

Документация к API кошельков. Версия 0.1

[Описание сервиса](#)

[Структура возвращаемого сервисом результата](#)

[Регистрация Агента в системе IntellectMoney](#)

[Регистрация пользователя в системе IntellectMoney](#)

[Описание методов API](#)

[Идентификация пользователя](#)

[Метод UserIdentification](#)

[Метод проверки результата идентификации пользователя](#)

[Метод UserIdentificationState](#)

[Метод осуществляющий привязку пользователя](#)

[Метод UserBind](#)

[Метод проверки результата привязки пользователя](#)

[Метод UserBindState](#)

[Метод осуществляющий отвязку](#)

[Метод UserUnbind](#)

[Метод проверки результата отвязки пользователя](#)

[Метод UserUnbindState](#)

[Запрос баланса пользователя](#)

[Метод UserGetBalance](#)

[Метод проверки результата запроса состояния счета](#)

[Метод UserGetBalanceState](#)

[Операция совершения перевода](#)

[Метод UserPay](#)

[Метод проверки результата операции перевода](#)

[Метод UserPayState](#)

[Формирование подписи запроса](#)

[Описание типов](#)

Описание сервиса

Система IntellectMoney предоставляет сервис API Кошельков (<https://api.intellectmoney.ru/walletapi/WalletApiService.asmx>), который позволяет осуществлять следующие операции над кошельками/пользователями:

1. Добавлять сервис поставщика (далее “Агент”) в систему IntellectMoney (на стадии разработки)
2. Осуществлять привязку пользователя к системе IntellectMoney
3. Производить идентификацию добавленного пользователя
4. Осуществлять отвязку пользователя от системы IntellectMoney
5. Запрашивать баланс пользователя
6. Совершать перевод между пользователями

В качестве API выставляется веб сервис, работающий по SOAP протоколу. Все методы сервиса работают в асинхронном режиме. Это означает следующее:

- У каждого метода сервиса, осуществляющего ту или иную операцию, есть метод запроса результата.
- Каждый метод API возвращает результат, который имеет поле Status. Если поле установлено в “Success”, то в поле Data содержится результат. В противном случае, результат еще не готов, либо возникла ошибка при выполнении операции. Об этом опять же можно узнать по полю Status.
- В простейшем случае, метод API может вернуть сразу готовый результат. Если же это не так, то необходимо пользоваться методом запроса статуса операции. Интервал, через который необходимо перезапрашивать состояние операции, будет оговорен на более поздних этапах разработки.

Все методы API, кроме методов возвращающих состояние выполнения операции, имеют в качестве параметра цифровую подпись. Это гарантирует что сообщение не было изменено злоумышленниками при транспортировке и однозначно идентифицирует сервис. Более подробно алгоритм шифрования ЦП будет рассмотрен на более поздних этапах работы по API. Методы возвращающие результат не имеют цифровой подписи, в целях уменьшения нагрузки на API.

Структура возвращаемого сервисом результата

Здесь приводится пример структуры результата, который возвращают методы сервиса (API).

Заголовок ответа:

HTTP/1.1 200 OK

Content-Type: text/xml; charset=utf-8

Content-Length: length

Тело ответа:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <UserRegistrationResponse xmlns="http://tempuri.org/">
      <UserRegistrationResult>
        <Status>
          <OperationStatus>ToProcess or InProcess or Success or Error</OperationStatus>
          <Error>string</Error>
        </Status>
        <OperationId>int</OperationId>
        <Data>T</Data>
      </UserRegistrationResult>
    </UserRegistrationResponse>
  </soap:Body>
</soap:Envelope>
```

Тег Status содержит два параметра. Статус операции (OperationStatus) и сообщение с расшифровкой ошибки.

Операция может находится в следующих статусах

- Created- В очереди обработки, еще не обрабатывался
- ToProcess - В процессе обработки
- Success - Успешно выполнена. Значит результат доступен в ответе (тег Data)
- Error - Ошибка. Смотрим описание ошибки в теге Error

Тег OperationId - Это идентификатор операции. Его предоставляет сервис. Он должен быть уникальным в рамках сервиса. Так как API асинхронный, то этот идентификатор, поможет избежать ошибочного дублирования операций. По данному идентификатору так же можно получить состояние операции, результат операции.

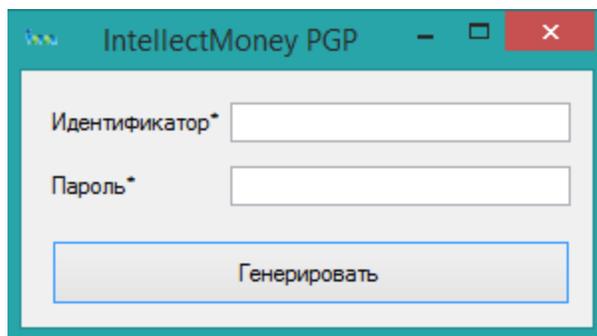
Тег Data содержит результат операции. Может быть разных типов, в зависимости от метода. Именно этот параметр интересует, если Status = Success.

Регистрация Агента в системе IntellectMoney

Чтобы начать пользоваться доступными методами API необходимо пройти регистрацию в системе IntellectMoney. Для регистрации Агента в системе IntellectMoney потребуются PGP ключи, которые можно сгенерировать с помощью утилиты, доступной по адресу

<https://intellectmoney.ru/common/walletapi/PgpKeysGeneratorPackage.exe>

Утилита представляет собой окно с 2-мя полями ввода



В поле идентификатор указывается идентификатор агента (участвует в формировании имени файла)

В поле пароль указывается секретное слово, которое участвует при генерации ключей. Ни идентификатор, ни пароль хранить нигде не требуется.

После генерации ключей Агент производит регистрацию сервиса с помощью метода API (на стадии разработки), или передает регистрационные данные и публичный ключ в на почту ts@intellectmoney.ru

Список параметров, необходимых для регистрации Агента:

- name - Название сервиса Агента
- email - контактная почта Агента
- description - описание сервиса Агента (можно использовать html)
- logourl - ссылка на логотип сервиса Агента (размер логотипа 130x50px)
- userurl - ссылка, на которую будет направлен пользователь после привязки к сервису Агента. Перенаправление пользователя на userurl будет осуществлено с дополнительными параметрами bindKey и serviceUserKey, где
 - bindKey - ключ привязки пользователя к системе IntellectMoney. Создается системой IntellectMoney после привязки пользователя к сервису Агента
 - [serviceUserKey](#) - внутренний идентификатор пользователя в системе Агента, если он был указан при регистрации пользователя
- publickey - публичный PGP ключ Агента

После чего сервис получает свой уникальный идентификатор - serviceld

Регистрация пользователя в системе IntellectMoney

Для регистрации нового пользователя в системе IntellectMoney Агент перенаправляет пользователя на страницу регистрации <https://intellectmoney.ru/ru/registration/personal/> со следующими параметрами:

- `serviceld` - Идентификатор сервиса в системе IntellectMoney (обязательное поле)
- `email` - email пользователя
- `firstName` - Имя пользователя
- `middleName` - Отчество пользователя
- `lastName` - Фамилия пользователя
- `birthdate` - Дата рождения пользователя (передается в формате dd.mm.yyyy)
- `serviceUserKey` - внутренний идентификатор пользователя в системе Агента. После регистрации пользователя в системе IntellectMoney и привязке пользователя к сервису Агента данный параметр будет отправлен на адрес `userurl` (адрес, который Агент указал в качестве ссылки, на которую будет направлен пользователь после привязки к сервису Агента).

Все параметры, передаваемые на страницу регистрации пользователя могут быть пустыми, кроме параметра `serviceld`.

Пример url адреса с параметрами, на который необходимо отправить пользователя для прохождения регистрации:

`https://intellectmoney.ru/ru/registration/personal/?serviceld=29393&email=test@service.ru&firstName=Ivanov&serviceUserKey=2939393`

Описание методов API

1. Идентификация пользователя

Метод UserIdentification

Список параметров:

- operationId (int) Идентификатор операции.
- serviceId (int) Идентификатор сервиса
- sign (string) Цифровая подпись (приватный ключ)
- bindKey (string) Ключ привязки пользователя к системе ИнтеллектМани
- firstName
- lastName
- middleName
- passportSeries (string) Серия паспорта
- passportNumber (string) Номер паспорта
- passportWhenGive (DateTime) Дата выдачи паспорта
- inn (string) Номер ИНН
- snils (string) Номер СНИЛС
- oms (string) Номер ОМС
- phone (string) Контактный номер телефона
- isResident (bool) Является ли резидентом РФ

Результат: в теге Data возвращается тип [UserIdentificationState](#). В данном случае он говорит о том, выполнена операция или нет, а не прошла ли она идентификацию. Об успешности идентификации следует узнавать по полю Status

Тело ответа:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <UserRegistrationResponse xmlns="http://tempuri.org/">
      <UserRegistrationResult>
        <Status>
          <OperationStatus>ToProcess or InProcess or Success or Error</OperationStatus>
          <Error>string</Error>
        </Status>
        <OperationId>int</OperationId>
        <Data>UserIdentificationState</Data>
      </UserRegistrationResult>
    </UserRegistrationResponse>
  </soap:Body>
</soap:Envelope>
```

```
</UserRegistrationResponse>  
</soap:Body>  
</soap:Envelope>
```

2. Метод проверки результата идентификации пользователя

Метод `UserIdentificationState`

Список параметров:

- `operationId` Идентификатор операции. Предоставляется сервисом.
- `serviceId` Идентификатор сервиса

Результат: в теге `Data` возвращается тип [UserIdentificationState](#).

Тело ответа:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <UserRegistrationResponse xmlns="http://tempuri.org/">
      <UserRegistrationResult>
        <Status>
          <OperationStatus>ToProcess or InProcess or Success or Error</OperationStatus>
          <Error>string</Error>
        </Status>
        <OperationId>int</OperationId>
        <Data>UserIdentificationState</Data>
      </UserRegistrationResult>
    </UserRegistrationResponse>
  </soap:Body>
</soap:Envelope>
```

3. Метод осуществляющий привязку пользователя

Метод UserBind

Список параметров:

- operationId (int) Идентификатор операции. Предоставляется сервисом.
- serviceId (int) Идентификатор сервиса
- sign (string) Цифровая подпись
- bindKey (string) Ключ привязки.

Результат: в теге Data возвращается булевский тип. В данном случае он говорит о том, выполнена операция или еще нет. Об успешности привязки следует узнавать по полю Status.

Тело ответа:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <UserRegistrationResponse xmlns="http://tempuri.org/">
      <UserRegistrationResult>
        <Status>
          <OperationStatus>ToProcess or InProcess or Success or Error</OperationStatus>
          <Error>string</Error>
        </Status>
        <OperationId>int</OperationId>
        <Data>bool</Data>
      </UserRegistrationResult>
    </UserRegistrationResponse>
  </soap:Body>
</soap:Envelope>
```

4. Метод проверки результата привязки пользователя

Метод UserBindState

Список параметров:

- operationId (int) Идентификатор операции. Предоставляется сервисом.
- serviceId (int) Идентификатор сервиса

Результат: в теге Data возвращается булевский тип. В данном случае он говорит о том, выполнена операция или еще нет. Об успешности привязки следует узнавать по полю Status.

Тело ответа:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <UserRegistrationResponse xmlns="http://tempuri.org/">
      <UserRegistrationResult>
        <Status>
          <OperationStatus>ToProcess or InProcess or Success or Error</OperationStatus>
          <Error>string</Error>
        </Status>
        <OperationId>int</OperationId>
        <Data>bool</Data>
      </UserRegistrationResult>
    </UserRegistrationResponse>
  </soap:Body>
</soap:Envelope>
```

5. Метод осуществляющий отвязку

Метод UserUnbind

Список параметров:

- operationId (int) Идентификатор операции.
- serviceId (int) Идентификатор сервиса
- sign (string) Цифровая подпись
- bindKey (string) Ключ привязки.

Результат: в теге Data возвращается булевский тип. В данном случае он говорит о том, выполнена операция или еще нет. Об успешности привязки следует узнавать по полю Status.

Тело ответа:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <UserRegistrationResponse xmlns="http://tempuri.org/">
      <UserRegistrationResult>
        <Status>
          <OperationStatus>ToProcess or InProcess or Success or Error</OperationStatus>
          <Error>string</Error>
        </Status>
        <OperationId>int</OperationId>
        <Data>bool</Data>
      </UserRegistrationResult>
    </UserRegistrationResponse>
  </soap:Body>
</soap:Envelope>
```

6. Метод проверки результата отвязки пользователя

Метод UserUnbindState

Список параметров:

- operationId (int) Идентификатор операции.
- serviceId (int) Идентификатор сервиса

Результат: в теге Data возвращается булевский тип. В данном случае он говорит о том, выполнена операция или еще нет. Об успешности привязки следует узнавать по полю Status.

Тело ответа:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <UserRegistrationResponse xmlns="http://tempuri.org/">
      <UserRegistrationResult>
        <Status>
          <OperationStatus>ToProcess or InProcess or Success or Error</OperationStatus>
          <Error>string</Error>
        </Status>
        <OperationId>int</OperationId>
        <Data>bool</Data>
      </UserRegistrationResult>
    </UserRegistrationResponse>
  </soap:Body>
</soap:Envelope>
```

7. Запрос баланса пользователя

Метод UserGetBalance

Список параметров:

- operationId (int) Идентификатор операции
- serviceId (int) Идентификатор сервиса
- sign (string) Цифровая подпись
- bindKey (string) Ключ привязки.

Результат: тег Data в случае State == Success содержит сумму остатка по счету на текущий момент и валюту счета.

Тело ответа:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <UserRegistrationResponse xmlns="http://tempuri.org/">
      <UserRegistrationResult>
        <Status>
          <OperationStatus>ToProcess or InProcess or Success or Error</OperationStatus>
          <Error>string</Error>
        </Status>
        <OperationId>int</OperationId>
        <Data>
          <Money>
            <Amount>decimal</Amount>
            <Currency>RUB or TST</Currency>
          </Money>
        </Data>
      </UserRegistrationResult>
    </UserRegistrationResponse>
  </soap:Body>
</soap:Envelope>
```

8. Метод проверки результата запроса состояния счета

Метод UserGetBalanceState

Список параметров:

- operationId (int) Идентификатор операции.
- serviceId (int) Идентификатор сервиса

Результат: тег Data в случае State == Success содержит сумму остатка по счету на текущий момент и валюту счета.

Тело ответа:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <UserRegistrationResponse xmlns="http://tempuri.org/">
      <UserRegistrationResult>
        <Status>
          <OperationStatus>ToProcess or InProcess or Success or Error</OperationStatus>
          <Error>string</Error>
        </Status>
        <OperationId>int</OperationId>
        <Data>
          <Money>
            <Amount>decimal</Amount>
            <Currency>RUB or TST</Currency>
          </Money>
        </Data>
      </UserRegistrationResult>
    </UserRegistrationResponse>
  </soap:Body>
</soap:Envelope>
```

9. Операция совершения перевода

Метод UserPay

Список параметров:

- operationId (int) Идентификатор операции.
- serviceId (int) Идентификатор пользователя
- sign (string) Цифровая подпись
- fromBindKey (string) Ключ привязки пользователя, со счета которого делаем перевод
- toBindKey (string) Ключ привязки пользователя, на счет которого делаем перевод
- fromUserPin (int) Секретный код пользователя, со счета которого делаем перевод
- amount (decimal) Сумма перевода
- currency (string) Валюта перевода. Пока только RUB

Результат: в теге Data возвращается булевский тип. В данном случае он говорит о том, выполнена операция или еще нет. Об успешности привязки следует узнавать по полю Status.

Тело ответа:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <UserRegistrationResponse xmlns="http://tempuri.org/">
      <UserRegistrationResult>
        <Status>
          <OperationStatus>ToProcess or InProcess or Success or Error</OperationStatus>
          <Error>string</Error>
        </Status>
        <OperationId>int</OperationId>
        <Data>bool</Data>
      </UserRegistrationResult>
    </UserRegistrationResponse>
  </soap:Body>
</soap:Envelope>
```

10. Метод проверки результата операции перевода

Метод UserPayState

Список параметров:

- operationId (int) Идентификатор операции.
- serviceId (int) Идентификатор сервиса

Результат: в теге Data возвращается булевский тип. В данном случае он говорит о том, выполнена операция или еще нет. Об успешности привязки следует узнавать по полю Status.

Тело ответа:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <UserRegistrationResponse xmlns="http://tempuri.org/">
      <UserRegistrationResult>
        <Status>
          <OperationStatus>ToProcess or InProcess or Success or Error</OperationStatus>
          <Error>string</Error>
        </Status>
        <OperationId>int</OperationId>
        <Data>bool</Data>
      </UserRegistrationResult>
    </UserRegistrationResponse>
  </soap:Body>
</soap:Envelope>
```

Формирование подписи запроса

Подпись формируется по алгоритму:

1. Получаем MD5 хеш строки "serviceId={0}::serviceOperationId={1}"
2. Переводим MD5 хеш в верхний регистр.
3. Полученный хеш подписываем приватным ключем
4. Полученную подпись передаем в параметре sign

Описание типов

UserIdentificationState

```
{  
    None, //необработан  
    Process, //в процессе  
    Success //идентифицирован  
}
```

Пример на php

Чтобы использовать php для запросов к API для начала нужно установить [gnupg библиотеку](#). Затем сформировать ключи БЕЗ ПАРОЛЯ. Приватный и публичный ключ генерируются без парольной защиты. Можно использовать утилиту [Kleopatra - Gpg4Win](#).

```
<?php
```

```
$operationId = <номер операции>;
$serviceId = <номер сервиса>;
$user['bindKey'] = <ключ привязки пользователя>;

$md5 = strtoupper(md5("serviceId=$serviceId::serviceOperationId=$
operationId"));

$ascii = file_get_contents(__DIR__ . '/private.asc');
$start = strpos($ascii, '-----BEGIN PGP PRIVATE KEY BLOCK-----');
$end = strpos($ascii, '-----END PGP PRIVATE KEY BLOCK-----')+34;

$key = substr($ascii, $start, ($end-$start));

$res = gnupg_init();
$info = gnupg_import($res,$key);
gnupg_addsignkey($res,$info['fingerprint']);
$signed = gnupg_sign($res, $md5);

if($err = gnupg_geterror($res)){
    var_dump($err);
}

$data = array(
    'operationId' => $operationId,
    'serviceId' => $serviceId,
    'bindKey' => $user['bindKey'],
    'sign' => $signed
);

$balans = sendApiRequest('UserGetBalance', $data);

function sendApiRequest($method_name, array $data) {
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL,
'https://api.intellectmoney.ru/walletapi/WalletApiService.asmx/' .
$method_name);
```

```
        curl_setopt($ch, CURLOPT_POSTFIELDS,  
http_build_query($data));  
        curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);  
        curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, false);  
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);  
        $r = curl_exec($ch);  
        curl_close($ch);  
        return $r;  
    }
```