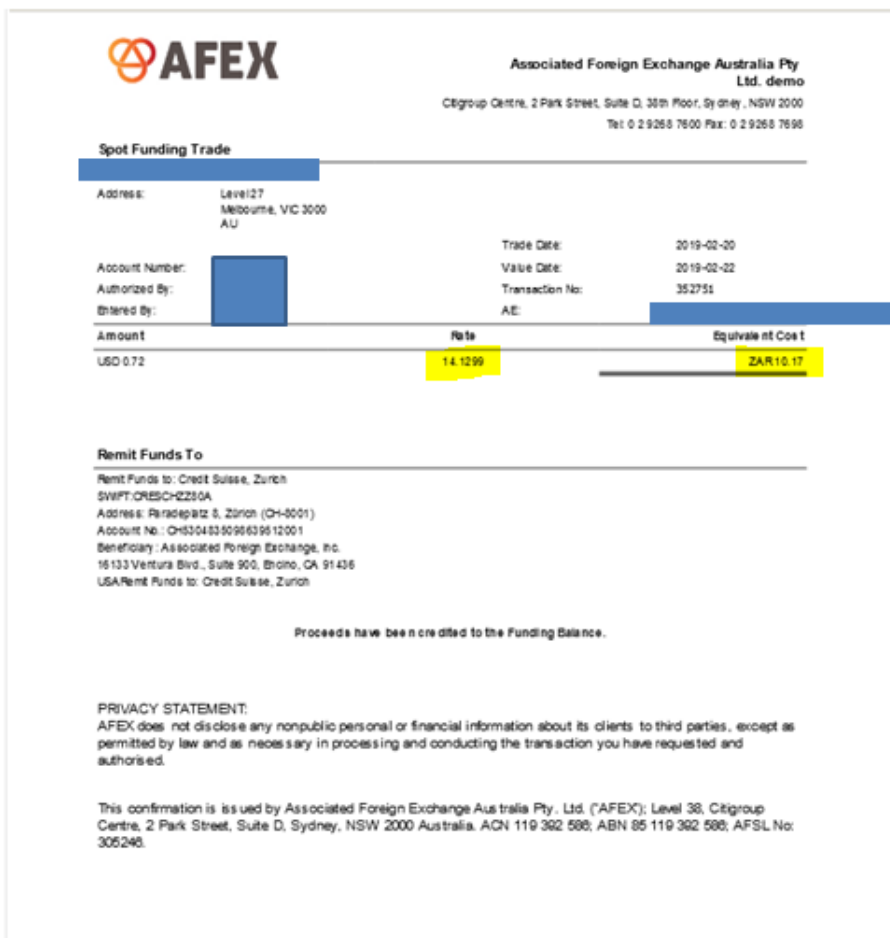
```

1 {
2   "TradeNumber": 352751,
3   "Amount": 0.72,
4   "RemaininBalance": 0.72,
5   "Rate": 14.1299,
6   "BeneficiaryName": "",
7   "TradeCcy": "USD",
8   "SettlementAmt": 10.17,
9   "SettlementCcy": "ZAR",
10  "TradeDate": "2019-02-20T08:32:39",
11  "ValueDate": "2019-02-22T08:00:00",
12  "Status": "Verified",
13  "Funded": true,
14  "Note": ""
15 }

```

Trade confirmation:

The PDF trade confirmation also shows the settlement amount is 10.17 which varies from the original settlement amount submitted of 10.11.



AFEX
Associated Foreign Exchange Australia Pty Ltd. demo
Citigroup Centre, 2 Park Street, Suite D, 38th Floor, Sydney, NSW 2000
Tel: 0 2 9268 7600 Fax: 0 2 9268 7698

Spot Funding Trade

Address: Level 27
Melbourne, VIC 3000
AU

Account Number: [Redacted]
Authorized By: [Redacted]
Entered By: [Redacted]

Trade Date: 2019-02-20
Value Date: 2019-02-22
Transaction No: 352751
A/E: [Redacted]

Amount	Rate	Equivalent Cost
USD 0.72	14.1299	ZAR 10.17

Remit Funds To

Remit Funds to: Credit Suisse, Zurich
SWIFT: CRESCH2330A
Address: Paradeplatz 5, Zurich (CH-8001)
Account No.: CH5304303096639912001
Beneficiary: Associated Foreign Exchange, Inc.
16133 Ventura Blvd., Suite 900, Bruno, CA 91436
USA Remit Funds to: Credit Suisse, Zurich

Proceeds have been credited to the Funding Balance.

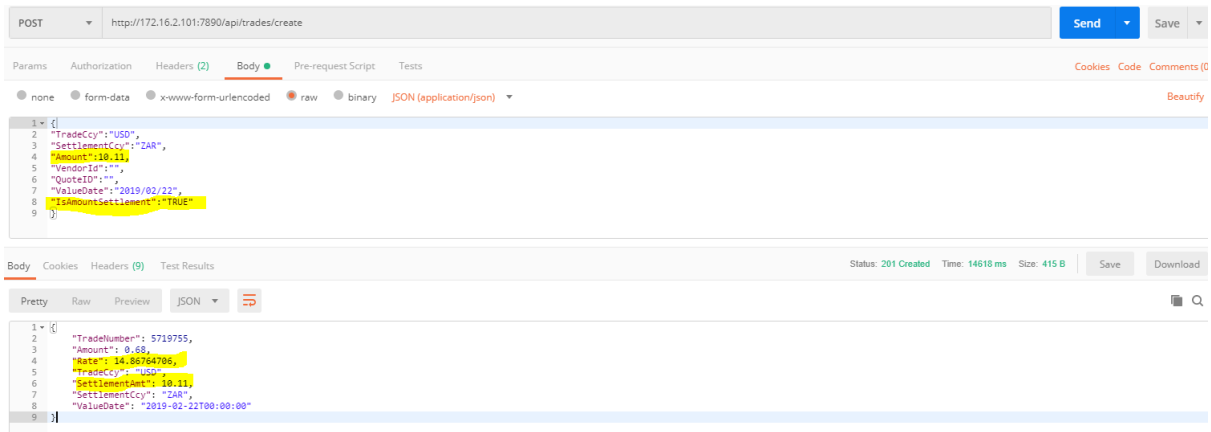
PRIVACY STATEMENT:
AFEX does not disclose any nonpublic personal or financial information about its clients to third parties, except as permitted by law and as necessary in processing and conducting the transaction you have requested and authorized.

This confirmation is issued by Associated Foreign Exchange Australia Pty. Ltd. ("AFEX"); Level 38, Citigroup Centre, 2 Park Street, Suite D, Sydney, NSW 2000 Australia. ACN 119 362 588; ABN 85 119 362 588; AFSL No: 305246.

Example of new / TO BE functionality – Fix for settlement amount discrepancy

Book a trade:

The screen shot below shows the TO BE response to [POST api/trades/create](#) where the Amount sent in the trade is the Settlement Amount. Note in the response returned, the Amount is 10.11 and the rate is to 8 decimal places.



The screenshot shows a REST client interface with a POST request to `http://172.16.2.101:7890/api/trades/create`. The request body is a JSON object:

```

1 {
2   "TradeCcy": "USD",
3   "SettlementCcy": "ZAR",
4   "Amount": 10.11,
5   "VendorID": "",
6   "QuoteID": "",
7   "ValueDate": "2019-02-22",
8   "IsAmountSettlement": true
9 }

```

The response body is a JSON object:

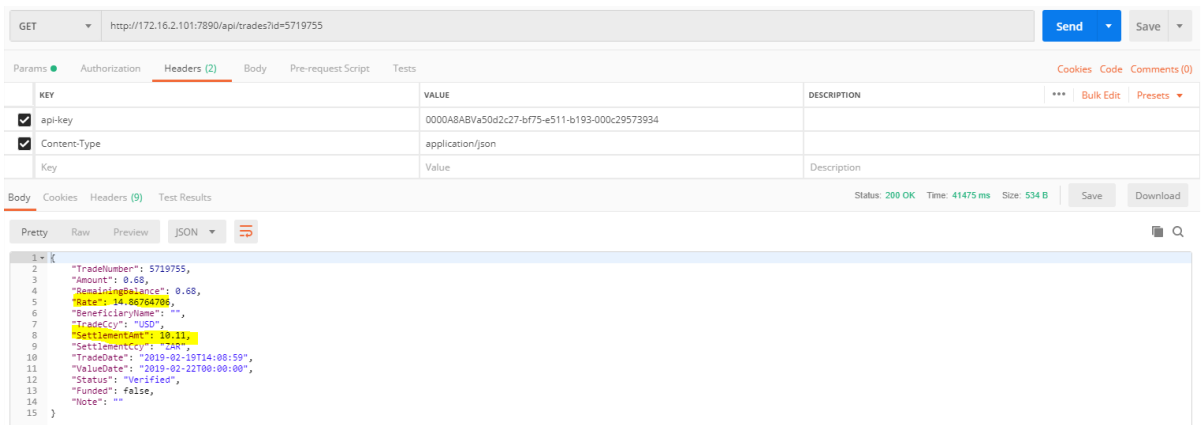
```

1 {
2   "TradeNumber": 5719755,
3   "Amount": 0.68,
4   "Rate": 14.86764706,
5   "TradeCcy": "USD",
6   "SettlementAmount": 10.11,
7   "SettlementCcy": "ZAR",
8   "ValueDate": "2019-02-22T00:00:00"
9 }

```

Get trade history:

Now using [GET api/trades?id=5719755](#), the trade created above is retrieved and it can be observed that the settlement amount shown remains 10.11 which matches the original settlement amount submitted in the `api/trades/create`. The rate is also shown to the full 8 decimal places.



The screenshot shows a REST client interface with a GET request to `http://172.16.2.101:7890/api/trades?id=5719755`. The Headers tab shows:

KEY	VALUE	DESCRIPTION
api-key	0000A8ABVa50d2c27-bf75-e511-b199-000c29573934	
Content-Type	application/json	
Key	Value	Description

The response body is a JSON object:

```

1 {
2   "TradeNumber": 5719755,
3   "Amount": 0.68,
4   "RemainingBalance": 0.68,
5   "Rate": 14.86764706,
6   "BeneficiaryName": "",
7   "TradeCcy": "USD",
8   "SettlementAmount": 10.11,
9   "SettlementCcy": "ZAR",
10  "TradeDate": "2019-02-19T14:08:59",
11  "ValueDate": "2019-02-22T00:00:00",
12  "Status": "Verified",
13  "Funded": false,
14  "Note": ""
15 }

```



API External Release Notes Release Sprint 1 Mar 2019

Trade confirmation:

The PDF trade confirmation also shows the settlement amount is 10.11 which matches the original settlement amount submitted in the api/trades/create. The rate is also shown to the full 8 decimal places.

		Associated Foreign Exchange, Inc. uat 21045 Califa St., Woodland Hills, CA 91367 Tel: (866) 459-2339 Fax: (818) 386-3295	
Spot Funding Trade			
Customer:	AFEX ERP Validation USA		
Address:	21045 Califa Street Woodland Hills, FL 91367 US		
Account Number:	0000ABAB	Trade Date:	2019-02-19
Authorized By:	Chandana Sarkar	Value Date:	2019-02-22
Entered By:	Chandana Sarkar	Transaction No:	5719755
		AE:	CHRISTOPHER EMPETT
Amount	Rate	Equivalent Cost	
USD 0.68	14.86764708	ZAR 10.11	
Remit Funds To			
Remit Funds to: Credit Suisse, Zurich			
SWIFT:CRESCHZZ80A			
Address: Paradeplatz 8, Zürich (CH-8001)			
Account No.: CH5304835098639512001			
Beneficiary: Associated Foreign Exchange, Inc.			
16133 Ventura Blvd., Suite 900, Encino, CA 91436			
USARemit Funds to: Credit Suisse, Zurich			
Proceeds have been credited to the Funding Balance.			
PRIVACY STATEMENT:			
AFEX does not disclose any nonpublic personal or financial information about its clients to third parties, except as permitted by law and as necessary in processing and conducting the transaction you have requested and authorized.			
CONSUMER DISCLOSURE:			
Associated Foreign Exchange, Inc. ("AFEX") is a licensed money transmitter under Chapter 560, Florida Statutes. For suspected violations of Chapter 560, Florida Statutes, call AFEX at (888) 428-2339 or contact, by mail, the Florida Office of Financial Regulation, 200 East Gaines Street, Tallahassee, FL 32399-0376, (850) 487-9687.			

4.4 43922 – Add Note field to Payments in API

This change applies to the following API methods:

- [POST api/payments/create](#)
- [GET api/payments?id=<ReferenceNumber>](#)

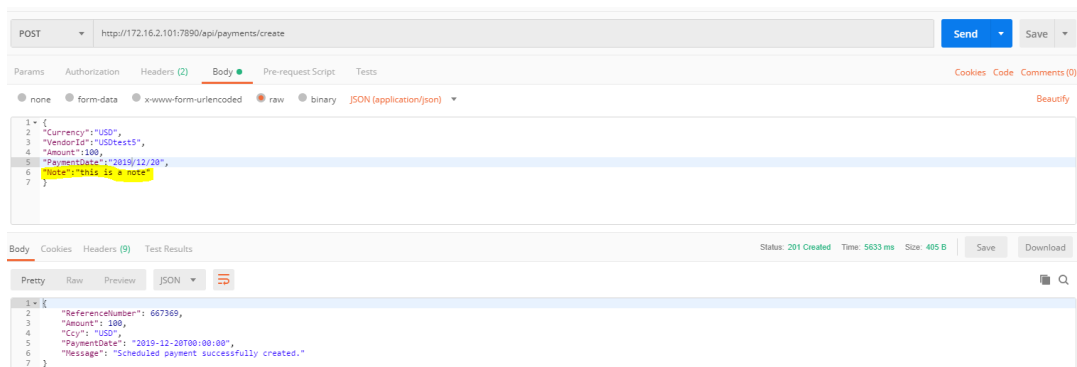
A new parameter “Note” will be available to be sent in [POST api/payments/create](#). If “Note” data is sent in an [api/payments/create](#), the parameter and its data will be returned in a query for that individual payment i.e. [GET api/payments?id=<ReferenceNumber>](#). Due to a limitation in the AFEX back end system, the **Note** field will **not** be returned when retrieving multiple payments i.e. [GET api/payments](#).

Note parameter specifications:

Parameter Name	Purpose / Use	Mandatory / Optional	Format	Error Message
Note	The end user can enter free text to store their own notes / reference information about the transaction	Optional	Alphanumeric string 250 characters @“^([a-zA-Z0-9/\- \?:\(\),\.\+\x20]){1,250}\$”	Invalid Internal Note

Example of new / TO BE functionality

Using [POST api/payments/create](#), a payment was created which includes data in the “Note” field.



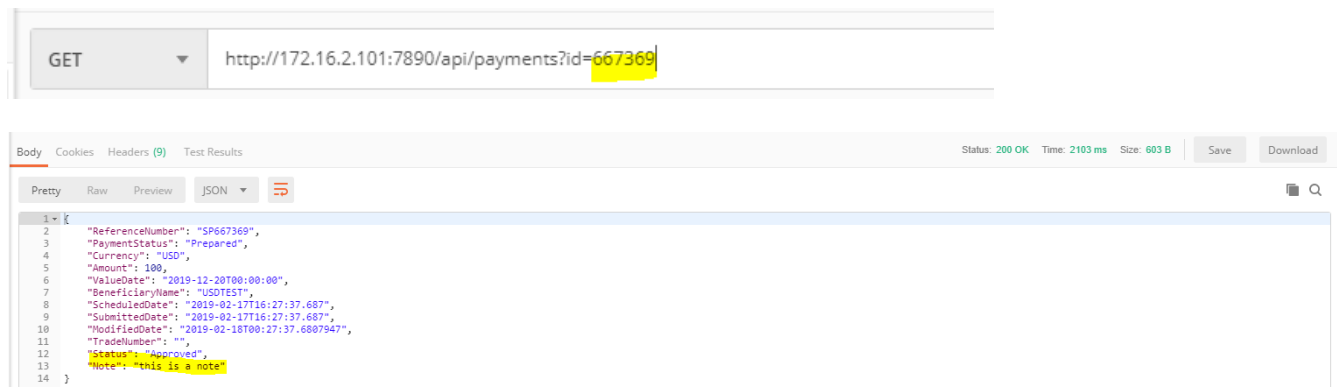
```

1 {
2   "Currency": "USD",
3   "VendorId": "USDTEST1",
4   "Amount": 100,
5   "PaymentDate": "2019-12-20",
6   "note": "this is a note"
7 }
    
```

```

1 {
2   "ReferenceNumber": 667369,
3   "Amount": 100,
4   "Ccy": "USD",
5   "PaymentDate": "2019-12-20T00:00:00",
6   "Message": "Scheduled payment successfully created."
7 }
    
```

Using [GET api/payments?id=](#) to retrieve the payment just created, please observe that the “Note” field is returned in the response. This matches data sent in the “Note” field when the payment was created.



```

1 GET http://172.16.2.101:7890/api/payments?id=667369
    
```

```

1 {
2   "ReferenceNumber": "SP667369",
3   "PaymentStatus": "Prepared",
4   "Currency": "USD",
5   "Amount": 100,
6   "ValueDate": "2019-12-20T00:00:00",
7   "BeneficiaryName": "USDTEST1",
8   "ScheduledDate": "2019-02-17T16:27:37.687",
9   "SubmittedDate": "2019-02-17T16:27:37.687",
10  "ModifiedDate": "2019-02-18T00:27:37.6807947",
11  "TradeNumber": "",
12  "Status": "Approved",
13  "Note": "this is a note"
14 }
    
```