

W1 Open API

[Введение](#)

[Описание API](#)

[Формат и версионность](#)

[Локализация](#)

[Обработка ошибок](#)

[Авторизация](#)

[OAuth 2.0: Bearer Tokens](#)

[Электронная цифровая подпись](#)

[Captcha](#)

Введение

W1 Open API представляет собой набор веб-сервисов в стиле REST, которые реализуют методы управления кошельком от имени пользователя. Запросы осуществляются по протоколу HTTP 1.1 поверх SSL по адресу <https://api.w1.ru/OpenApi/>.

Описание API

Группа API	Статус	Описание
Balance	Stable	Получение баланса в разных валютах
Profile	Stable	Управление профилем пользователя, получение информации о других пользователях, сохранение пользовательских настроек
Transfers	Stable	Переводы между счетами, получение выписки по операциям
Security	Stable	Получение сессий, восстановление пароля, регистрация
Invoices	Stable	Оплата/выставление счетов, получение их статусов
Payments	Stable	Управление каталогом поставщиков услуг, проведение платежей в пользу поставщиков услуг
Exchanges	Beta	Обмен валют

Notifications	Beta	Получение уведомлений пользователя
Merchants	Beta	Управление интернет магазином
Timeline	Beta	Получение истории действий пользователя в виде Timeline
UserEntry	Beta	Получение истории операций в виде списка
ExternalAccounts	Beta	Внешние платежные инструменты
Partner	Beta	Партнерская программа
BalanceHold	Beta	Управление холдированием средств пользователя

Формат и версионность

W1 Open API поддерживает два формата представления объектов: JSON и XML.

Простой пример запроса выглядит следующим образом:

```
GET /OpenApi/balance/643
Authorization: Bearer 3F9B04DA-B46F-40F5-ACBD-9A4B3FAECD5B
Accept: application/vnd.wallet.openapi.v1+json
```

Ответ

```
HTTP/1.1 200 OK
Content-Type: application/vnd.wallet.openapi.v1+json
[
  {
    "CurrencyId" : 643,
    "Amount" : 0.0000
    ...
  }
]
```

Все запросы **должны** содержать заголовок Ассепт, указывающий на формат ожидаемого ответа.

```
// для формата JSON
Accept: application/vnd.wallet.openapi.v1+json
// для формата XML
```

```
Accept: application/vnd.wallet.openapi.v1+xml
```

В случае, если он не передан или имеет некорректное значение, возвращается HTTP-ответ со статусом [406 Not Acceptable](#).

Для передачи аргументов в теле запроса, дополнительно **должен** передаваться заголовок Content-Type:

```
// для формата JSON
Content-Type: application/vnd.wallet.openapi.v1+json
// для формата XML
Content-Type: application/vnd.wallet.openapi.v1+xml
```

В случае, если заголовок Content-Type не соответствует перечисленным вариантам или отсутствует, возвращается HTTP-ответ со статусом [415 Unsupported Media Type](#).

Локализация

Для получения текстовой информации на желаемом языке **может** передаваться заголовок Accept-Language. Примеры:

```
Accept-Language: ru-RU
```

```
Accept-Language: en-US
```

Обработка ошибок

Ошибки возвращаются в HTTP-статусе кодом 4xx и 5xx, согласно [RFC 2616](#).

Пример ответа с ошибкой

```
401 Unauthorized
WWW-Authenticate: X-Wallet-Signature realm="wallet",
error="invalid_signature", error_description="invalid signature"
Content-Length: 68
Content-Type: application/vnd.wallet.openapi.v1+json

{"Error":"invalid_signature","ErrorDescription":"invalid
signature"}
```

Ответ с ошибкой **может** включать тело со следующими полями:

Поле	Описание
Error	Код ошибки.
ErrorDescription	Описание ошибки.

Общие коды ошибок:

Код ошибки	Описание
INTERNAL_SERVER_ERROR	Внутренняя ошибка сервера.
ARGUMENT_ERROR	Неверное значение аргумента метода.
NOT_FOUND	Указанный ресурс не найден.

Код и описание ошибки авторизации возвращается также в заголовке `WWW-Authenticate`, который имеет следующий вид:

```
WWW-Authenticate: {схема_авторизации} realm="wallet"  
[, error="код_ошибки", error_description="описание ошибки"]
```

где схема авторизации - `Bearer` (в случае неверного кода авторизации) или `X-Wallet-Signature` (неверная электронная цифровая подпись).

Авторизация

OAuth 2.0: Bearer Tokens

Авторизация осуществляется по протоколу [OAuth 2.0: Bearer Tokens](#). Каждый запрос должен иметь заголовок `Authorization` следующего вида:

```
Authorization: Bearer {access_token}
```

где `{access_token}` - код авторизации, полученный с использованием [W1 OAuth API](#).

В случае, если в запросе не используется метод авторизации Bearer, то возвращается ответ со статусом 401:

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Bearer realm="wallet"
```

В случае ошибки авторизации по методу Bearer, возможны следующие HTTP-статусы и значения кода ошибки:

HTTP-статус	Код ошибки	Описание
401 Unauthorized	invalid_token	Неверный или просроченный код авторизации.
403 Forbidden	insufficient_scope	Недостаточно прав доступа.

Например:

```
HTTP/1.1 403 Forbidden
WWW-Authenticate: Bearer realm="wallet",
error="insufficient_scope", error_description="insufficient_scope"
Content-Type: application/vnd.wallet.openapi.v1+json

{"Error": "insufficient_scope",
"ErrorDescription": "insufficient scope"}
```

Электронная цифровая подпись

В качестве дополнительной, но не обязательной, меры безопасности запрос **может** содержать электронную цифровую подпись (ЭЦП), сформированную с использованием [выбранного](#) метода формирования ЭЦП, переданную в заголовке X-Wallet-Signature, например:

```
X-Wallet-Signature: 0J7Qv9C70LDRgtCwINC30LDQutCw0LfQsA==
```

При передаче заголовка X-Wallet-Signature, запрос **должен** содержать заголовок X-Wallet-Timestamp, содержащий дату формирования запроса в часовом поясе UTC+0. Значение заголовка должно иметь формат yyyy-MM-ddTHH:mm:ss.

Цифровая подпись запроса формируется путем конкатенации URL, на который отправляется запрос, кода авторизации {access_token}, значения заголовка

X-Wallet-Timestamp, тела запроса и секретного ключа приложения. Полученная строка, представленная в кодировке UTF-8, подписывается с помощью выбранного алгоритма ЭЦП, после чего байтовое представление подписи кодируется в Base64:

```
Signature = Base64(SignatureMethod(UTF8(
    URL + AccessToken + Timestamp + RequestBody + SecretKey1)));
```

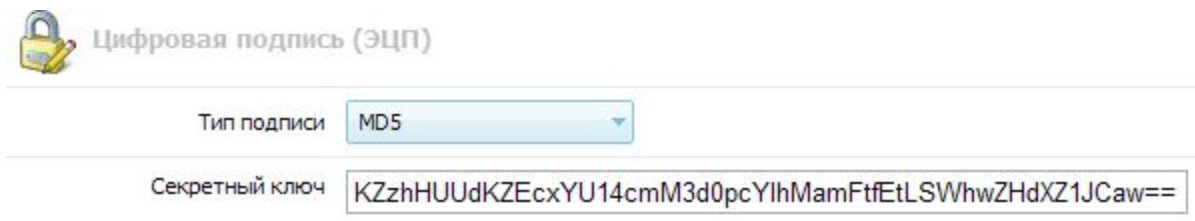
На запросы приложений, содержащие подпись, сервер генерирует ответ, который так же может быть проверен на подлинность, при этом ответу добавляются заголовки:


- X-Wallet-Timestamp - дата формирования ответа в часовом поясе и формате аналогичном дате формирования запроса.
- X-Wallet-Signature - ЭЦП ответа сервера.

Цифровая подпись ответа формируется путем конкатенации значения подписи запроса, заголовка ответа X-Wallet-Timestamp, тела ответа и секретного ключа приложения. Полученная строка преобразовывается и хешируется аналогично алгоритму для запросов:

```
Signature = Base64(SignatureMethod(UTF8(
    IncomingSignature + Timestamp + ResponseBody + SecretKey)))
```

Метод формирования ЭЦП и «секретный ключ» приложения можно настроить в личном кабинете учетной записи «владельца» приложения на вкладке «Настройки» в разделе «Интернет-магазин», как показано на рисунке:



 Цифровая подпись (ЭЦП)

Тип подписи: MD5

Секретный ключ: KZzhHUUdKZEcYU14cmM3d0pcYlhMamFtfEtLSWhwZHdXZ1JCaw==

Примечание: При использовании цифровой подписи, основанной на сертификатах, секретная фраза не применяется, используется алгоритм хеширования и вычисления ЭЦП, заданный в сертификате. Обмен ключевой информацией осуществляется в рамках договора об электронном документообороте.

Коды ошибок:

¹В качестве функции вычисления хеша/подписи должен быть использован алгоритм, заданный в настройках.

Код ошибки	Описание
INVALID_SIGNATURE	Неверная подпись.
INVALID_TIMESTAMP	Неверное значение заголовка X-Wallet-Timestamp.

Captcha

При вызове некоторых методов может вернуться ошибка

```
{
  "Error": "captcha_required",
  "ErrorDescription": "Method not allowed without captcha params"
}
```

Это означает, что чтобы воспользоваться методом, нужно добавить в запрос заголовки

Заголовок	Описание
X-Wallet-Captchalid	Идентификатор капчи
X-Wallet-CaptchaCode	Строка, набранная пользователем, “решение” капчи

Нужно создать капчу с помощью [специального метода](#), потом показать пользователю картинку по адресу, полученному в ответе, и подставить соответствующие поля.

При неправильном значении кода вернется ошибка

```
{
  "Error": "invalid_captcha",
  "ErrorDescription": "unable to verify specified captcha"
}
```