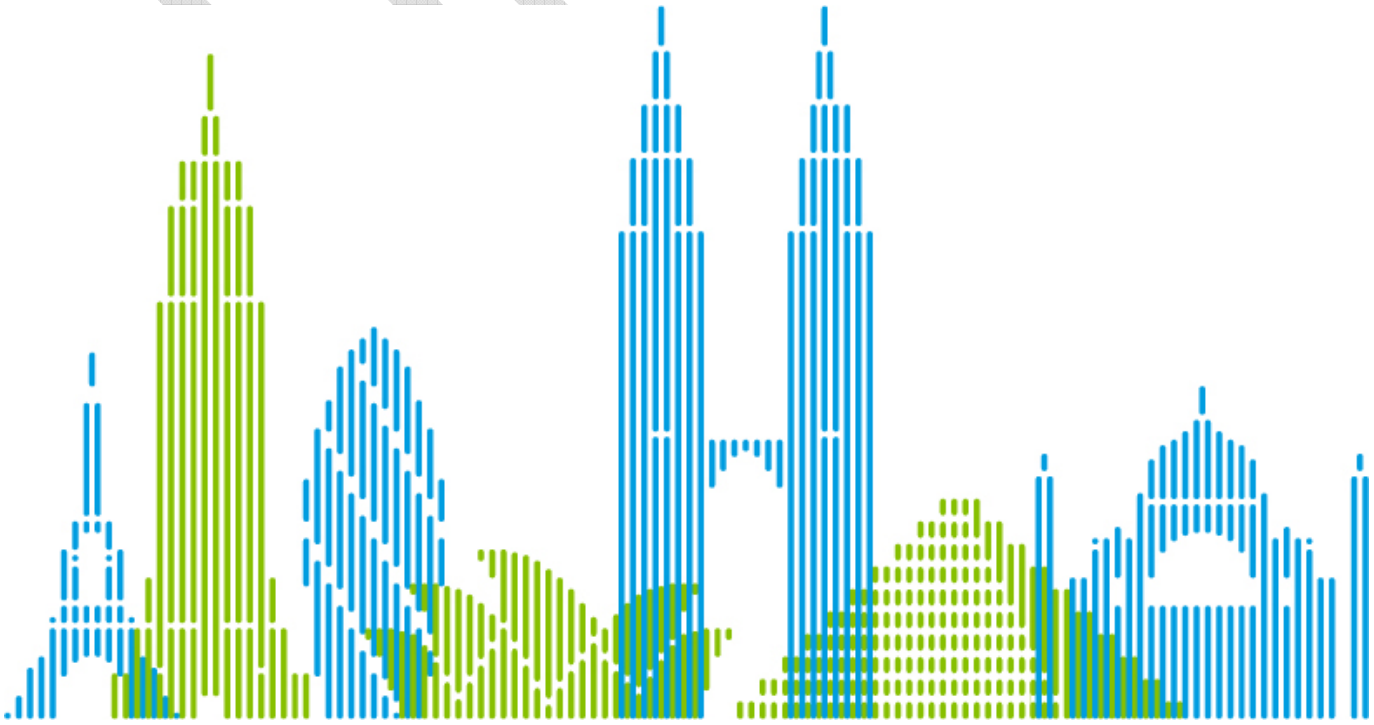
On your side

World First
Foreign Exchange

AutoPay
Integration guide
version 1.13



Contents

Introduction

This document describes the technical specification of the World First AutoPay web service.

The invocation of AutoPay involves the sending and receiving of XML messages to/from a web service via HTTPS.

The messages are described in this document via sample XML files and a message elements tables that detail the usage of each element in the request and response.

Code	Description
M	Mandatory
O	Optional
C	Conditional

Obtaining Access

Before AutoPay can be accessed, an account must first be set up for clients of the API including a username and password; also a secret key must be set and exchanged between the parties in order to create the required tokens that are passed with each API request.

Once these pre-requisites have been met, the API can be accessed at:

System	Endpoint
Live	https://autopay.worldfirst.com/service/v1.13/hash/<hash>
Staging	https://staging-autopay.worldfirst.com/service/v1.13/hash/<hash>

Sample PHP, C# and JAVA interface programs as well as schemas and test data are provided at:

https://staging-autopay.worldfirst.com/samples/

Please note that all requests processed through the staging site will not incur any costs or be traded in the market. All requests processed through the live site will incur costs and be traded in the market. Any trades you make in the market are contractual and require funding to World First by the settlement date in the response.

Authentication

Every request sent to AutoPay is authenticated on two levels: HTTP authentication and hash authentication.

HTTP authentication

HTTP authentication should be included with each request. A username and password for each client will be supplied by World First, to be used in HTTP authentication.

Username	demo
Password	d3m0u5r

Token authentication

Once HTTP authentication of the client has passed, each request will contain a hash to be validated using a secret hashing key provided by World First to the client. Sending of an invalid hash will result in the request packet not being processed. The hash must be sent with each request to the AutoPay API.

The algorithm to create the hash is as follows:

1. Create an MD5 hash of the request XML packet. Note: create the MD5 hash of only the XML packet containing the request, not the entire request itself.
2. Use HMAC (Hash-based Message Authentication Code) to create a HMAC MD5 hash of the MD5 hash you just created, using the secret key.

Append the complete hash to the URL in the request. Eg:

https://autopay.worldfirst.com/service/v1.13/hash/<hash>

Where <hash> is the hash from the HMAC hash you have just created.

Secret Key	abcdefghihjklmnopqrstuvwxyz0123456789

Request structure

Every request sent to AutoPay must be in XML format and follow the schema shown in Figure 1.

The elements must also be in the order specified in the schema, so although n `<getquote>` elements may be included in the request, they must all appear before any `<payment>` elements in the XML request.

Request elements

Tag	Type	Description	Usage
atomic	boolean	If true, tells AutoPay to execute all sub-requests (`<payment>`s and `<getquote>` requests) in the packet or none. I.e. if one message fails, do not process any of the messages.	M
client-details	void	Indicates that account specific settings are to be sent back in the response packet.	O
getquote	getquote	Message object for a `<quote>` on a particular symbol. Can include multiple instances of this object.	O
payment	payment	Message object for a `<payment>` on a `<quote>`, or a `<payment>` without a `<quote>` (a "trade implied" `<payment>`). Can include multiple instances of this object.	O
past-paymentid	string	The payment id of a previously processed `<payment>`. Can include multiple instances of this object.	O
past-payment-range	past-payment-range	Sub-request object for a range of payment dates. All past `<payment>`s within the range will be returned. Only one instance of this object is permitted.	O

Request sample

```
<?xml version="1.0"?>
<request>
    <atomic>false</atomic>

    <client-details />

    <getquote>
        **** getquote structure ****
    </getquote>

    <payment>
        **** payment structure ****
    </payment>

    <past-paymentid>
        **** payment id of previously processed payment ****
    </past-paymentid>

    <past-payment-range>
        **** past-payment-range structure ****
    </past-payment-range>

</request>
```

Response structure

The elements that may be included in a response packet are defined below.

Response elements

Tag	Type	Description	Usage
atomic	boolean	Indicates how the corresponding request was handled. Will take the value passed in the request it is responding to.	C
error	error	Any errors that may have occurred in high level packet processing.	C
client-details	client-details	Response details of a request for `<client-details />`, if they were requested.	O
getquote	getquote	Response details of any `<getquote>`s in the request.	O
payment	payment	Response details of any `<payment>`s in the request.	O
past-payment	past-payment	Response details of any `<past-paymentid>`s in the request, or `<past-payment>`s requested by `<past-payment-range>`.	O
trade-confirmation	trade-confirmation	Includes the details of any foreign exchange deals transacted to service `<payment>`s or `<past-payment>`s. Also includes the funds in details of where account holders should pay the settlement balance to service their `<payment>`s.	O

Response sample

```
<?xml version="1.0"?>
<response>
    <atomic>false</atomic>

    <client-details>
        **** client-details structure ****
    </client-details>

    <quote>
        **** quote structure ****
    </quote>

    <payment>
        **** payment structure ****
    </payment>

    <past-payment>
        **** past-payment structure ****
    </past-payment>

    <trade-confirmation>
        **** trade-confirmation structure ****
    </trade-confirmation>

</response>
```

Message Formats

Client Details

This message is for checking on account configuration and is not normally used operationally.

Request

In order to request the user data associated with the account making the request, one empty `<client-details />` element just needs to be included in the request packet. This will ensure that the response includes all relevant data to that client such as their deposit percentage or what their set fees are. An example of how to call the client-details is shown below.

Request elements

Tag	Type	Description	Usage
client-details	void	I Indicates that account specific settings are to be sent back in the response packet.	O

Request sample

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <atomic>true</atomic>
    <client-details/>
</request>
```

Response

Response elements

Tag	Type	Description	Usage
forward-deposit-working-days	integer	Number of working days before a forward incurs a deposit.	M
deposit-percentage	float	The percentage of trade sell amounts required as a deposit for trades qualifying for a deposit.	M
payment-fees	float	The cost incurred for each payment request made.	M
payment-fees-currency	string	The currency of the `<payment-fees>`. This fee will be converted to the sell currency of the trades you process `<payment>`s on. Specified as an ISO-4217 alpha 3 code.	M
daily-trade-limit	float	Maximum amount permitted to be traded in a single day.	M
daily-trade-limit-currency	string	Specified as an ISO-4217 alpha 3 code.	M
single-trade-limit	float	Maximum amount permitted to be traded in a single trade.	M
single-trade-limit-currency	string	Specified as an ISO-4217 alpha 3 code.	M
default-sell-currency	string	Specified as an ISO-4217 alpha 3 code.	M
aggregate-payments	boolean	Is the account is permitted to process multiple `<payment>`s per trade.	M

Response sample

```
<?xml version="1.0"?>
<response>
    <atomic>true</atomic>
    <client-details>
        <forward-deposit-working-days>5</forward-deposit-working-days>
        <deposit-percentage>5.00</deposit-percentage>
        <payment-fees>1.95</payment-fees>
        <payment-fees-currency>GBP</payment-fees-currency>
        <daily-trade-limit>10000000</daily-trade-limit>
        <daily-trade-limit-currency>GBP</daily-trade-limit-currency>
        <single-trade-limit>50000</single-trade-limit>
        <single-trade-limit-currency>GBP</single-trade-limit-currency>
        <default-sell-currency>GBP</default-sell-currency>
        <aggregate-payments>true</aggregate-payments>
    </client-details>
</response>
```

Get quote

This message gets a quote for a given currency pair, amount and date. Quotes can later be referenced to process a payment (effectively booking in a trade). Processing quotes alone do not cause funds to be traded or any fees incurred.

Request

Request element

Tag	Type	Description	Usage
settlementdate	date	Date of settlement of the trade. This is the intended date for funds to be traded in the market. If booked, the trade will need to be funded by this date and any payments will be processed on or after this date. If this date is set beyond the `<forward-deposit-working-days>` setting available in a `<client-details />` request, the trade will incur a deposit, due to be settled at an earlier date.	M
buycurr	string	The currency being bought by the account holder (currency transferred from World First to client nominated `<beneficiary>`). Specified as an ISO-4217 alpha 3 code.	M
sellcurr	string	The currency being sold by the account holder (currency transferred from the account holder to World First before the `<settlementdate>`).	M
side	string	The `<side>` of the pair the `<amount>` is specified in. Possible values are (`B`, `S`) for `B`uy or `S`ell. E.g. If specifying to buy 123.45 GBP for EUR `<side>` would be set to `B`.	M
amount	float	The `<amount>` to trade specified in `<side>`'s currency. E.g. If specifying to buy 123.45 GBP for EUR `<amount>` would be set to `123.45`.	M

Request sample

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <atomic>true</atomic>
    <getquote>
        <settlementdate>2011-10-12</settlementdate>
        <buycurr>GBP</buycurr>
        <sellcurr>EUR</sellcurr>
        <side>B</side>
        <amount>123.45</amount>
    </getquote>
</request>
```

Response

Response elements

Tag	Type	Description	Usage
success	boolean	Indicates if the request was successfully processed.	M
tradeid	string	The trade identifier. This is used to reference the quote for a sub sequential `<payment>` request. Not present if the `<quote>` failed.	C
buycurr	string	The currency being bought by the account holder (currency transferred from World First to client nominated `<beneficiary>`). Specified as an ISO-4217 alpha 3 code.	M
sellcurr	string	The currency being sold by the account holder (currency transferred from the account holder to World First before the `<settlementdate>`).	M
buyamt	float	The amount being purchased given in the `<buycurr>`. Not present if the `<quote>` failed.	C
sellamt	float	The amount being sold given in the `<sellcurr>`. Not present if the `<quote>` failed.	C
rate	float	The rate used to convert `<amount>` given in the request. See Appendix B for trading symbol calculation. Not present if the `<quote>` failed.	C
settlementdate	date	Date of settlement of the trade.	M
expiry	integer	The number of seconds the `<quote>` is valid for. Attempting to book a `<payment>` on a quote that has expired will produce an error. Not present if the `<quote>` failed.	C
side	string	The `<side>` specified in the request. Only present if the `<quote>` failed.	C
amount	float	The `<amount>` specified in the request. Only present if the `<quote>` failed.	C
error	error	Any errors that occurred while trying to process the request. Only present if the `<quote>` failed. Can include multiple instances of this object.	C

Response sample

```
<?xml version="1.0"?>
<response>
    <atomic>true</atomic>
    <quote>
        <success>true</success>
        <tradeid>APY_demo_14e8c34968fcee</tradeid>
        <buycurr>GBP</buycurr>
        <sellcurr>EUR</sellcurr>
        <buyamt>123.45</buyamt>
        <sellamt>144.48</sellamt>
        <rate>0.85443633</rate>
        <settlementdate>2011-10-12</settlementdate>
        <expiry>10</expiry>
    </quote>
</response>
```

Payment

This message will process a trade and instruct payments out to nominated beneficiaries. This can either be called on a <quote>d rate by specifying a <tradeid> or standalone without a <tradeid> ("trade implied"). If processed as a trade implied payment, the trade will be processed at the current rate regardless of what it is. Processing a payment enters the account holder into a contractual agreement with World First to settle any outstanding funds by the <settlementdate>.

Request

Payment request element

Tag	Type	Description	Usage
tradeid	string	A <tradeid> returned by a valid <quote> response. If omitted the trade will be processed at the current rate ("trade implied").	O
paymentid	string	Client specified unique identifier for this <payment>. This can be used later to reference previously processed <payment>s.	M
sellcurrency	string	The currency being sold to World First. If omitted, the sell currency will default to the <default-sell-currency> visible in <client-details>. Specified as an ISO-4217 alpha 3 code.	O
amount	float	The amount to credit to the <beneficiary> specified in the currency of the <beneficiary>. This will also be the same as the <buycurr> of the trade. The <amount> of all <payment>(s) must total to the amount of the referenced <quote> (if any).	M
paymentdate	date	The date for the <payment> to be made on. This must be on or after the <settlementdate> of the trade (if <quote>d) and will only be paid out once the trade has been settled.	M
notes1	string	Client specified notes to appear on the <payment> record.	O
notes2	string	Client specified notes to appear on the <payment> record.	O
reason	integer	Reason for <payment>. Must be one of the payment reason codes specified in the below table. If <reason> is 6 (Other), <reason_if_other> must be specified.	M
reason_if_other	string	Indicates reason for <payment> if <reason> is 6 (Other).	C
beneficiary	beneficiary	A <beneficiary> request element as defined in the blow table. This is the details of the account the <payment> is to be made to.	M

Payment reason codes

These are the acceptable codes allowed for the `<reason>` element of a `<payment>`.

Code	Meaning
1	Emigration
3	Overseas Mortgage Payments
4	Sending Money Home
5	Transfer to Own Account
6	Other
7	Paying Overseas Suppliers
8	Repatriating Overseas Earnings
9	Investing Abroad
16	Holiday Home/Second Home Purchase
17	Investment Property Purchase
18	Overseas Purchase
19	Property Sale
20	Returning From Abroad

Beneficiary request element

Tag	Type	Description	Usage
accholder	string	The name of the beneficiary/account holder.	M
accholderadd1	string	Address of the beneficiary.	M
accholderadd2	string	Address of the beneficiary.	O
accholderadd3	string	Address of the beneficiary.	O
curr	string	The currency of the beneficiary account. Specified as an ISO-4217 alpha 3 code.	M
bankname	string	Name of the beneficiary bank.	M
bankcode	string	Bank Code/Sort Code. Subject to beneficiary rules to determine whether required or not.	C
accno	string	Bank account number of the beneficiary. Subject to beneficiary rules to determine whether required or not.	C
bankadd1	string	Address of the bank.	O
bankadd2	string	Address of the bank.	O
bankadd3	string	Address of the bank.	O
bankcountry	string	Country of the bank. Specified as an ISO-3166 alpha 2 code.	M
notes	string	Notes pertaining to the beneficiary about the payment to them.	O
iban	string	IBAN identifier for this account. Subject to beneficiary rules to determine whether required or not.	C
bic	string	BIC/SWIFT identifier for the beneficiary bank. Subject to beneficiary rules to determine whether required or not. Specified as an ISO-9362 code.	C

Request sample

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <atomic>true</atomic>
    <payment>
        <tradeid>APY_demo_14e8c34968fcee</tradeid>
        <paymentid>1317816643-30</paymentid>
        <sellcurrency>EUR</sellcurrency>
        <amount>123.45</amount>
        <paymentdate>2011-10-13</paymentdate>
        <notes1>Tea re-supply</notes1>
        <notes2>Purchase Order: 39824</notes2>
        <reason>6</reason>
        <reason_if_other>Emergency tea re-stock</reason_if_other>
        <beneficiary>
            <accholder>Twinings</accholder>
            <accholderadd1>South Way</accholderadd1>
            <accholderadd2>Andover</accholderadd2>
            <accholderadd3>SP10 SAQ</accholderadd3>
            <curr>GBP</curr>
            <bankname>Lloyds</bankname>
            <bankcode>30-90-21</bankcode>
            <accno>87654321</accno>
            <bankadd1>Church Street, Amesbury</bankadd1>
            <bankadd2>Salisbury, Wiltshire</bankadd2>
            <bankadd3>SP4 7EZ</bankadd3>
            <bankcountry>GB</bankcountry>
            <notes>Batch 17816643</notes>
            <bic>LOYDGB2L</bic>
        </beneficiary>
    </payment>
</request>
```

Response

Payment response element

Tag	Type	Description	Usage
success	boolean	Indicates if the request was successfully processed.	M
tradeid	string	The <trade-confirmation-id> of the <trade-confirmation> this <payment> is consuming funds from. Not present if the <payment> failed and not specified in the request.	C
paymentid	string	Client specified unique identifier for this <payment>. This can be used later to reference previously processed <payment>s.	M
buycurr	string	The currency being bought by the account holder (currency transferred from World First to client nominated <beneficiary>). Specified as an ISO-4217 alpha 3 code. Not present if the <payment> failed.	C
sellcurr	string	The currency being sold to World First. Specified as an ISO-4217 alpha 3 code. Not present if the <payment> failed.	C
amount	float	The amount being credited to the <beneficiary> specified in the currency of the <beneficiary>. This will also be the same as the <buycurr> of the trade. Not present if the <payment> failed.	C
rate	float	The rate used to convert <amount> given in the request. See Appendix B for trading symbol calculation. Not present if the <payment> failed.	C
paymentdate	date	The date for the <payment> to be made on. This must be on or after the <settlementdate> of the trade and will only be paid out once the trade has been settled. Not present if the <payment> failed.	C
payment-fee	float	The fee that applies to this <payment> specified in the <sellcurr> of the trade. Not present if the <payment> failed.	C
error	error	Any errors that occurred while trying to process the request. Only present if the <payment> failed. Can include multiple instances of this object.	C

Trade confirmation response element

Tag	Type	Description	Usage
success	boolean	Indicates if the request was successfully processed.	M
trade-confirmation-id	string	The `<tradeid>` of the `<payment>` this `<trade-confirmation>` is funding.	M
settlement-amount	float	The amount required to settle this trade. The `<settlement-amount>` is inclusive of the `<deposit-amount>`.	M
settlement-date	date	Date the `<settlement-amount>` is due to World First.	M
deposit-amount	float	The amount required as a deposit on this trade.	M
deposit-date	date	Date the `<deposit-amount>` is due to World First.	M
settlement-method	string	Can be one of 'DirectDebit' or 'Bank'. If 'Bank' then funds will need to be instructed to World First with enough time to clear by the `<settlement-date>`. Not present if the trade failed.	C
wf-account-name	string	Name of settlement account. Not present if the trade failed.	C
wf-account-currency	string	Currency of settlement account. Not present if the trade failed.	C
wf-account-address1	string	Address of settlement account. Not present if the trade failed.	C
wf-account-address2	string	Address of settlement account. Not present if the trade failed.	C
wf-account-address3	string	Address of settlement account. Not present if the trade failed.	C
wf-bank-name	string	Name of settlement bank. Not present if the trade failed.	C
wf-bank-address1	string	Address of settlement bank. Not present if the trade failed.	C
wf-bank-address2	string	Address of settlement bank. Not present if the trade failed.	C
wf-bank-address3	string	Address of settlement bank. Not present if the trade failed.	C
wf-iban	string	International Bank Account Number for the settlement account. Subject to beneficiary rules to determine whether required or not. Specified as an ISO-13616 code. Not present if the trade failed.	C
wf-bic	string	BIC/SWIFT identifier for the settlement bank. Subject to beneficiary rules to determine whether required or not. Specified as an ISO-9362 code. Not present if the trade failed.	C
wf-account-number	string	The account number of the settlement account. Subject to beneficiary rules to determine whether required or not. Not present if the trade failed.	C
wf-bank-code	string	Sort/Routing/Transit/BSB Number/Code of the settlement bank. Subject to beneficiary rules to determine whether required or not. Not present if the trade failed.	C
error	error	Any errors that occurred while trying to process the request. Only present if the `<trade-confirmation>` failed. Can include multiple instances of this object. Only present if the trade failed.	C

Response sample

```
<?xml version="1.0"?>
<response>
        <atomic>true</atomic>
        <payment>
                <success>true</success>
                <tradeid>APY_demo_14e8c34968fcee</tradeid>
                <paymentid>1317816643-30</paymentid>
                <buycurr>GBP</buycurr>
                <sellcurr>EUR</sellcurr>
                <amount>123.45</amount>
                <rate>0.85443633</rate>
                <paymentdate>2011-10-13</paymentdate>
                <payment-fee>2.27</payment-fee>
        </payment>
        <trade-confirmation>
                <success>true</success>
                <trade-confirmation-id>APY_demo_14e8c34968fcee</trade-
confirmation-id>
                <settlement-amount>146.75</settlement-amount>
                <settlement-date>2011-10-13</settlement-date>
                <deposit-amount>7.22</deposit-amount>
                <deposit-date>2011-10-12</deposit-date>
                <settlement-method>Bank</settlement-method>
                <wf-account-name>World First UK Ltd</wf-account-name>
                <wf-account-currency>EUR</wf-account-currency>
                <wf-account-address1>Regent House, 16-18 Lombard Road</wf-
account-address1>
                <wf-account-address2>London SW11 3RB</wf-account-address2>
                <wf-account-address3>UK</wf-account-address3>
                <wf-bank-name>Barclays Bank (Docklands Branch)</wf-bank-name>
                <wf-bank-address1>One Churchill Place</wf-bank-address1>
                <wf-bank-address2>London </wf-bank-address2>
                <wf-bank-address3>BX3 2BB</wf-bank-address3>
                <wf-iban>GB61 BARC 202646 49244077</wf-iban>
                <wf-bic>BARCGB22</wf-bic>
                <wf-account-number></wf-account-number>
                <wf-bank-code></wf-bank-code>
        </trade-confirmation>
</response>
```

Past payments

Request by id

If you need to query details of a previously processed payment, this can be done by using the past payment request elements. All that is required in the request is the `<paymentid>` of the `<payment>` originally sent to AutoPay. The number of past-payment elements in the request that can be included is presently unlimited.

Request element

Tag	Type	Description	Usage
paymentid	string	Client specified unique identifier for the `<payment>` to retrieve.	M

Request sample

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <atomic>true</atomic>
    <past-paymentid>1317816643-30</past-paymentid>
</request>
```

Request by range

Alternatively, `<past-payment>`s may be queried by date range. This requires the passing of a `<startdate>` element and an `<enddate>` element. The dates specified reference the `<paymentdate>` element of the `<payment>`.

When this request type is made, all previous `<payment>`s with payment dates inclusively between the dates specified will be returned.

This request element type can only be made once per API call.

Request element

Tag	Type	Description	Usage
startdate	date	Starting date (inclusive) to match against `<paymentdate>` for the `<payment>s` to retrieve.	M
enddate	date	Ending date (inclusive) to match against `<paymentdate>` for the `<payment>s` to retrieve.	M

Request sample

```
<?xml version="1.0" encoding="utf-8"?>
<request>
    <atomic>true</atomic>
    <past-payment-range>
        <startdate>2011-10-13</startdate>
        <enddate>2011-10-14</enddate>
    </past-payment-range>
</request>
```

Response

Response elements

The response contains both `<past-payment>`s and `<trade-confirmation>`s as seen in a `<payment>` response. For the definition of a `<trade-confirmation>` please see the `<payment>` response section.

Tag	Type	Description	Usage
requested-paymentid	string	Client specified unique identifier for this `<payment>`.	M
success	boolean	Indicates if the request was successfully processed.	M
created	datetime	The date and time the `<payment>` was created. Not present if the request failed.	C
tradeid	string	The `<trade-confirmation-id>` of the `<trade-confirmation>` this `<payment>` is consuming funds from. Not present if the request failed.	C
buycurr	string	The currency being bought by the account holder (currency transferred from World First to client nominated `<beneficiary>`). Specified as an ISO-4217 alpha 3 code. Not present if the request failed.	C
sellcurr	string	The currency being sold to World First. Specified as an ISO-4217 alpha 3 code. Not present if the request failed.	C
amount	float	The amount being credited to the `<beneficiary>` specified in the currency of the `<beneficiary>`. This will also be the same as the `<buycurr>` of the trade. Not present if the request failed.	C
rate	float	The rate used to convert `<amount>` given in the request. See Appendix B for trading symbol calculation. Not present if the request failed.	C
paymentdate	date	The date for the `<payment>` to be made on. This must be on or after the `<settlementdate>` of the trade and will only be paid out once the trade has been settled. Not present if the request failed.	C
payment-fee	float	The fee that applies to this `<payment>` specified in the `<sellcurr>` of the trade. Not present if the request failed.	C
error	error	Any errors that occurred while trying to process the request. Only present if the request failed. Can include multiple instances of this object.	C

Response sample

```
<?xml version="1.0"?>
<response>
    <atomic>true</atomic>
    <past-payment>
        <requested-paymentid>1317816643-30</requested-paymentid>
        <success>true</success>
        <created>2011-10-05T12:10:45Z</created>
        <tradeid>APY_demo_14e8c49434e365</tradeid>
        <buycurr>GBP</buycurr>
        <sellcurr>EUR</sellcurr>
        <amount>123.45</amount>
        <rate>0.854436</rate>
        <paymentdate>2011-10-13</paymentdate>
        <payment-fee>2.27</payment-fee>
    </past-payment>
    <trade-confirmation>
        <success>true</success>
        <trade-confirmation-id>APY_demo_14e8c34968fcee</trade-confirmation-id>
        <settlement-amount>146.75</settlement-amount>
        <settlement-date>2011-10-13</settlement-date>
        <deposit-amount>7.22</deposit-amount>
        <deposit-date>2011-10-12</deposit-date>
        <settlement-method>Bank</settlement-method>
        <wf-account-name>World First UK Ltd</wf-account-name>
        <wf-account-currency>EUR</wf-account-currency>
        <wf-account-address1>Regent House, 16-18 Lombard Road</wf-account-
address1>
        <wf-account-address2>London SW11 3RB</wf-account-address2>
        <wf-account-address3>UK</wf-account-address3>
        <wf-bank-name>Barclays Bank (Docklands Branch)</wf-bank-name>
        <wf-bank-address1>One Churchill Place</wf-bank-address1>
        <wf-bank-address2>London </wf-bank-address2>
        <wf-bank-address3>BX3 2BB</wf-bank-address3>
        <wf-iban>GB61 BARC 202646 49244077</wf-iban>
        <wf-bic>BARCGB22</wf-bic>
        <wf-account-number></wf-account-number>
        <wf-bank-code></wf-bank-code>
    </trade-confirmation>
</response>
```

Appendix A – Error codes

All errors returned by AutoPay take the format described by the element definition below. Every error has a code and a corresponding error message. Each of these codes and messages are listed in the table below for reader convenience:

Error element

Tag	Type	Description	Usage
code	Integer	Code number of the `<error>` that occurred.	M
message	string	A human readable message describing the `<error>`.	M

Error codes

Code	Message
000	API Offline
100	Authentication Failed: User record not found in Database
101	Authentication Failed: User has been removed
102	Authentication Failed: Key is not set
103	Authentication Failed: Bad token
104	Authentication Failed: User disabled
105	Authentication Failed: No Back Office Account Number set
106	Authentication Failed: Unable to match Back Office Account number to any BO records
107	Authentication Failed: Some required user data for this user is missing or bad
200	Request XML not well-formed
201	World First Error: Cannot locate validation schema
202	Request XML does not validate against schema
250	Permissions Error: Client does not have GET QUOTE permission
251	Permissions Error: Client does not have TRADE IMPLIED PAYMENT permission
252	Permissions Error: Client does not have PAYMENT ON PREVIOUS QUOTE permission
253	Permissions Error: Client does not have PAST PAYMENTS VIA DATE RANGE permission
254	Permissions Error: Client does not have permission to book forwards past date that would incur a deposit
300	Bad value date, general
301	Value date must be set in future
302	Quote request amount must be greater than zero
303	Mismatched currencies
304	Same currencies requested in buy/sell
305	Value date breaches forward value date limit
306	Quote requested on a non-working date
500	Past Payment Not Found
900	AutoPay Database connection error
901	Failed selecting AutoPay database
902	Query on AutoPay database has failed
910	Back Office database connection error

911	Failed selecting Back Office database
912	Query on Back Office database has failed
920	Common Resource Database connection error
921	Failed selecting common resource database
922	Query on Common resource database has failed
400	Duplicate payment IDs found
401	Beneficiary not found in Payment
402	Payment Amount must be greater than zero
403	Invalid Payment Reason ID
404	If Payment Reason ID == 6 (other), a 'reason_if_other' must be given
405	Payment Date is in the past
406	Payment Date falls on a weekend, please ensure payment dates are set for weekdays
407	The Payment Date chosen falls on a UK or US non-working public holiday
408	Payment date breaches forward value date limit
450	Payment attempted on an expired quote - quote expired
451	The accrued total of payments for this trade does not match the amount that was quoted
452	The user sell currency does not match the Sell Currency specified in the quote
453	The payment/beneficiary buy currency does not match the Buy Currency specified in the quote
454	The referenced trade has already been previously covered
455	The referenced trade is currently unavailable
456	The payment date specified is before the trade settlement date. Payment date must be on or after the trade settlement date
460	The trade ID associated with this payment did not match any quoted trades
461	A payment referencing this trade ID has failed, so the trade was cancelled
462	Payment not processed because atomic was on and there were 1 or more failed payments
463	Failed placing order for this quote
464	Payment aborted because "atomic" was set and other payments failed in this request
470	The trade associated with this payment has breached the limit on the amount that can be traded on a single trade for this client
471	The limit on how much can be traded by this client in a 24 hour period has been breached
472	There are too many payments in this request
600	Currencies specified did not resolve to a valid trading symbol
601	Account Holder name is not set <accholder>
602	Account Holder Name can only be 34 characters in length - length exceeded
603	Account Holder address was not found in the beneficiary node
604	Account Holder address may only have alpha-numeric characters, spaces and these symbols: . - = + \' , : ( )
605	Bank Address was not found in the beneficiary node
606	Bank Address may only have alpha-numeric characters, spaces and these symbols: . - = + \' , : ( )
607	BIC/SWIFT numbers must be either 8 or 11 characters in length
608	BIC/SWIFT numbers may only have alpha-numeric characters, spaces and these symbols: . - = + \' , : ( )
609	UK: Either a Sort Code or a BIC must be provided
610	UK: If a BIC is provided, an IBAN or an account number must also be provided
611	Missing IBAN
612	Invalid IBAN
613	Sort Code is missing
614	Invalid Sort Code
615	Missing Bank Code
616	Invalid Bank Code
617	Account Number is missing
618	Either a Bank Code or a BIC/SWIFT must be provided

800	Quote received but failed placing order on quote, trade not booked
801	[general connector error] - comes from rate provider
700	Unable to match trade confirmation to World First pay bank record
701	Query attempting to pull trade confirmations has failed

Appendix B – Legal trading symbols

When working with a currency pair e.g. EUR and GBP, its symbol can be presented as EUR/GBP. There is a standard for the order of the pair; GBP/EUR is not a legal symbol. The order for each symbol can be determined by a rank on each currency.

The table on the right lists the currencies in order of precedence. Since EUR comes before AUD the legal symbol would be EUR/AUD. Since NZD comes before USD the legal symbol would be NZD/USD and so on.

This becomes important when quoting rates as a quote for EUR/GBP is only ever quoted as how much GBP is 1 EUR worth? This is regardless of the direction of the trade (buying GBP selling EUR or buying EUR selling GBP).

The reason for this standard is to avoid any loss of precision when working with inverted rates. If EUR/GBP is quoted at 0.85364928 and the rate is inverted, the resulting rate is:

1.171441273868350243322409877742 8

With possibly infinite decimal places to follow (the above number is rounded).

If an amount is then multiplied by the inverted rate the result may not be the same as the amount divided by the actual rate (0.85364928). Therefore it is common practice in foreign exchange calculations to never invert a rate and always work with rates represented the same way (hence the legal symbols).

It is possible to use AutoPay without performing any rate calculation on as both the buy and sell amounts are provided in quotes, payments and trade responses. This information is only provided as a warning if you intend to calculate amounts from quoted rates yourself.

EUR ↓
GBP
AUD
NZD
BWP
USD
CAD
CHF
SGD
DKK
ZAR
HKD
NOK
TRY
PLN
AED
SEK
BBD
BHD
CYP
CZK
ILS
INR
ISK
JMD
JOD
KES
KWD
MAD
MTL
MUR
MXN
OMR
QAR
SAR
THB
TND
TZS
XCD
SKK
CNY
EGP
FJD
MYR
PHP
PKR
TTD
JPY
HUF
RON
BGN
BDT
XAF
EEK
LTL
HRK
LVL
LKR
UGX

Appendix C – Supported currency pairs

The following is a list of currency pairs that are supported by World First AutoPay that can be used to transfer money from one to the other and reverse.

AUD/CAD	EUR/CHF	GBP/JPY	NZD/USD	GBP/CZK	AUD/PLN
AUD/CHF	EUR/GBP	GBP/NZD	USD/CAD	GBP/NOK	EUR/DKK
AUD/JPY	EUR/HKD	GBP/SGD	USD/CHF	GBP/SEK	USD/DKK
AUD/SGD	EUR/JPY	GBP/USD	USD/JPY	GBP/PLN	GBP/DKK
AUD/USD	EUR/NZD	GBP/ZAR	USD/SGD	EUR/CZK	AUD/DKK
CAD/CHF	EUR/SGD	HKD/JPY	USD/ZAR	EUR/NOK	NZD/DKK
CAD/JPY	EUR/USD	NZD/CAD	AUD/NZD	EUR/SEK	USD/HKD
CHF/HKD	GBP/AUD	NZD/CHF	USD/CZK	EUR/PLN	
CHF/JPY	GBP/CAD	NZD/HKD	USD/NOK	AUD/CZK	
CHF/SGD	GBP/CHF	NZD/JPY	USD/SEK	AUD/NOK	
EUR/AUD	GBP/HKD	NZD/SGD	USD/PLN	AUD/SEK	