small world
Money Transfer

API Service specification

www.smallworldfs.com

Fast Easy Secure

Table of contents

Security features 3

Environments 3

Flow overview to import transactions to Small World 4

Transaction created in Small World 4

Transaction created and requested to be voided 4

Additional services 5

calculate 5

addTransactionNote 6

getMtnByOriginalId 6

getTransactionsByChangeDate 6

getTransactionsByCloseDate 6

cancelTransaction 6

Customer creation and modification process 6

getClientIdByOriginalId 7

getOccupations 7

Compliance procedure 8

Compliance issues workflow 8

Compliance requirements 9

Import documents through web service 12

Statements 12

Transaction Status 13

In progress - Received 13

In progress - Inquiry 13

Other status 14

Web Service objects 14

importTransaction 14

TransactionImport 14

TransactionImportClient 16

TransactionImportClientId 18

TransactionImportClientBank 19

calculate 19

TransactionCalculationResult 20

acknowledgeTransaction 20

addTransactionNote 20
cancelTransaction 20
getMtnByOriginalId 21
getTransactionsByChangeDate 21
TransactionMtnAndStatus 21
getTransactionsByCloseDate 22
importClient 22
getClientByOriginalId 23
getOccupations 23
Occupation 23
importAttachment 23
Attachment 24
AttachmentId 24
getFolioNumber 24
getStatements 25
ReportStatementResult 25
Examples **26**
ImportTransaction example Request - Bank deposit 26
ImportTransaction example Response - Bank deposit 28
ImportTransaction example Request - Pick Up 30
ImportTransaction example Response - Pick Up 32
Import Client example Request 34
Import Client example Response 35
Mobile Wallet example Request 36
Attachment example Request 38
Calculate example Request 38
Calculate example Response 39
getFolio example Request 40
FAQ **41**

Security features

A certificate will be provided to access our service, together with credentials (username and password). Please install this certificate in your browser and access to our end point to confirm our wsdl is correctly displayed.
Password must be encrypted with SHA1 algorithm.

Java example:

```
public static String makeSHA1Hash(String input) {
  try {
             MessageDigest md = MessageDigest.getInstance("SHA1");
             md.reset();
             md.update(input.getBytes());
             byte[] digest = md.digest();

             String hexStr = "";
             for (int i = 0; i < digest.length; i++) {
             hexStr += Integer.toString((digest[i] & 0xff) + 0x100, 16).substring(1);
             }
             return hexStr;
  } catch (NoSuchAlgorithmException e) {
             return null;
  }}
```

Environments

Test environment - End Point
https://ua.oswfs.com/ImportExport/SimpleTransactionImportService?WSDL
Live environment - End Point
https://main.oswfs.com/ImportExport/SimpleTransactionImportService?WSDL

Please find associated *XSD* in *WSDL*.

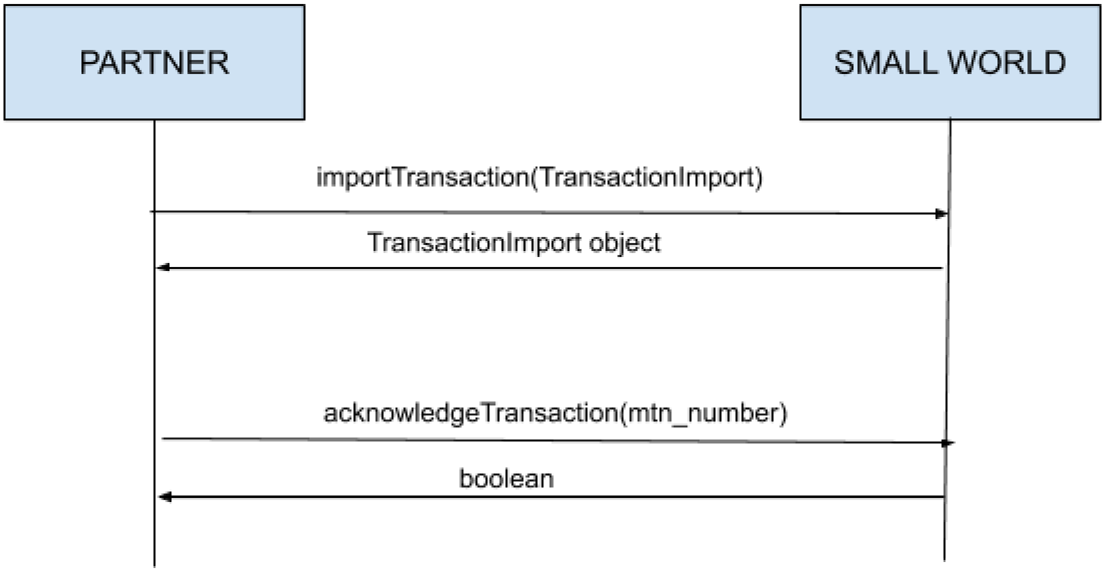
IP WHITELIST
In case IP needs to be whitelisted in partner's system, please take those into consideration:

195.20.223.132
195.110.76.229

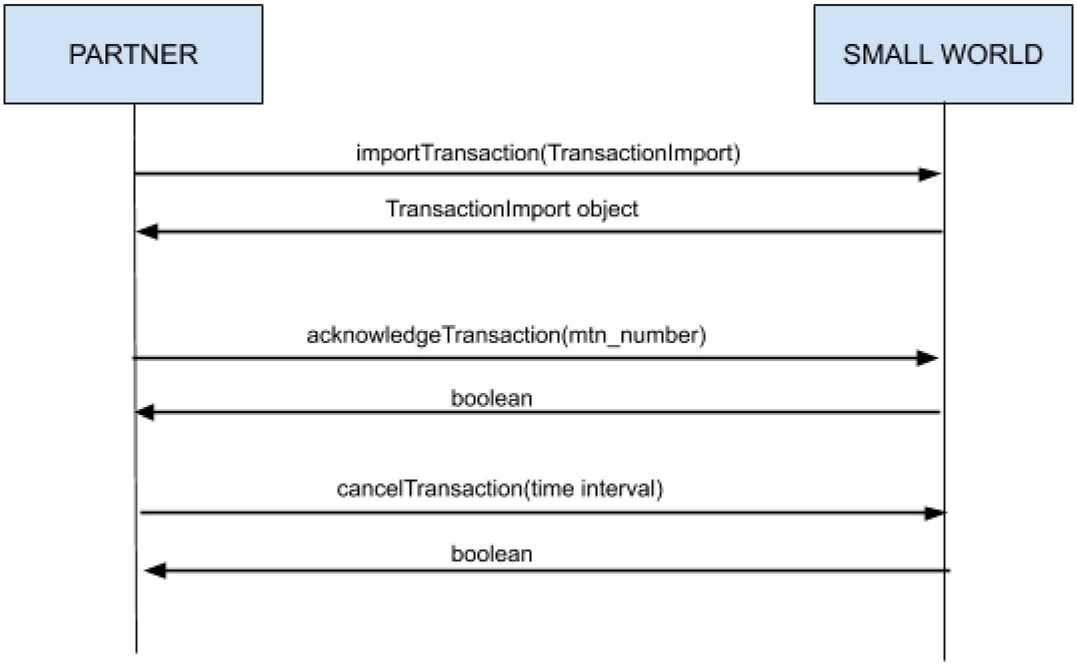
Please never use IP to connect to our service but hostname accordingly.

Flow overview to import transactions to Small World

1) Transaction created in Small World


PARTNER
SMALL WORLD
importTransaction(TransactionImport)
TransactionImport object
acknowledgeTransaction(mtn_number)
boolean

2) Transaction created and requested to be voided


PARTNER
SMALL WORLD
importTransaction(TransactionImport)
TransactionImport object
acknowledgeTransaction(mtn_number)
boolean
cancelTransaction(time interval)
boolean

In order to import transactions to Small World, you need to invoke two different services:

1) ***importTransaction***

Parameters: *TransactionImport* object (see attributes in section "Web service objects"). This service allows to send all information related to a specific transaction. Once the order is successfully created, it can be identified by its *MTN* (money transfer number) provided in the response.
Returns: *TransactionImport* object, with all the information related to the order created in Small World's system.

2) acknowledgeTransaction
Parameters: *MTN* and credentials. This service confirms the creation of the transaction in Small World's system. It should be called immediately after *importTransaction*.
Returns: *boolean*

Additional services

1) calculate
Parameters: *TransactionImport* object. This service provides information of the figures related to the transaction, such as fees, rates or payout amount (see attributes in section "Web service objects").
Returns: Array of *TransactionCalculation* objects (see attributes in section "Web service objects").

2) addTransactionNote
Parameters: credentials, *MTN* and note. This service allows to add a note to an imported transaction.
Returns: *boolean*

3) getMtnByOriginalId
Parameters: credentials and originalId. *MTN* number is provided given specific originalId of the transaction.
Returns: *MTN* number.

4) getTransactionsByChangeDate
Parameters: credentials and time interval. Any changes occurred during the specified time interval are reflected in the response of this service.
Returns: TransactionMtnAndStatus object (see attributes in section "Web service objects").

5) getTransactionsByCloseDate
Parameters: credentials and time interval. This service provides all transactions with a final status in the time frame specified in the request.
Please check possible status in "Transaction Status" section.
Returns: TransactionMtnAndStatus object (see attributes in section "Web service objects").

6) cancelTransaction
This method is used to ask for cancellation for a certain transaction previously inserted.

Parameters: credentials and *MTN* to be cancelled.

Returns: *boolean. True* will be returned if transaction has not been processed at payer side. If *False*, it doesn't necessarily mean that cancellation request has failed, but some time has to pass to request cancellation to payer correspondingly. Later, when transaction status is requested, cancellation can be confirmed.

Customer creation and modification process

- Customers (sender or beneficiary) are created through *importTransaction* service, together with transaction's information.
  Customer's unique identification in Small World's system (*clientId*) will be provided in the response. We strongly recommend to identify customers by *originalId* in the request, in order to avoid duplications or for modifications (you may assign your internal customer identification to *originalId* field).

- In order to modify an existing customer, please invoke ***importClient*** service together with its corresponding *originalId*:
  Parameters: *TransactionImportClient* object (see attributes in "Web service objects" section), credentials and information related to the agent. Customer gets identified by its *clientId* provided in the response.
  Returns: *TransactionImportClient* object.

Additional services for Customers' process:

1) getClientIdByOriginalId
Parameters: credentials and *originalId* of the customer.
Returns: *clientId*

2) getOccupations
Parameters: username and password. This service provides a list of occupations per country if need be.
Returns: array of *Occupation* objects.

Please take into consideration these comments:

1) It is highly recommended the usage of the field "originalId" to identify all objects in our system. **This field allows to detect duplicated objects mistakenly imported**.
2) Given the large amount of information related to our payout network, it is not technically reasonable to provide such details through our web service.
   If your payout network available is not limited to a few locations , you will be provided a *sftp* access with files containing all necessary information on a daily basis.
3) Like payout network, bank branches available per countries can be found under the same *sftp* folder on a daily basis.

4) Maximum of 7 calculation results in calculate service's response. The more information provided, the more accurate the calculation will be.
5) Time interval in getTransactionByChangeDate or getTransactionByCloseDate is 1 hour maximum.
6) *getTransactionsByChangeDate* won't return any changes in specific time frame when other updates have been taken place afterwards.
   In order to have orders correctly updated in partner's system we strongly recommend to call this service frequently.
7) Countries and currencies must be indicated with *ISO3* code.
8) In order to identify the transaction, beneficiary needs to provide *folio* number. This reference number, field named "*folio*", is obtained in the response of *importTransaction* once the order is created.
   In case this parameter is empty, *MTN* shall be enough to identify the transaction.

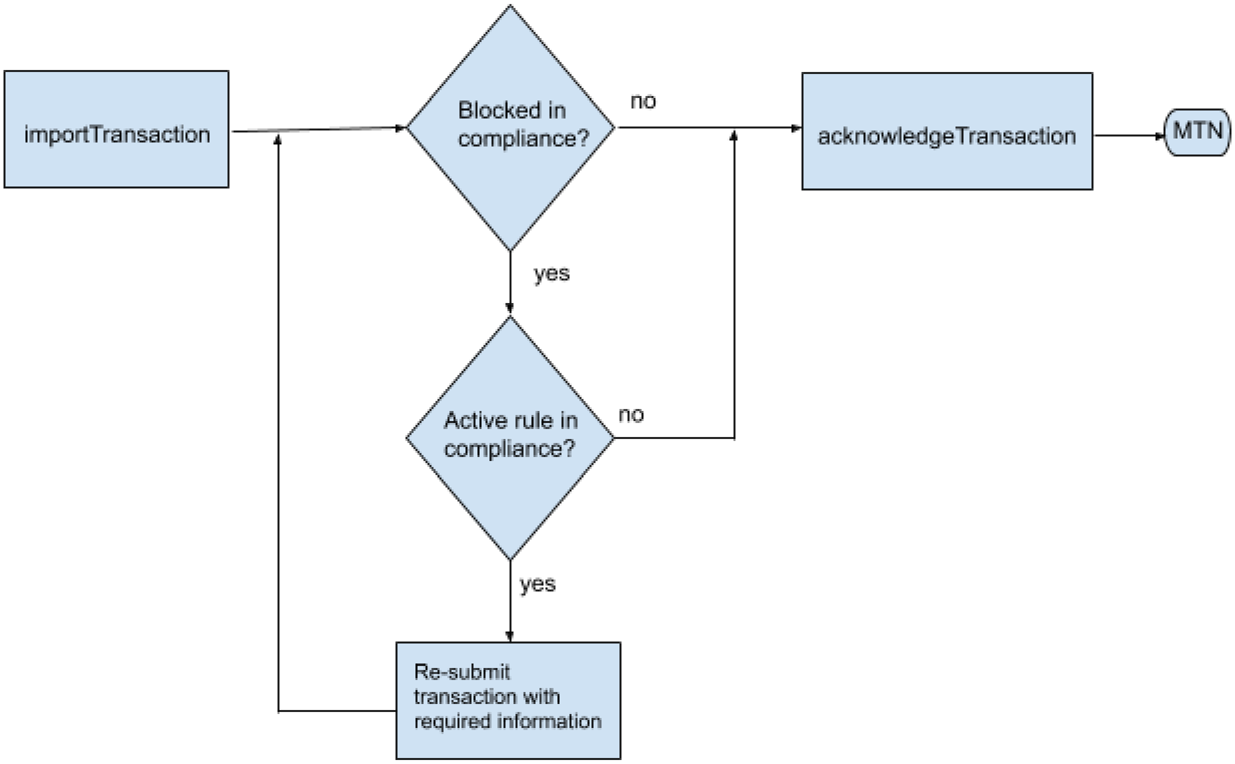
9) Some partners in our payout network have their own remittance number format.
   If you need to know this folio before the transaction is created, you can use our service ***getFolioNumber*** to reserve it.
   Parameters: sending date, agent code, original Id of the transaction, payer representative code and credentials.
   Returns: folio reference
   Please send this folio reference in your *importTransaction* request. Let us know if this functionality is needed.

Compliance procedure

Small World is committed to operating business in a transparent and open manner that is consistent with regulatory obligations. Our compliance department will always ensure that all suspicious activities are reported to the relevant authorities. It is our policy that commercial considerations shall never take precedence over our AML commitments.

Compliance issues workflow

Please check the diagram below. Small World's system has a compliance module with a set of rules that might be triggered when a transaction is intended to be created.
These rules are closely related to sending countries and amounts aggregated.



There are two different types of rules:

1) ***Blocking***: the order can not be acknowledged until all required information is provided.
2) ***Non blocking***: a Compliance Officer will review transaction's details once the order is acknowledged by the agent (*acknowledgeTransaction* service).

Compliance requirements

There are several types of compliance requirements (check XSD linked to WSDL); details are provided in the response of *importTransaction* service.
The most significant are the following:

1) Purpose of the transaction, client relation, source of funds or ID missing or expired.

Source of Funds

```
<complianceDetails>
    <blocking>true</blocking>
    <id>*</id>
    <reason>SPECIFY_SOURCE_OF_FUND</reason>
    <reasonTranslated>Specify Source Of Fund</reasonTranslated>
    <ruleId>*******</ruleId>
```

```
        <ruleType>COMPLIANCE</ruleType>
        <status>ACTIVE</status>
        <type>SOURCE_REQUIRED</type>
    </complianceDetails>
```

Purpose

```
    <complianceDetails>
        <blocking>true</blocking>
        <id>*</id>
        <reason>SPECIFY_PURPOSE</reason>
        <reasonTranslated>Specify Purpose</reasonTranslated>
        <ruleId>******</ruleId>
        <ruleType>COMPLIANCE</ruleType>
        <status>ACTIVE</status>
        <type>PURPOSE_REQUIRED</type>
    </complianceDetails>
```

Client Relation with the beneficiary

```
    <complianceDetails>
        <blocking>true</blocking>
        <id>*</id>
        <reason>CLIENT_RELATION</reason>
        <reasonTranslated>Client Relation</reasonTranslated>
        <ruleId>*****</ruleId>
        <ruleType>COMPLIANCE</ruleType>
        <status>ACTIVE</status>
        <type>CLIENT_RELATION_REQUIRED</type>
    </complianceDetails>
```

Client ID document

```
    <complianceDetails>
        <blocking>false</blocking>
        <id>*</id>
        <idType>PASSPORT</idType>
        <idType>DRIVERS_LICENSE</idType>
        <idType>ALIENCARD</idType>
        <idType>CEDULA</idType>
        <idType>OTHER</idType>
         <reason>REQUIRED_ID_OF_TYPE__IS_MISSING</reason>
         <reasonTranslated>Required Id Of Type Passport, Drivers License, Registration Card, National Id
        Card, Other Is Missing</reasonTranslated>
        <ruleId>*******</ruleId>
        <ruleType>COMPLIANCE</ruleType>
        <status>ACTIVE</status>
        <type>ID_MISSING_OR_EXPIRED</type>
    </complianceDetails>
```

2) Compulsory documentation related to the customer or the transaction, such as proof of domicile, proof of funds, digital copy of the document…

```
<complianceDetails>
    <blocking>false</blocking>
    <formType>KYC</formType>
    <id>*</id>
    <reason>_NEEDS_TO_BE_ATTACHED</reason>
    <reasonTranslated>Kyc Needs To Be Attached</reasonTranslated>
    <ruleId>******</ruleId>
    <ruleType>COMPLIANCE</ruleType>
    <status>ACTIVE</status>
    <type>KYC_REQUIRED</type>
</complianceDetails>
```

In order to attach a document to a specific customer or transaction, you need to use service *importAttachment* (please check “Import documents” section).

3) Additional information about the sender or the beneficiary (digital copy is not required):

- Two Last names required
- Date of Birth required
- SSN/CPF/CNPJ/INPS required (Depending on a country. ex: CPF is only for Brazil)
- Country of Birth required
- City of Birth required
- Occupation required
- Phone required
- Nationality required
- Country of Residence
- IDs Expiration Date

For example, if nationality is required, the response would be the following:

```
<complianceDetails>
    <blocking>true</blocking>
    <clientExtraFields>NATIONALITY</clientExtraFields>
    <id>*</id>
    <reason>MISSING_DATA_</reason>
    <reasonTranslated>Missing Data : Nationality</reasonTranslated>
    <ruleId>******</ruleId>
    <ruleType>COMPLIANCE</ruleType>
    <status>ACTIVE</status>
    <type>MISSING_DATA</type>
  </complianceDetails>
```

In this case, just add nationality tag accordingly in *importTransaction’s* request.

Import documents through web service

This *api* has a separate *WSDL* End Point, available at:

Test environment:
https://ua.oswfs.com/ImportExport/SimpleTransactionImportAttachmentService?WSDL

Live environment:
https://main.oswfs.com/ImportExport/SimpleTransactionImportAttachmentService?WSDL

Main service available to attach documents to a customer / transaction is ***importAttachment***:
Parameters: credentials and *Attachment* object (check attributes in "Web service objects" section). This service allows to import a document associated to specific customer or transaction.
Returns: *AttachmentId* object (check attributes in "Web service objects section").

Compliance issue / AttachmentId fields to be populated / Attachment type relation:

Compliance Issue	**AttachmentId fields**	**Attachment type**
COMPLIANCE_FORM	mtn	COMPLIANCE_FORM
LETTER	clientId	LETTER
FAX_ATTACHMENT_NEEDS_TO_BE_ATTACHED	clientId	FAX_ATTACHMENT
KYC_REQUIRED	clientId	KYC
PROOF_OF_OCCUPATION	clientId	PROOF_OF_OCCUPATION
PROOF_OF_DOMICILE	clientId	PROOF_OF_DOMICILE
PROOF_OF_FUNDS	clientId	PROOF_OF_FUNDS
WIRE_CUSTOMER_PENDING	clientId	NON_FACETOFACE
NON_FACE_TO_FACE_FORM_PENDING	clientId	NON_FACETOFACE
DIGITAL_COPY_MISSING_FOR_PRIMARY_ID	clientId+clientIdId	FRONT
DIGITAL_COPY_MISSING_FOR_PRIMARY_ID	clientId+clientIdId	BACK

Statements

Statements within last 31 days are available in following *api* End Point:

Test environment:
https://ua.oswfs.com/ImportExport/ReportService?WSDL

Live environment
https://main.oswfs.com/ImportExport/ReportService?WSDL

Please find associated *XSD* in *WSDL*.

getStatements
Parameters: agent code, agent location code, agent operator, currency, time frame and credentials.
Returns: *ReportStatatementResult* array object (check “Web service objects” section).

Transaction Status

In progress - Received

Code	Description
AV	AVAILABLE FOR PICKUP (PICKUP TRANSACTION)
BP	BEING PROCESSED BY PAYER (BANK DEPOSIT)
CB	CONTACTED BENEFICIARY BY PHONE
LM	LEFT MESSAGE
NA	NOBODY ANSWERS THE PHONE
CY	DOCUMENTATION OK
OD	ORDER OUT FOR DELIVERY
AB	DELIVERY ATTEMPTED BENEFICIARY NOT AVAILABLE
TB	PHONE IS BUSY
AN	AVAILABLE FOR BENEFICIARY

In progress - Inquiry

Code	Description
WP	WRONG PHONE NUMBER
DP	TELEPHONE OUT OF ORDER
PW	WRONG PAYEE NAME
NR	NOBODY KNOWS THE SENDER
NB	NOBODY KNOWS THE BENEFICIARY
AC	AUTHORIZE BENEFICIARY NAME CHANGE
WA	WRONG BANK ACCOUNT NUMBER
WB	WRONG BANK NAME
MT	ORDER MISSING TELEPHONE
MA	ORDER MISSING ADDRESS
WO	BRANCH TEMPORARLY UNAVAILABLE
CN	INSUFFICIENT DOCUMENTATION
CD	CANCELLATION REQUEST DENIED
PD	CHECK POSSIBLE DUPLICATE
CR	UNDER COMPLIANCE REVISION

IA	WRONG ADDRESS
IC	WRONG CITY
BF	SEND ORDER AGAIN
XI	MISCELLANOUS
IB	INVALID BRANCH

Other status

Status	Code	Description
Paid out - closed status	YP	PAID
In progress - Void Request	RC	CANCELLATION REQUESTED
Order cancelled - closed status	OC	ORDER CANCELLED
Order expired - closed status	XX	EXPIRED ORDER

Web Service objects

importTransaction

Signature: *TransactionImport importTransaction*(*TransactionImport* transaction)
Parameters and associated objects: please check tables below

TransactionImport

Field Name	Type	Required	Validation	Description
originalId	Integer	No		A number that identifies the transaction, provided by your system.
mtn	Long	No****		A number that identifies the transaction provided by our system.
importStatus	Enum	No****		Status of the transaction import.
folioNumber	String	No****	max 25 chars	A textual number that identifies the transaction, it could be provided by your system.
paymentType	Enum	Yes		The type of payment used to pay the transaction.
deliveryMethod	Enum	Yes		The delivery method
sendingPrincipal	Double	Yes		The sending amount
totalSale	Double	Yes		The total sale amount
payoutPrincipal	Double	No****		The payout amount
fixedFee	Double	Yes		The fixed fee
variableFee	Double	No		The variable fee
creationDate	Timestamp	Yes	Must be in	The timestamp when the transaction was registered in

			the past	your system.
notes	String	No	max 300 chrs	Note attached to the transaction.
serviceMessage	String	No	max 300 chrs	Message attached to this transaction.
suspiciousActivityNotes	String	No	max 1000 chr	Suspicious Activity note attached to this transaction.
sendingCountry	String	Yes	3 chars	The ISO3 code of the sending country.
sendingCurrency	String	Yes	3 chars	The ISO3 code of the sending currency.
agentCode	Integer	Yes		The sending agent code.
agentLocationCode	String	Yes	max 25 chars	The sending agent location code.
agentOperator	String	Yes	max 20 chars	The sending agent operator username.
payoutCountry	String	Yes	3 chars	The ISO3 code of the payout country.
payoutCurrency	String	Yes	3 chars	The ISO3 code of the payout currency.
payerCode	Integer	Yes		The paymaster code.
payerLocationCode	String	Yes	max 25 chars	The paymaster location code.
depositBankId	Integer	No**		The bank identification number for the deposit.
depositBankName	String	No**	max 55 chars	The bank name for the deposit.
depositBankCode	String	No**	max 15 chars	The bank code for the deposit.
depositBankBranchId	Integer	No***		The bank's branch identification number for the deposit.
depositBankBranchName	String	No**	max 100 chrs	The bank's branch name for the deposit.
depositBankBranchCode	String	No**	max 12 chars	The bank's branch code for the deposit.
depositDate	Timestamp	No**		The date of the deposit
depositNotes	String	No**	max 100 chrs	The additional notes for the deposit.
sourceOfFunds	String	No*	max 250 chrs	The source of the funds.
purpose	Enum	No*		The purpose of the transaction.
promotionNumber	String	No		The promotion number/code.
promotionName	String	No****		The promotion name.
promotionAmount	Double	No****		The promotion discount amount.
clientRelation	Enum	No*		The relation with the beneficiary of the sender.
selectedPayoutClientBankOriginalId	Integer	No*****		If the beneficiary has multiple banks you can specify one with this.
selectedSendingClientBankOriginalId	Integer	No*****		If the sender has multiple banks you can specify one with this.
companyUserName	String	Yes	max 20 chars	The userName we provided to use this WebService.
companyPassword	String	Yes		The password we provided to use this WebService.
employer	TransactionImportEmployer	Yes		The information on the sending company of the transaction (for B2C, C2B, B2B transactions)
sender	TransactionImportClient	Yes		The information on the sender of the transaction
beneficiary	TransactionImport	Yes		The information on the beneficiary of the transaction

	Client			

* Can dynamically be set to required by compliance rules.
** Except for paymentType bank wire and bank deposit. (if you didn't choose a branch from our branches then branchCode should be provided)
*** Only required if the selected country bank has countryBankType FORCE_BRANCH
**** Will be populated in the response message.
***** Only required if the customer has multiple banks.

TransactionImportClient

Field Name	**Type**	**Required**	**Validation**	**Description**
originalId	Integer	No		A number that identifies the client, provided by your system.
clientId	Integer	No*8		A number that identifies the client in Small World's system.
firstName	String	Yes	min 2 chars max 50 chars	The first name
secondName	String	No	max 50 chars	The second or middle name
firstLastName	String	Yes	max 50 chars	The first lastname
secondLastName	String	No *4	max 50 chars	The second lastname
type	Enum	Yes		The type of client (only INDIVIDUAL is allowed).
address	String	Yes *1	max 255 chars	The address
zip	String	No	max 30 chars	The zip code or area code
city	String	No	max 50 chars	The city
state	String	No	max 100 chars	The state
country	String	Yes	3 chars	The ISO3 code of the client's country.
streetType	Enum	No *7		The street type.
streetName	String	No *7	max 255 chars	The street name.
streetNumber	Integer	No *7	max 12 chars	The street number.
gender	Enum	No		The gender
taxCode	String	No	max 50 chars	The italian tax code and The Norway tax code (NIN 11 characters)
mobilePhoneNumber	String	No	max 100 chars	The mobile phone number
email	String	No	max 100 chars	The email address
countryOfBirth	String	No *4	3 chars	The ISO3 code of the country of birth.
stateOfBirth	String	No *4	max 100 chars	The state of birth.
cityOfBirth	String	No *4	max 50 chars	The city of birth.
dateOfBirth	String	No *4	Must be in the past	The date of birth

nationality	String	No	3 chars	The ISO3 code of the client's nation.
idType	Enum	No *4		The type of ID
ssn	String	No *4	max 60 chars	The SSN (depending on the country: CPF, CNPJ, SSN, INPS)
occupationId	Integer	No *4		The occupation id provided, restricted by country.
idNumber	String	No *4	max 60 chars	The number of the ID
idIssuedBy	String	No *4	max 50 chars	The ID is issued by
idExpirationDate	Timestamp	No *4		The expiration date of the ID
idIssueCountry	String	No *4	3 chars	The country's ISO3 code of the ID
idIssueDate	Timestamp	No *4	Must be in the past	The issue date of the ID
ids	TransactionImportClientId[]			A list of IDs
bankAccountType	Enum	Yes *2		The type of the bank account
bankId	Integer	Yes *2		The SWFS country bank ID of the bank
bankName	String	Yes *2	max 55 chars	The name of the bank
bankCode	String	Yes *2	max 15 chars	The code of the bank
bankBranchId	Integer	Yes *5		The SWFS country bank branch ID of the bank's branch
bankBranchName	String	Yes *2	max 100 chars	The name of the bank's branch
bankBranchCode	String	Yes *2	max 12 chars	The code of the bank's branch
bankBic	String	Yes *3	8 chars	The bank account's BIC/SWIFT code
bankIban	String	Yes *3	max 35 chars	The bank account's IBAN code
bankRoutingNumber	String	No	9 chars	The bank routing number
bankAccountNumber	String	Yes *3	max 60 chars	The bank account number
banks	TransactionImportClientBank[]	No		A list of bank accounts
employerOriginalId	Integer	No *6		The originalId of the employer
employerId	Integer	No *6		The id of the employer
employeeType	Enum	No *6		The relation of the client with the employer
changeDate	Timestamp			The date of the last change

*1 Only required for senders or also for beneficiary when delivery method is physical delivery.
*2 Only required for beneficiary when delivery method is bank deposit.
*3 Either provide AccountNumber or both IBAN and BIC.
*4 Can dynamically be set to required by compliance rules.
*5 Only required if the selected country bank has countryBankType FORCE_BRANCH
*6 Only required if the client is used as part of a TransactionImportEmployer.
*7 Replaces address field in Spain, UK and Netherlands.
*8 Populated in the response.

TransactionImportClientId

Field Name	Type	Required	Validation	Description
originalId	Integer	No		A number that identifies the client, provided by your system.
clientIdId	Integer	No		A number that identifies the client, provided by SWFS system.
idType	Enum	Yes		The type of ID
idNumber	String	Yes	max 60 chars	The number of the ID
idIssuedBy	String	No	max 50 chars	The ID is issued by
idExpirationDate	Timestamp	Yes		The expiration date of the ID
idIssueCountry	String	No	3 Chars	The country's ISO3 code of the ID
idIssueState	String	No		The state's code of the ID
idIssueDate	Timestamp	No	Must be in the past	The issue date of the ID
changeDate	Timestamp	No		The date of the last change

TransactionImportClientBank

Field Name	Type	Required	Validation	Description
originalId	Integer	No		A number that identifies the client, provided by your system.
clientBankId	Integer	No		Number that identifies the client, provided by SWFS.
bankAccountType	Enum	Yes		The type of the bank account
bankId	Integer	No *3		The SWFS country bank ID of the bank
bankName	String	No *3	max 55 chars	The name of the bank
bankCode	String	No *3	max 15 chars	The code of the bank
bankBranchId	Integer	No *2		The SWFS country bank branch ID of the bank's branch
bankBranchName	String	Yes	max 100 chars	The name of the bank's branch
bankBranchCode	String	No	max 12 chars	The code of the bank's branch
bankBic	String	No *	8 or 11 chars	The bank account's BIC/SWIFT code
bankIban	String	No *	max 35 chars	The bank account's IBAN code
bankRoutingNumber	String	No	9 chars	The bank routing number
bankAccountNumber	String	No *	max 60 chars	The bank account number
changeDate	Timestamp	No		The date of the last change
providerRepCode	Integer	No		The RepCode where this Mobile Wallet Account is associated to
providerLocationCode	String	No		The LocationCode where this Mobile Wallet Account is associated to
bankAccountPrefix	String	No		Country phone prefix for the MW bank account. MW bank account phone has to belong to the same phone as the customer and as the payout location country (when the customer owning this bank account is the beneficiary)

* Either provide AccountNumber or both IBAN and BIC.
*2 Only required if the selected country bank has countryBankType FORCE_BRANCH
*3 Either provide bankId, bankName or bankCode.
*4 For MW accounts we need providerRepCode, providerLocationCode, bankName (to be verified), bankBranchName (to be verified), bankAccountType and bankAccount, the bankAccountPrefix won't be read in the requests only be served in the response.

calculate

Signature: *TransactionCalculationResult*[] *calculate*(*TransactionImport* transaction)
Parameters and associated objects: please check tables below

TransactionCalculationResult

Field Name	**Type**	**Required**	**Description**
payoutRepresentativeCode	Integer	No	A number that identifies the payer provided by our system.
rate	Double	No	
totalFee	Double	No	
principal	Double	No	
totalSale	Double	No	
payoutPrincipal	Double	No	

acknowledgeTransaction

Signature: *boolean acknowledgeTransaction*(long mtn, String companyUserName, String companyPassword)
Parameters and associated objects: please check tables below

Parameter Name	**Type**	**Required**	**Validation**	**Description**
mtn	Long	Yes		A number that identifies the transaction provided by our system.
companyUserName	String	Yes	max 20 chars	The userName we provided to use this WebService.
companyPassword	String	Yes		The password we provided to use this WebService.

addTransactionNote

Signature: *boolean addTransactionNote*(long mtn, String note, boolean export, String companyUserName, String companyPassword)
Parameters and associated objects: please check tables below

Parameter Name	**Type**	**Required**	**Validation**	**Description**
mtn	Long	Yes		A number that identifies the transaction provided by our system.

note	String	Yes	max 1000 chars	The note text.
companyUserName	String	Yes	max 20 chars	The userName we provided to use this WebService.
companyPassword	String	Yes		The password we provided to use this WebService.

cancelTransaction

Signature: *boolean cancelTransaction*(long mtn, String companyUserName, String companyPassword)
Parameters and associated objects: please check tables below

Parameter Name	Type	Required	Validation	Description
mtn	Long	Yes		A number that identifies the transaction provided by our system.
companyUserName	String	Yes	max 20 chars	The userName we provided to use this WebService.
companyPassword	String	Yes		The password we provided to use this WebService.

getMtnByOriginalId

Signature: Long *getMtnByOriginalId*(Integer originalId, String companyUserName, String companyPassword)
Parameters and associated objects: please check tables below

Parameter Name	Type	Required	Validation	Description
originalId	Integer	Yes		A number that identifies the transaction provided by your system.
companyUserName	String	Yes	max 20 chars	The userName we provided to use this WebService.
companyPassword	String	Yes		The password we provided to use this WebService.

getTransactionsByChangeDate

This method will show up transactions no matter which their current status is, made on between the dates provided as parameters. Note that the interval indicated within dates cannot be higher than 1 hour.
Signature: *TransactionMtnAndStatus*[] *getTransactionsByChangeDate*(Calendar fromDateTime, Calendar toDateTime, String companyUserName, String companyPassword)
Parameters and associated objects: please check tables below

Parameter Name	Type	Required	Validation	Description
fromDateTime	DateTime	Yes		Start date.
toDateTime	DateTime	Yes		Stop date.
companyUserName	String	Yes	max 20 chars	The userName we provided to use this WebService.
companyPassword	String	Yes		The password we provided to use this WebService.

TransactionMtnAndStatus

Field Name	Type	Required	Description
mtn	Long	No	A number that identifies the transaction provided by our system.
status	Enum	No	
originalId	Integer	No	A number that identifies the transaction, provided by your system.
payoutStatus	Enum	No	
progressStatus	Enum	No	
payoutNote	Double	No	
changeDate	Timestamp	No	
payoutDate	Timestamp	No	
closeDate	Timestamp	No	
agentMtn	Integer	No	
folioNumber	Integer	No	
pinNumber	Integer	No	
payerMtn	Integer	No	

getTransactionsByCloseDate

This method will show up all transactions with current status paid or cancelled, made on between the dates provided as parameters. Note that the interval indicated within dates cannot be higher than 1 hour.

Signature: *TransactionMtnAndStatus*[] *getTransactionsByCloseDate*(Calendar fromDateTime, Calendar toDateTime, String companyUserName, String companyPassword) \
\

Parameters and associated objects: please check tables below

Parameter Name	Type	Required	Validation	Description
fromDateTime	DateTime	Yes		Start date.
toDateTime	DateTime	Yes		Stop date.
companyUserName	String	Yes	max 20 chars	The userName we provided to use this WebService.
companyPassword	String	Yes		The password we provided to use this WebService.

importClient

Signature: *TransactionImportClient importClient*(*TransactionImportClient* client, Integer agentCode, String agentLocationCode, String agentOperator, String companyUserName, String companyPassword)
Parameters and associated objects: please check tables below

Parameter Name	Type	Required	Validation	Description
client	TransactionImport Client	Yes		The client information that you want to import.
agentCode	Integer	Yes		The code of the agent that is importing.
agentLocationCode	String	Yes		The code of the agent's location that is importing.
agentOperator	String	Yes		The username of the agent's operator that is importing.
companyUserName	String	Yes	max 20 chars	The userName we provided to use this WebService.
companyPassword	String	Yes		The password we provided to use this WebService.

getClientByOriginalId

Signature: Integer *getClientIdByOriginalId*(Integer originalId, String companyUserName, String companyPassword)
Parameters and associated objects: please check tables below

Parameter Name	Type	Required	Validation	Description
originalId	Integer	Yes		A number that identifies the transaction provided by your system.
companyUserName	String	Yes	max 20 chars	The userName we provided to use this WebService.
companyPassword	String	Yes		The password we provided to use this WebService.

getOccupations

Signature: *Occupation*[] *getOccupations*(String companyUserName, String companyPassword)
Parameters and associated objects: please check tables below

Parameter Name	Type	Required	Validation	Description
companyUserName	String	Yes	max 20 chars	The userName we provided to use this WebService.
companyPassword	String	Yes		The password we provided to use this WebService.

Occupation

Field Name	Type	Required	Validation	Description
occupationId	Integer	No		A number that identifies the occupation provided by our system.
country	String	No	3 chars	ISO3 country code

occupationName	String	No		
occupationClientType	Enum	No		

importAttachment

Signature: public *AttachmentId importAttachment*(Attachment attachment, String companyUserName, String companyPassword)
Parameters and associated objects: please check tables below

Parameter Name	Type	Required	Validation	Description
attachment	Attachment	Yes		
companyUserName	String	Yes	max 20 chars	The userName we provided to use this WebService.
companyPassword	String	Yes		The password we provided to use this WebService.

Attachment

Field Name	Type	Required	Validation	Description
attachmentId	AttachmentId	No		A number that identifies the attachment provided by our system.
name	String	No		A name for the attachment.
title	String	No		A title for the attachment.
attachment	byte[]	No		The attachment itself.
mimeType	String	No		The mime type of the attachment.
attachmentType	Enum	No		The type of the attachment.
verifiedFlag	Enum	No		The verification status for the attachment.

AttachmentId

Field Name	Type	Required	Validation	Description
attachmentId*	int	No		A number that identifies the attachment provided by our system.
mtn	long	No		A number that identifies the transaction provided by our system.
clientId	int	No		A number that identifies the client provided by our system.
clientIdId	int	No		A number that identifies the client's ID provided by our system.

* Will be populated in the response message.

getFolioNumber

Signature: String *getFolioNumber*(Calendar sendingDate, int sendingRepresentativeCode, int sendingRepresentativeOriginalTransactionId, int payoutRepresentativeCode, String companyUserName, String companyPassword)
Parameters and associated objects: please check tables below

Parameter Name	Type	Required	Validation	Description
sendingDate	DateTime	Yes		Sending date of the transtion to which this folio will be assigned.
sendingRepresentativeCode	int	Yes		Sending representative code for the agent of the transaction.
sendingRepresentativeOriginalTransactionId	int	Yes		The transaction number within the sending representative.
payoutRepresentativeCode	int	Yes		Payout representative code for the agent of the transaction.
companyUserName	String	Yes	max 20 chars	The userName we provided to use this WebService.
companyPassword	String	Yes		The password we provided to use this WebService.

getStatements

Signature: *ReportStatementResult*[] *getStatements*(Integer agentCode, String agentLocationCode, String agentOperator, String currencyIso3, Calendar fromDateTime, Calendar toDateTime, String companyUserName, String companyPassword)
Parameters and associated objects: please check tables below

Parameter Name	Type	Required	Validation	Description
agentCode	Integer	Yes		The agent code.
agentLocationCode	String	Yes		The location code.
agentOperator	String	Yes		The username of the agent operator.
currencyIso3	String	Yes	3 chars	The ISO3 code for a currency.
fromDateTime	String	Yes		The start time of the report.
toDateTime	String	Yes		The stop time of the report.
companyUserName	String	Yes	max 20 chars	The userName we provided to use this WebService.
companyPassword	String	Yes		The password we provided to use this WebService.

ReportStatementResult

Field Name	**Type**	**Required**	**Description**
representativeCode	Integer	No	A number that identifies the attachment provided by our system.
currencyIso3	String	No	A name for the attachment.
date	Calendar	No	The attachment itself.
type	Enum	No	The mime type of the attachment.
amount	Double	No	The type of the attachment.
balance	Double	No	The type of the attachment.
referenceMtn	Long	No	The type of the attachment.

Examples

ImportTransaction example Request - Bank deposit

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:web="http://webservice.ie.smallworldfs.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:importTransaction>
      <arg0>
        <companyUserName>apiXXXXX</companyUserName>
        <companyPassword>PASSWORD SHA1 ENCRYPTED HERE</companyPassword>
        <creationDate>2017-05-25T09:59:14.578</creationDate>
        <agentCode>XXXXX</agentCode>
        <agentLocationCode>XXXXX</agentLocationCode>
        <agentOperator>XXXXXX</agentOperator>
        <paymentType>CASH</paymentType>
        <sendingCurrency>GBP</sendingCurrency>
        <purpose>DONATION</purpose>
        <clientRelation>FAMILY</clientRelation>
        <sourceOfFunds>PAYROLL</sourceOfFunds>
        <sendingCountry>GBR</sendingCountry>
        <payerLocationCode>XXXXX</payerLocationCode>
        <payoutCountry>IND</payoutCountry>
        <payoutCurrency>INR</payoutCurrency>
        <totalSale>300</totalSale>
        <variableFee>0</variableFee>
        <payerCode>XXXX</payerCode>
        <deliveryMethod>BANK_DEPOSIT</deliveryMethod>
        <fixedFee>0</fixedFee>
        <sender>
          <firstLastName>TEST</firstLastName>
          <firstName>SENDER</firstName>
          <secondLastName/>
          <secondName/>
          <address>TEST STREET</address>
          <city>LONDON</city>
          <clientId>22961766</clientId>
          <countryOfBirth>PAK</countryOfBirth>
          <dateOfBirth>1989-12-31T09:59:14.578</dateOfBirth>
          <country>GBR</country>
          <email>test@email.com</email>
          <mobilePhoneCountry>GBR</mobilePhoneCountry>
          <mobilePhoneNumber>07123456798</mobilePhoneNumber>
          <occupationId>165</occupationId>
          <originalId>8498851</originalId>
          <streetName>TEST STREET</streetName>
          <streetNumber/>
          <type>INDIVIDUAL</type>
          <zip>ABC123</zip>
          <gender>MALE</gender>
          <ids>
            <clientIdId>11803806</clientIdId>
            <idExpirationDate>2020-07-01T09:59:14.578</idExpirationDate>
            <idIssueCountry>GBR</idIssueCountry>
            <idIssuedBy>PAK</idIssuedBy>
            <idNumber>123456</idNumber>
            <idType>PASSPORT</idType>
            <originalId>959595</originalId>
          </ids>
          <ids>
            <clientIdId>11803806</clientIdId>
            <idExpirationDate>2020-07-01T09:59:14.578</idExpirationDate>
            <idIssueCountry>GBR</idIssueCountry>
            <idIssuedBy>PAK</idIssuedBy>
            <idNumber>022256320</idNumber>
            <idType>PASSPORT</idType>
            <originalId>65485498</originalId>
          </ids>
        </sender>
        <beneficiary>
          <firstLastName>TEST</firstLastName>
          <firstName>DAVID</firstName>
          <nationality>IND</nationality>
          <originalId>30647815</originalId>
          <address>ADDRESS</address>
          <city>DELHI</city>
          <country>IND</country>
          <email>152@MAIL.COM</email>
          <phoneNumber>8611045525</phoneNumber>
          <type>INDIVIDUAL</type>
          <zip>2393</zip>
```

```
          <countryOfBirth>IND</countryOfBirth>
          <banks>
            <bankId>1786</bankId>
            <bankAccountType>CHECKING</bankAccountType>
            <bankIban>XXXXXXXXXXXXXXXXX</bankIban>
            <bankBic>BKID0004186</bankBic>
            <originalId>4126410</originalId>
            <bankBranchName>APCAR GARDEN</bankBranchName>
          </banks>
        </beneficiary>
       * <selectedPayoutClientBankOriginalId>4126410</selectedPayoutClientBankOriginalId>
      </arg0>
    </web:importTransaction>
  </soapenv:Body>
</soapenv:Envelope>
```

**(This field is necessary when there are more then one Bank Account details and you need to specify which one to use)*

ImportTransaction example Response - Bank deposit

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns2:importTransactionResponse xmlns:ns2="http://webservice.ie.smallworldfs.com/">
      <return>
        <companyPassword>PASSWORD SHA1 ENCRYPTED HERE </companyPassword>
        <companyUserName>apiXXXX</companyUserName>
        <server>serverua</server>
        <agentCode>XXXXX</agentCode>
        <beneficiary>
          <address>ADDRESS</address>
          <changeDate>2018-05-21T12:07:08.347+01:00</changeDate>
          <city>DELHI</city>
          <clientId>35634452</clientId>
          <country>IND</country>
          <email>152@MAIL.COM</email>
          <originalId>30647815</originalId>
          <phoneCountry>IND</phoneCountry>
          <phoneNumber>8611045525</phoneNumber>
          <type>INDIVIDUAL</type>
          <zip>2393</zip>
          <countryOfBirth>IND</countryOfBirth>
          <firstLastName>TEST</firstLastName>
          <firstName>DAVID</firstName>
          <nationality>IND</nationality>
          <secondLastName/>
          <secondName/>
        </beneficiary>
        <creationDate>2017-05-25T09:59:14.578+01:00</creationDate>
```

```
<deliveryMethod>BANK_DEPOSIT</deliveryMethod>
<fixedFee>0.0</fixedFee>
<folioNumber>FOLIO WILL BE GENERATED HERE</folioNumber>
<mtn>MTN WILL BE GENERATED HERE</mtn>
<payerLocationCode>10704</payerLocationCode>
<payerMtn>0</payerMtn>
<payoutCountry>IND</payoutCountry>
<payoutCurrency>INR</payoutCurrency>
<payoutPrincipal>36000.0</payoutPrincipal>
<promotionAmount>0.0</promotionAmount>
<sender>
  <address>TEST STREET</address>
  <changeDate>2018-05-21T12:07:08.347+01:00</changeDate>
  <city>LONDON</city>
  <clientId>35634451</clientId>
  <country>GBR</country>
  <email>test@email.com</email>
  <mobilePhoneCountry>GBR</mobilePhoneCountry>
  <mobilePhoneNumber>07123456798</mobilePhoneNumber>
  <occupationId>165</occupationId>
  <originalId>8498851</originalId>
  <streetName>TEST STREET</streetName>
  <streetNumber/>
  <type>INDIVIDUAL</type>
  <zip>ABC123</zip>
  <countryOfBirth>PAK</countryOfBirth>
  <dateOfBirth>1989-12-31T09:59:14.578Z</dateOfBirth>
  <firstLastName>TEST</firstLastName>
  <firstName>SENDER</firstName>
  <gender>MALE</gender>
  <secondLastName/>
  <secondName/>
</sender>
<sendingCountry>GBR</sendingCountry>
<totalSale>300.0</totalSale>
<variableFee>0.0</variableFee>
<agentLocationCode>2003900</agentLocationCode>
<agentMtn>5</agentMtn>
<agentOperator>api2003900</agentOperator>
<clientRelation>FAMILY</clientRelation>
<complianceDetails>
  <blocking>false</blocking>
  <id>1</id>
  <idType>NATURALIZATION</idType>
  <reason>SPECIFY_PURPOSE</reason>
  <reasonTranslated>Specify Purpose</reasonTranslated>
  <ruleId>9009</ruleId>
  <ruleType>COMPLIANCE</ruleType>
  <status>INACTIVE</status>
  <type>PURPOSE_REQUIRED</type>
```

```
        </complianceDetails>
        <complianceDetails>
          <blocking>false</blocking>
          <id>2</id>
          <idType>PASSPORT</idType>
          <reason>CLIENT_RELATION</reason>
          <reasonTranslated>Client Relation</reasonTranslated>
          <ruleId>9009</ruleId>
          <ruleType>COMPLIANCE</ruleType>
          <status>INACTIVE</status>
          <type>CLIENT_RELATION_REQUIRED</type>
        </complianceDetails>
        <complianceDetails>
          <blocking>false</blocking>
          <id>3</id>
           <reasonTranslated>ID: 97 Phone: 2 Address: 0 First Lastname: 19 Second Lastname: 0 TaxCode:
         0</reasonTranslated>
          <ruleId>460</ruleId>
          <ruleType>COMPLIANCE</ruleType>
          <status>INACTIVE</status>
          <type>CLIENT_MATCHING_OVERFLOW</type>
        </complianceDetails>
        <importStatus>ACKNOWLEDGE_PENDING</importStatus>
        <payerCode>XXX</payerCode>
        <paymentType>CASH</paymentType>
        <purpose>DONATION</purpose>
        <selectedPayoutClientBankOriginalId>4126410</selectedPayoutClientBankOriginalId>
        <sendingCurrency>GBP</sendingCurrency>
        <sendingPrincipal>300.0</sendingPrincipal>
        <sourceOfFunds>PAYROLL</sourceOfFunds>
        <userId>412781</userId>
      </return>
    </ns2:importTransactionResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

ImportTransaction example Request - Pick Up

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:web="http://webservice.ie.smallworldfs.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:importTransaction>
      <arg0>
        <companyUserName>apiXXXX</companyUserName>
        <companyPassword>Password SHA1 Encrypted here</companyPassword>
        <creationDate>2017-05-25T09:59:14.578</creationDate>
        <agentCode>XXXX</agentCode>
        <agentLocationCode>XXXX</agentLocationCode>
```

```
<agentOperator>wsXXXX</agentOperator>
<paymentType>CASH</paymentType>
<sendingCurrency>GBP</sendingCurrency>
<purpose>DONATION</purpose>
<clientRelation>FAMILY</clientRelation>
<sourceOfFunds>PAYROLL</sourceOfFunds>
<sendingCountry>GBR</sendingCountry>
<payerLocationCode>XXXX</payerLocationCode>
<payoutCountry>IND</payoutCountry>
<payoutCurrency>INR</payoutCurrency>
<totalSale>300</totalSale>
<variableFee>0</variableFee>
<payerCode>XXXX</payerCode>
<deliveryMethod>PICK_UP</deliveryMethod>
<fixedFee>0</fixedFee>
<sender>
  <firstLastName>TEST</firstLastName>
  <firstName>SENDER</firstName>
  <secondLastName/>
  <secondName/>
  <address>TEST STREET</address>
  <city>LONDON</city>
  <clientId>22961766</clientId>
  <countryOfBirth>PAK</countryOfBirth>
  <dateOfBirth>1989-12-31T09:59:14.578</dateOfBirth>
  <country>GBR</country>
  <email>test@email.com</email>
  <mobilePhoneCountry>GBR</mobilePhoneCountry>
  <mobilePhoneNumber>07123456798</mobilePhoneNumber>
  <occupationId>165</occupationId>
  <originalId>8498851</originalId>
  <streetName>TEST STREET</streetName>
  <streetNumber/>
  <type>INDIVIDUAL</type>
  <zip>ABC123</zip>
  <gender>MALE</gender>
  <ids>
    <clientIdId>11803806</clientIdId>
    <idExpirationDate>2020-07-01T09:59:14.578</idExpirationDate>
    <idIssueCountry>GBR</idIssueCountry>
    <idIssuedBy>PAK</idIssuedBy>
    <idNumber>123456</idNumber>
    <idType>PASSPORT</idType>
    <originalId>959595</originalId>
  </ids>
  <ids>
    <clientIdId>11803806</clientIdId>
    <idExpirationDate>2020-07-01T09:59:14.578</idExpirationDate>
    <idIssueCountry>GBR</idIssueCountry>
    <idIssuedBy>PAK</idIssuedBy>
```

```
            <idNumber>022256320</idNumber>
            <idType>PASSPORT</idType>
            <originalId>65485498</originalId>
          </ids>
        </sender>
        <beneficiary>
          <firstLastName>TEST</firstLastName>
          <firstName>DAVID</firstName>
          <nationality>IND</nationality>
          <originalId>30647815</originalId>
          <address>ADDRESS</address>
          <city>DELHI</city>
          <country>IND</country>
          <email>152@MAIL.COM</email>
          <phoneNumber>8611045525</phoneNumber>
          <type>INDIVIDUAL</type>
          <zip>2393</zip>
          <countryOfBirth>IND</countryOfBirth>
        </beneficiary>
      </arg0>
    </web:importTransaction>
  </soapenv:Body>
</soapenv:Envelope>
```

ImportTransaction example Response - Pick Up

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns2:importTransactionResponse xmlns:ns2="http://webservice.ie.smallworldfs.com/">
      <return>
        <companyPassword>PASSWORD SHA1 ENCRYPTED here </companyPassword>
        <companyUserName>apiXXXXX</companyUserName>
        <server>serverua</server>
        <agentCode>XXXX</agentCode>
        <beneficiary>
          <address>ADDRESS</address>
          <changeDate>2018-05-21T11:53:19.769+01:00</changeDate>
          <city>DELHI</city>
          <clientId>35634452</clientId>
          <country>IND</country>
          <email>152@MAIL.COM</email>
          <originalId>30647815</originalId>
          <phoneCountry>IND</phoneCountry>
          <phoneNumber>8611045525</phoneNumber>
          <type>INDIVIDUAL</type>
          <zip>2393</zip>
          <countryOfBirth>IND</countryOfBirth>
          <firstLastName>TEST</firstLastName>
          <firstName>DAVID</firstName>
          <nationality>IND</nationality>
          <secondLastName/>
          <secondName/>
        </beneficiary>
        <creationDate>2017-05-25T09:59:14.578+01:00</creationDate>
        <deliveryMethod>PICK_UP</deliveryMethod>
        <fixedFee>0.0</fixedFee>
        <folioNumber>Folio will be generated here </folioNumber>
        <mtn>MTN will be generated here</mtn>
        <payerLocationCode>XXXXX</payerLocationCode>
        <payerMtn>0</payerMtn>
        <payoutCountry>IND</payoutCountry>
        <payoutCurrency>INR</payoutCurrency>
        <payoutPrincipal>36000.0</payoutPrincipal>
        <promotionAmount>0.0</promotionAmount>
        <sender>
          <address>TEST STREET</address>
          <changeDate>2018-05-21T11:53:19.769+01:00</changeDate>
          <city>LONDON</city>
          <clientId>35634451</clientId>
          <country>GBR</country>
          <email>test@email.com</email>
          <mobilePhoneCountry>GBR</mobilePhoneCountry>
          <mobilePhoneNumber>07123456798</mobilePhoneNumber>
          <occupationId>165</occupationId>
          <originalId>8498851</originalId>
          <streetName>TEST STREET</streetName>
          <streetNumber/>
          <type>INDIVIDUAL</type>
          <zip>ABC123</zip>
          <countryOfBirth>PAK</countryOfBirth>
          <dateOfBirth>1989-12-31T09:59:14.578Z</dateOfBirth>
          <firstLastName>TEST</firstLastName>
          <firstName>SENDER</firstName>
          <gender>MALE</gender>
          <secondLastName/>
          <secondName/>
        </sender>
        <sendingCountry>GBR</sendingCountry>
        <totalSale>300.0</totalSale>
        <variableFee>0.0</variableFee>
        <agentLocationCode>2003900</agentLocationCode>
        <agentMtn>2</agentMtn>
        <agentOperator>api2003900</agentOperator>
        <clientRelation>FAMILY</clientRelation>
        <complianceDetails>
          <blocking>false</blocking>
          <id>1</id>
          <idType>NATURALIZATION</idType>
          <reason>SPECIFY_PURPOSE</reason>
```

```
          <reasonTranslated>Specify Purpose</reasonTranslated>
          <ruleId>9009</ruleId>
          <ruleType>COMPLIANCE</ruleType>
          <status>INACTIVE</status>
          <type>PURPOSE_REQUIRED</type>
        </complianceDetails>
        <complianceDetails>
          <blocking>false</blocking>
          <id>2</id>
          <idType>PASSPORT</idType>
          <reason>CLIENT_RELATION</reason>
          <reasonTranslated>Client Relation</reasonTranslated>
          <ruleId>9009</ruleId>
          <ruleType>COMPLIANCE</ruleType>
          <status>INACTIVE</status>
          <type>CLIENT_RELATION_REQUIRED</type>
        </complianceDetails>
        <complianceDetails>
          <blocking>false</blocking>
          <id>3</id>
          <reasonTranslated>ID: 97 Phone: 2 Address: 0 First Lastname: 19 Second Lastname: 0 TaxCode:
          0</reasonTranslated>
          <ruleId>460</ruleId>
          <ruleType>COMPLIANCE</ruleType>
          <status>INACTIVE</status>
          <type>CLIENT_MATCHING_OVERFLOW</type>
        </complianceDetails>
        <importStatus>ACKNOWLEDGE_PENDING</importStatus>
        <payerCode>XXXXX</payerCode>
        <paymentType>CASH</paymentType>
        <purpose>DONATION</purpose>
        <sendingCurrency>GBP</sendingCurrency>
        <sendingPrincipal>300.0</sendingPrincipal>
        <sourceOfFunds>PAYROLL</sourceOfFunds>
        <userId>412781</userId>
      </return>
    </ns2:importTransactionResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Import Client example Request

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://webservice.ie.smallworldfs.com/">
<SOAP-ENV:Body>
<ns1:importClient>
<arg0>
  <originalId>225544878</originalId>
  <address>4 TEST</address>
  <city>LONDON</city>
  <country>GBR</country>
  <email>ab99@bc.es</email>
  <mobilePhoneCountry>GBR</mobilePhoneCountry>
  <mobilePhoneNumber>652003211</mobilePhoneNumber>
  <phoneCountry>GBR</phoneCountry>
  <phoneNumber>00125553211</phoneNumber>
  <streetName>TEST</streetName>
  <streetNumber>4</streetNumber>
  <type>INDIVIDUAL</type>
  <zip>ZIP</zip>

  <ids>
    <idIssueCountry>GBR</idIssueCountry>
    <idNumber>5012254134</idNumber>
    <idType>PASSPORT</idType>
    <idExpirationDate>2021-04-09</idExpirationDate>
    <originalId>34544455</originalId>
  </ids>
  <occupationId>142</occupationId>
  <countryOfBirth>GBR</countryOfBirth>
  <dateOfBirth>1968-02-25T00:00:00Z</dateOfBirth>
  <firstLastName>TEST</firstLastName>
  <firstName>test</firstName>
  <nationality>ARE</nationality>
  <secondLastName>TESTING</secondLastName>
  <secondName>CLIENT</secondName>
</arg0>
<arg1>AGENT CODE</arg1>
<arg2>LOCATION CODE</arg2>
<arg3>AGENT OPERATOR</arg3>
<arg4>API USER</arg4>
<arg5>API PASSWORD</arg5>
</ns1:importClient>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Import Client example Response

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Body>
<ns2:importClientResponse xmlns:ns2="http://webservice.ie.smallworldfs.com/">
<return>
    <address>4 TEST</address>
    <changeDate>2018-08-22T12:25:09.911+01:00</changeDate>
    <city>LONDON</city>
    <clientId>38114120</clientId>
    <country>GBR</country>
    <email>ab99@bc.es</email>
    <mobilePhoneCountry>GBR</mobilePhoneCountry>
    <mobilePhoneNumber>652003211</mobilePhoneNumber>
    <occupationId>142</occupationId>
    <phoneCountry>GBR</phoneCountry>
    <phoneNumber>00125553211</phoneNumber>
    <streetName>TEST</streetName>
    <streetNumber>4</streetNumber>
    <type>INDIVIDUAL</type>
    <zip>ZIP</zip>
    <countryOfBirth>GBR</countryOfBirth>
    <dateOfBirth>1968-02-25T00:00:00+01:00</dateOfBirth>
    <firstLastName>ISSUES</firstLastName>
    <firstName>TEST</firstName>
    <ids>
         <clientIdId>13788809</clientIdId>
         <idExpirationDate>2021-04-09T00:00:00+01:00</idExpirationDate>
         <idIssueCountry>GBR</idIssueCountry>
         <idNumber>5012254134</idNumber>
         <idType>PASSPORT</idType>
         <originalId>34544455</originalId>
    </ids>
  <nationality>ARE</nationality>
  <secondLastName>TESTING</secondLastName>
  <secondName>ID</secondName>
</return>
</ns2:importClientResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Mobile Wallet example Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:web="http://webservice.ie.smallworldfs.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <web:importTransaction>
         <!--Optional:-->
         <arg0>
            <companyPassword>PASSWORD SHA1 ENCRYPTED HERE</companyPassword>
            <companyUserName> api******</companyUserName>
            <agentOperator>*****</agentOperator>
            <agentCode>*****</agentCode>
            <agentLocationCode>******.</agentLocationCode>
            <payerCode>....</payerCode>
            <payerLocationCode>....</payerLocationCode>
            <deliveryMethod>MOBILE_WALLET</deliveryMethod>
```

```
<creationDate>2016-01-19T11:24:44+02:00</creationDate>
<sendingCountry>GBR</sendingCountry>
<sendingCurrency>GBP</sendingCurrency>
<sendingPrincipal>1</sendingPrincipal>
<variableFee>0</variableFee>
<fixedFee>0</fixedFee>
<payoutCountry>UGA</payoutCountry>
<payoutCurrency>UGX</payoutCurrency>
<paymentType>ONLINEPAYMENT</paymentType>
<sender>
  <ids>
    <idExpirationDate>2020-12-23</idExpirationDate>
    <idIssueCountry>UGA</idIssueCountry>
    <idIssueDate>2010-01-01</idIssueDate>
    <idIssuedBy>Authority</idIssuedBy>
    <idNumber>....</idNumber>
    <idType>....</idType>
    <originalId>.....</originalId>
  </ids>
  <address>Street 123 h Wycombe</address>
  <city>London</city>
  <country>GBR</country>
  <email>Deepak.m@imimobile.com</email>
  <phoneNumber>....</phoneNumber>
  <mobilePhoneNumber>....</mobilePhoneNumber>
  <originalId>551</originalId>
  <state/>
  <type>INDIVIDUAL</type>
  <zip>....</zip>
  <cityOfBirth/>
  <countryOfBirth>UGA</countryOfBirth>
  <dateOfBirth>1980-01-01</dateOfBirth>
  <firstLastName>Malhotra</firstLastName>
  <firstName>Deepak</firstName>
  <gender>MALE</gender>
  <nationality>UGA</nationality>
  <secondLastName/>
  <secondName/>
  <stateOfBirth/>
  <occupationId>153</occupationId>
</sender>
<beneficiary>
  <banks>
    <originalId>.....</originalId>
    <bankAccountNumber>.....</bankAccountNumber>
    <bankAccountType>MOBILE</bankAccountType>
    <bankBranchName>UGANDA</bankBranchName>
    <bankName> Provider name here </bankName>
    <providerLocationCode>....</providerLocationCode>
    <providerRepCode>....</providerRepCode>
```

```
          </banks>
          <address>Tytyty 677</address>
          <city>Kigali</city>
          <country>UGA</country>
          <email>Jayaprakash.s@imimobile.com</email>
          <originalId>2222222</originalId>
          <phoneNumber>250788380280</phoneNumber>
          <mobilePhoneNumber>250788380280</mobilePhoneNumber>
          <type>INDIVIDUAL</type>
          <countryOfBirth>RWA</countryOfBirth>
          <firstLastName>Sangapati</firstLastName>
          <firstName>Jayaprakash</firstName>
          <nationality>RWA</nationality>
        </beneficiary>
        <clientRelation>BUSINESS</clientRelation>
        <purpose>PAYMENT_EXPENSES</purpose>
        <sourceOfFunds>Payroll</sourceOfFunds>
         <selectedPayoutClientBankOriginalId> If there are more than one banks, this field is
       mandatory</selectedPayoutClientBankOriginalId>
      </arg0>
    </web:importTransaction>
  </soapenv:Body>
</soapenv:Envelope>
```

Attachment example Request

```
<soapenv:Envelope ……. />
    <soapenv:Header/>
     <soapenv:Body>
        <web:importAttachment>
          <arg0>
            <attachment> attachment here...</attachment>
            <attachmentId>
              <attachmentId>18591512</attachmentId>
              <clientId>20812166</clientId>
            </attachmentId>
            <attachmentType>KYC</attachmentType>
            <mimeType/>
            <name>test.pd </name>
            <title>sedner_ KYC</title>
          </arg0>
          <arg1>apiXXXXX</arg1>
          <arg2>PASSWORD SHA1 ENCRYPTED HERE</arg2>
        </web:importAttachment>
      </soapenv:Body>
</soapenv:Envelope>
```

As a result document will be uploaded in client’s profile and you will receive the following response:

```
<return xmlns:ns2 = "http://webservice.ie.smallworldfs.com/">
   <attachmentId>5126825</attachmentId>
   <clientId>21095567</clientId>
   <clientIdId>0</clientIdId>
    <mtn>0</mtn>
</return>
```

Calculate example Request

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:ns1="http://webservice.ie.smallworldfs.com/">
  <SOAP-ENV:Body>
    <ns1:calculate>
      <arg0>
        <companyPassword>PASSWORD SHA1 ENCRYPTED HERE</companyPassword>
        <companyUserName>api*****</companyUserName>
        <agentCode>********</agentCode>
        <creationDate>2019-04-16 11:49:51</creationDate>
        <fixedFee>0</fixedFee>
        <payerLocationCode>10100</payerLocationCode>
        <payoutCountry>POL</payoutCountry>
        <payoutCurrency>PLN</payoutCurrency>
        <sendingCountry>BRA</sendingCountry>
        <totalSale>100</totalSale>
        <agentLocationCode>******</agentLocationCode>
        <agentOperator>*******</agentOperator>
        <deliveryMethod>BANK_DEPOSIT</deliveryMethod>
        <clientRelation>FAMILY</clientRelation>
        <sendingCurrency>BRL</sendingCurrency>
        <sendingPrincipal>100</sendingPrincipal>
      </arg0>
    </ns1:calculate>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Calculate example Response

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns2:calculateResponse xmlns:ns2="http://webservice.ie.smallworldfs.com/">
      <return>
        <payoutPrincipal>91.11304708061215</payoutPrincipal>
        <payoutRepresentativeCode>10100</payoutRepresentativeCode>
        <principal>100.0</principal>
        <promotionAmount>0.0</promotionAmount>
        <rate>0.9111304708061214</rate>
        <taxes>
          <direction>SEND</direction>
          <id>0</id>
          <percentage>0.38</percentage>
          <sequenceId>1</sequenceId>
          <taxAmount>0.38</taxAmount>
          <taxCode>IOF</taxCode>
        </taxes>
        <totalFee>20.0</totalFee>
        <totalSale>120.38</totalSale>
      </return>
      <return>
        <payoutPrincipal>91.11304708061215</payoutPrincipal>
        <payoutRepresentativeCode>10288</payoutRepresentativeCode>
        <principal>100.0</principal>
        <promotionAmount>0.0</promotionAmount>
        <rate>0.9111304708061214</rate>
        <taxes>
          <direction>SEND</direction>
          <id>0</id>
          <percentage>0.38</percentage>
          <sequenceId>1</sequenceId>
          <taxAmount>0.38</taxAmount>
          <taxCode>IOF</taxCode>
        </taxes>
        <totalFee>20.0</totalFee>
        <totalSale>120.38</totalSale>
      </return>
      <return>
        <payoutPrincipal>91.11304708061215</payoutPrincipal>
        <payoutRepresentativeCode>10289</payoutRepresentativeCode>
        <principal>100.0</principal>
        <promotionAmount>0.0</promotionAmount>
        <rate>0.9111304708061214</rate>
        <taxes>
          <direction>SEND</direction>
          <id>0</id>
          <percentage>0.38</percentage>
          <sequenceId>1</sequenceId>
          <taxAmount>0.38</taxAmount>
          <taxCode>IOF</taxCode>
        </taxes>
        <totalFee>20.0</totalFee>
        <totalSale>120.38</totalSale>
      </return>
    </ns2:calculateResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

getFolio example Request

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:web="http://webservice.ie.smallworldfs.com/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:getFolioNumber>
      <!--Optional:-->
      <arg0>2015-06-03T00:00:00</arg0>
      <arg1>agentCode</arg1>
      <arg2>OriginalID of the transaction</arg2>
      <arg3>payer rep code</arg3>
      <!--Optional:-->
      <arg4>api*****</arg4>
      <!--Optional:-->
      <arg5>PASSWORD SHA1 ENCRYPTED HERE</arg5>
    </web:getFolioNumber>
  </soapenv:Body>
</soapenv:Envelope>
```

FAQ

Q: How can we identify which country or currency codes to use?
A: Please always use standard ISO3 codes for countries and currencies.

Q: Are MTN and Folio numbers unique?
A: Yes, folio and *MTN* are unique per transaction.

Q: What value should we use for eventDate and ChangeDate tags?
A: You should use current date with Gregorian format for both tags. Example of Gregorian time format: 2019-06-25T11:58:18+02:00

Q: To Which countries can we transfer money?
A: You should consult with your business developer who signed the agreement about cooperation.

Q: Which sending currency should we use?
A: You should consult with your business developer who signed the agreement about cooperation.

Q: Can a country receive more than 1 currency?

A: Yes.

Q: Is a branch required when transferring money?
A: Branch information is not required for all payer companies.

Q: Can we cancel transactions?
A: Yes, by calling service cancelTransaction. We highly recommend do discuss cancellation flow before implementing it.

Q: Do we need to call Calculate before importing a transaction?
A: We highly recommend to use calculate method. Even so, financial data received from calculate method response, can also be found in importTransaction response

Q: Is there an API for collecting rates? How do we know when rates change?
A: You need to call calculate method to know exact rate in every moment. Rates can be send by email as well, depending on agreement.

Q: Will we need to send any compliance documents via API or we are not required to send any compliance documents to Smallworld?
A: If compliance is dealt by Small World, then yes, you will need to use importAttachment service to send any required documents for compliance purposes.

Q: When should we call AcknowledgeTransaction method?
A: You should call AcknowledgeTransaction method immediately after transaction is created, you should call this method for every single transaction.

Q: Which originalID should I use?
A: We highly recommend to use originalIds in every method where this field is available in order to avoid duplications. For some methods you will need to provide originalId in order to obtain results.

Q: What should we do If response does not include MTN number when we call importTransaction?
A: This means that transaction was not created because there are some compliance blocking issues, as for example, required documents to be attached (Indicated in the response), for this purpose you should use importAttachment method.

Q: I get HTTP error 403 in response.
A: This error means the certificate is not correctly installed in your application. Please check you can access our end point from a browser to discard network problems.
Find below examples for code certificate usage:

.net

```
System.Net.ServicePointManager.ServerCertificateValidationCallback = New
System.Net.Security.RemoteCertificateValidationCallback(AddressOf AcceptAllCerts)

Private Function AcceptAllCerts(ByVal sender As Object, ByVal cert As
System.Security.Cryptography.X509Certificates.X509Certificate, ByVal chain As
System.Security.Cryptography.X509Certificates.X509Chain, ByVal SslPolicyErrors As
System.Net.Security.SslPolicyErrors) As Boolean
       Return True
   End Function
```

PHP

```
//intialising the curl for XML parsing
      $cURL = curl_init();
      //http settings
      $HttpHeader[] = 'SOAPAction:';
      $HttpHeader[] = 'Content-Type: text/xml; charset = utf-8';
      $HttpHeader[] = 'Connection: close';
      $certFile = cert_file.pem';
      $userAgent = 'Small World LCC Money Transfer';

 $url = "https://main.oswfs.com/ImportExport/SimpleTransactionImportService?WSDL";
 $soapMsg = "soap xml goes here...";

      curl_setopt($cURL, CURLOPT_HEADER, false);
      curl_setopt($cURL, CURLOPT_HTTPHEADER, $HttpHeader);
      curl_setopt($cURL, CURLOPT_POST, true);
      curl_setopt($cURL, CURLOPT_URL, $url);
      curl_setopt($cURL, CURLOPT_USERAGENT, $userAgent);
      curl_setopt($cURL, CURLOPT_POSTFIELDS, $soapMsg);
      curl_setopt($cURL, CURLOPT_RETURNTRANSFER, true);
      curl_setopt($cURL, CURLOPT_ENCODING, "UTF-8");
      curl_setopt($cURL, CURLOPT_SSL_VERIFYHOST, false);
      curl_setopt($cURL, CURLOPT_SSL_VERIFYPEER, false);
      curl_setopt($cURL, CURLOPT_SSLCERT, $certFile);
 curl_setopt($cURL, CURLOPT_SSLCERTPASSWD, "password here");

      $response = curl_exec($cURL);
      $errorNo = curl_errno($cURL);
      $errorMsg = curl_error($cURL);
      curl_close($cURL);
```

JAVA

```
System.setProperty("javax.net.ssl.keyStore", CERT_PATH);
System.setProperty("javax.net.ssl.keyStoreType", "PKCS12");
System.setProperty("javax.net.ssl.keyStorePassword", CERT_PASS );
```