



SOAP INTERACTION OF QIWI WALLET AND ONLINE-STORES

DESCRIPTION
ver. 1.3

TABLE OF CONTENTS

1.	GENERAL INFORMATION	3
2.	FUNCTIONS	4
2.1.	CREATE A BILL	4
2.2.	CANCEL A BILL	4
2.3.	CHECK BILL STATUS	4
2.4.	GET BILLS LIST	5
3.	PROVIDERS FUNCTIONS	6
3.1.	PAYMENT STATUS CHANGE NOTIFICATION	6
4.	SCHEME OF INTERACTION	7
APPENDIX A:	TERMINATION CODES REFERENCE TABLE	8
APPENDIX B:	BILL STATUSES REFERENCE TABLE	8
	LIST OF FIGURES	9
	LIST OF EXAMPLES	9

1. GENERAL INFORMATION

An interaction between *QIWI Wallet (QW)* and providers' servers is performing with [SOAP](#) protocol on the Web-service level.

In order to do so WDSL-descriptions should be located on each server (IShopServerWS.wsdl – for QW server and and IShopClientWS.wsdl for providers server).

To provide secure data transfer SSL protocol is used on QW server and WSS X.509 digital signature check is used on providers side.

Web-service URL:

- <https://ishop.qiwi.ru/services/ishop>,
- <https://mylk.qiwi.ru/services/ishop>.

2. FUNCTIONS

- [createBill\(\)](#) – creating a bill;
- [cancelBill\(\)](#) – cancellation a bill;
- [checkBill\(\)](#) – checking bill status and getting bill details;
- [getBillList\(\)](#) – getting bills list with a current statuses indication (maximum request period – 31 days).

2.1. Create a Bill

To create account call function **createBill()** with the following parameters:

- **login** – shop login (ID);
- **password** – shop password;
- **user** – user ID (MSISDN);
- **amount** – amount of bill;
- **comment** – comment to the bill displayed to the user;
- **txn** – unique bill ID;
- **lifetime** – bill lifetime (in dd.MM.yyyy HH:mm:ss format);
- **alarm** – send a notification to the user (sms or call);
- **create** – flag to create a new user (if he's not registered in the system yet).

The response contains the function executing result (see [Termination codes reference table](#)).

2.2. Cancel a Bill

To cancel the bill call the **cancelBill()** function with the following parameters:

- **login** – shop login (ID);
- **password** – shop password;
- **txn** – unique bill ID;
-

The response contains the function executing result (see [Termination codes reference table](#)).

2.3. Check Bill Status

To get bill information call the **checkBill()** function with the following parameters:

- **login** – shop login (ID);
- **password** – shop password;
- **txn** – unique bill ID;
- **user** – user ID (MSISDN);
- **amount** – amount of bill;
- **date** – billing date (in dd.MM.yyyy HH:mm:ss format);
- **lifetime** – bill lifetime (in dd.MM.yyyy HH:mm:ss format);

- **status** – bill status (see [Bill statuses reference table](#)) in case of positive value or error code (in case of negative value).

2.4. Get Bills List

getBillList() function allows to get accounts list created in certain time range (31 days – maximum).

Also you can perform search for accounts with a certain status (optional).

Function parameters list:

- **login** – shop login (ID);
- **password** – shop password;
- **dateFrom** – period starting date;
- **dateTo** – period ending date;
- **status** – bills statuses (To get all statuses for all bills indicate «0», to get certain bill status see [Bill statuses reference table](#));
- **txns** – bills ID list in xml format;
- **count** – bills count (in case of positive value) /error code (in case of negative value).

3. PROVIDERS FUNCTIONS

- **updateBill()** – bill status change notification. Function calls after bill payment/revocation.

NOTE

If provider is not performing digital signature check, it's recommended to check bill statuses with **checkBill()** method.

Full description of the functions launching by provider is presented as a javadoc HTML (is-doc.zip file)

3.1. Payment Status Change Notification

To perform bill status change notification **updateBill()** function has to be called with the following parameters:

- **login** – shop (id) login;
- **password** – password. There are 2 ways to form this parameter:
 - Using the WSS X.509 signature when each notification is signed by OSMP server. This option is much safer but more complicated in implementing;
 - Using the simplified algorithm. There is a special value calculated with a following algorithm:

```
uppercase(md5(txn + uppercase(md5(password))))
```

All lines where md5 function is calculating is transforming into bytes in windows-1251 encoding. This option is easier to perform but less secure.

Example 1. Password value calculation example using the simplified algorithm

Let *order*="Order1", and *password*="Shop password", then function

MD5("Shop password")=936638421CA12C3E15E72FA7B75E03CE.

The following value will be written in **password** field:

MD5("Order1"+MD5("Shop password"))=MD5("Order1"+"936638421CA12C3E15E72FA7B75E03CE")=EC19350E3051D8A9834E5A2CF25FD0D9

- **txn** – bill unique ID;
- **status** – new bill status (see [Bill statuses reference table](#)).

Request result is returning as a response (see [Termination codes reference table](#)).

4. SCHEME OF INTERACTION

Fig. 1. Bill payment interaction scheme

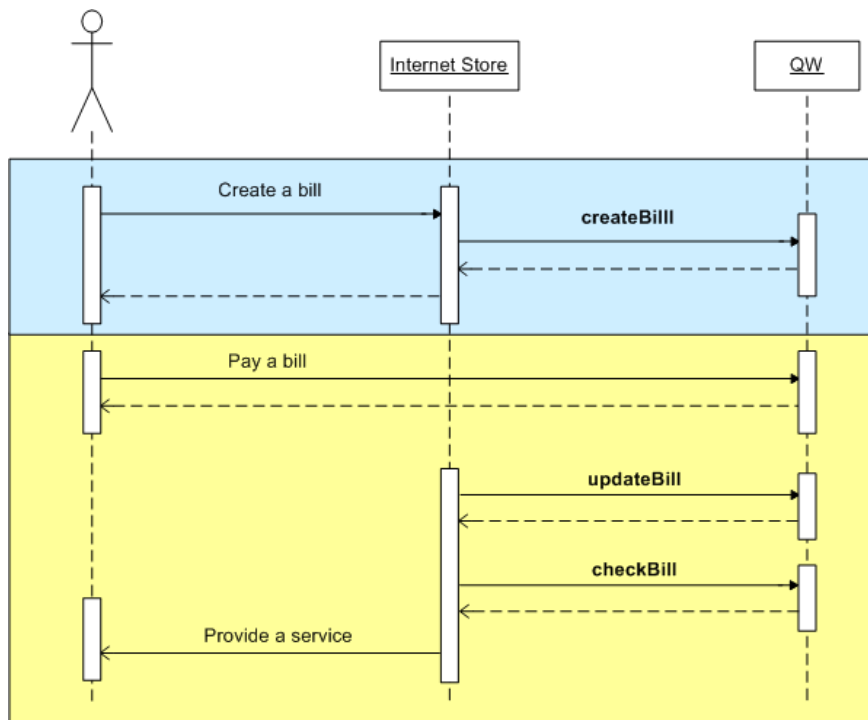
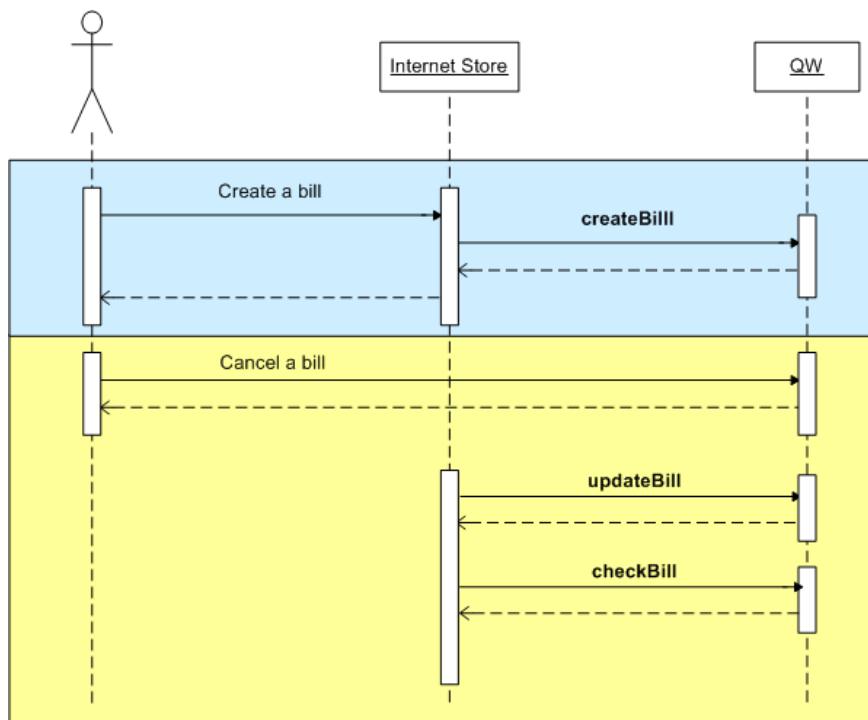


Fig. 2. Bill canceling interaction scheme



ПРИЛОЖЕНИЕ А: TERMINATION CODES REFERENCE TABLE

Termination code	Description
0	Success
13	Server is busy, please repeat your request later
150	Authorization error (wrong login/password)
215	Bill with this txn-id already exists
278	Bill list maximum time range exceeded
298	No such agent in the system
300	Unknown error
330	Encryption error
370	Maximum allowed concurrent requests overlimit

ПРИЛОЖЕНИЕ Б: BILL STATUSES REFERENCE TABLE

Status	Description
50	Made
52	Processing
60	Payed
150	Cancelled (Machine error)
160	Cancelled
161	Cancelled (Timeout)

LIST OF FIGURES

Fig. 1. Bill payment interaction scheme.....	7
Fig. 2. Bill canceling interaction scheme.....	7

LIST OF EXAMPLES

Example 1. Password value calculation example using the simplified algorithm	6
--	---