



API СЕРВИСА «ЭКСПРЕСС ПЛАТЕЖИ»

Документация по интеграции производителей услуг

Версия 1

ООО «ТриИнком»

Минск
2016

ОГЛАВЛЕНИЕ

| | |
|--|----|
| Описание API..... | 4 |
| Формат запроса | 4 |
| Формат ответа | 4 |
| Включение API..... | 6 |
| Система ерип | 7 |
| Просмотр списка счетов по параметрам | 7 |
| Входные параметры..... | 7 |
| Выходные параметры..... | 7 |
| Пример кода на С#..... | 8 |
| Пример кода на PHP..... | 9 |
| Выставление счета..... | 10 |
| Входные параметры..... | 10 |
| Выходные параметры..... | 11 |
| Пример кода на С#..... | 11 |
| Пример кода на PHP..... | 12 |
| Детальная информация о счете | 14 |
| Входные параметры..... | 14 |
| Выходные параметры..... | 14 |
| Пример кода на С#..... | 15 |
| Пример кода на PHP..... | 15 |
| Статус счета..... | 17 |
| Входные параметры..... | 17 |
| Выходные параметры..... | 17 |
| Пример кода на С#..... | 17 |
| Пример кода на PHP..... | 18 |
| Отменить счет..... | 19 |
| Входные параметры..... | 19 |
| Пример кода на С#..... | 19 |
| Пример кода на PHP..... | 19 |
| Получить список оплат..... | 21 |
| Входные параметры..... | 21 |
| Выходные параметры..... | 21 |
| Пример кода на С#..... | 22 |
| Пример кода на PHP..... | 22 |
| Детальная информация об оплате..... | 24 |
| Входные параметры..... | 24 |
| Выходные параметры..... | 24 |

| | |
|---|----|
| Пример кода на С#..... | 25 |
| Пример кода на PHP..... | 25 |
| Интернет-эквайринг..... | 27 |
| Выставление счета по карте..... | 27 |
| Входные параметры..... | 27 |
| Выходные параметры..... | 28 |
| Пример кода на С#..... | 28 |
| Пример кода на PHP..... | 29 |
| Форма оплаты..... | 30 |
| Входные параметры..... | 30 |
| Выходные параметры..... | 30 |
| Пример кода на С#..... | 30 |
| Пример кода на PHP..... | 31 |
| Статус счета по карте..... | 32 |
| Входные параметры..... | 32 |
| Выходные параметры..... | 32 |
| Пример кода на С#..... | 32 |
| Пример кода на PHP..... | 33 |
| Отмена счета по карте..... | 34 |
| Входные параметры..... | 34 |
| Выходные параметры..... | 34 |
| Пример кода на С#..... | 34 |
| Пример кода на PHP..... | 35 |
| Уведомления на сайт..... | 36 |
| Параметры уведомления о платежах..... | 37 |
| Пример кода на С#..... | 39 |
| Пример кода на PHP..... | 40 |
| Цифровая подпись..... | 42 |
| Пример кода на С#..... | 45 |
| Пример кода на PHP..... | 48 |
| Тестовый стенд..... | 50 |
| Тестовые API ключи (токены)..... | 50 |
| Тестовые счета на оплату..... | 50 |
| Тестовые платежи..... | 51 |
| Тестовые счета на оплату банковской картой..... | 51 |
| Скачать исходные коды примеров..... | 53 |

ОПИСАНИЕ API

API предназначен для обеспечения взаимодействия производителей услуг с программно-техническим комплексом «Экспресс Платежи». Конфиденциальность передаваемых данных обеспечивается протоколом TLS 1.0. Аутентификация производителя услуг осуществляется по уникальному API-ключу (токену). Целостность передаваемых данных (защита от MITM-атак) обеспечивается с помощью цифровой подписи алгоритмом HMAC-SHA1, который описан в стандарте [RFC 2104](#). Дополнительная защита выполняется путем разграничения доступа IP-адресам программного комплекса производителя услуг.

Взаимодействие производителей услуг с программно-техническим комплексом «Экспресс Платежи» осуществляется на клиент-серверной технологии RESTful. Производитель услуг выступает в роли клиента, отправляет HTTP-запросы на сервер API. Возможно использование методов GET, POST, DELETE в HTTP-запросе. Данные передаются в кодировке UTF-8. В ответ на HTTP-запрос сервер отправляет HTTP-ответ клиенту в JSON-формате.

Формат запроса

Во всех запросах передаются обязательные параметры:

- *{version}* – версия API;
- *token={token}* – API-ключ (токен) доступа к серверу, который задается в личном кабинете;
- *signature={hash}* – цифровая подпись запроса, которая формируется на основании секретного слова (задается в личном кабинете) и передаваемых данных. Данное поле является опциональным.

Формат ответа

Любой ответ от сервера представлен в JSON-формате. В случае успешного выполнения запроса возвращается необходимый набор ключей и значений, предназначенный для конкретного ответа на запрос.

Пример ответа при успешном выполнении операции получения списка счетов:

```
i {
  "Items":[{
    "InvoiceNo":7,
    "AccountNo":"70",
    "Status":1,
    "Created":"20150101120000",
    "Expiration":"20150201"
    "Amount":100000.0,
    "Currency":974
  }, {
    "InvoiceNo":8,
```

```
"AccountNo": "80",
"Status": 3,
"Created": "20150201120000",
"Expiration": "20150301",
"Amount": 110000.0,
"Currency": 974
}]
}
```

В случае выполнения операции с ошибкой возвращается JSON-узел Error, содержащий:

- *Code* – код ошибки выполнения операции;
- *Msg* – сообщение об ошибке;
- *MsgCode* – код сообщения об ошибке.

Пример ответа при ошибке выполнения запроса представлен ниже:

```
i "Error": {
  "Code": 500,
  "Msg": "Отменить можно только тот счет, который находится в статусе
  \"Ожидание\"",
  "MsgCode": 5000000
}
```

Возможные коды ошибок о выполнении операций:

- 400 – некорректный запрос;
- 404 – ресурс не найден;
- 500 – внутренняя ошибка сервера.

Возможные коды сообщений выполнения операции:

- 4000003 – Некорректный запрос
- 4040001 – Платеж не найден
- 4040002 – Счет на оплату не найден
- 5000000 – Внутренняя ошибка по умолчанию

ВКЛЮЧЕНИЕ API

Для работы программного комплекса производителя услуг через API “Экспресс Платежи” необходимо в личном кабинете разрешить использование API. Для этого необходимо в личном кабинете:

1. Перейти в раздел настройки (пункт меню “Настройка”);
2. Перейти на вкладку “Услуги”
3. В списке услуг для нужной услуги перейти к настройкам API;
4. Разрешить использование API для услуги (см. рисунок 1).

Главная Услуги Счета Выставить счет Платежи **Настройки** Сводная информация

Настройки

Личный кабинет > Настройки > Услуги > API

API

Разрешить использование API

API ключ

Применять цифровую подпись ?

Секретное слово ?

Разрешенные IP-адреса для подключения (рекомендуется) ?

Рисунок 1 – Разрешение использования API для услуги

СИСТЕМА ЕРИП

Просмотр списка счетов по параметрам

Метод возвращает список выставленных счетов. Если входные параметры не заданы, то метод возвращает выставленные счета на последние 30 дней.

i Метод: GET

Адрес вызова: <https://api.express-pay.by/v1/invoices?token={Token}>

Входные параметры

| ПОЛЕ | ТИП | ОПИСАНИЕ |
|-----------|----------------------|--|
| Token* | String(32) | API-ключ производителя услуг |
| From | String(8) - yyyyMMdd | Дата оплаты с – начало периода |
| To | String(8) - yyyyMMdd | Дата оплаты по – конец периода |
| AccountNo | String(30) | Номер лицевого счета |
| Status | Integer | Статус счета на оплату: 1 - Ожидает оплату 2 - Просрочен 3 - Оплачен 4 - Оплачен частично 5 - Отменен |

* - обязательные поля

Выходные параметры

| ПОЛЕ | ТИП | ОПИСАНИЕ |
|-----------|------------|-----------------------|
| InvoiceNo | Integer | Номер счета на оплату |
| AccountNo | String(30) | Номер лицевого счета |

| | | |
|------------|---------------|--|
| Status | Integer | Статус счета на оплату: Ожидает оплату Просрочен Оплачен Оплачен частично Отменен |
| Created | String(14) | Дата выставления счета. Формат - ууууММддННммсс |
| Expiration | String(8) | Дата истечения срока действия выставлена счета на оплату. Формат - ууууММдд |
| Amount | Decimal(19,2) | Сумма счета на оплату |
| Currency | Integer | Код валюты |

Пример кода на C#

```

i private const string NumberInvoice = "1";

public static void GetListInvoices()
{
    var requestParams = new Dictionary<string, string>
    {
        {"Token", AppSettings.Token},

        //Параметры фильтра являются опциональными, по умолчанию возвращает
        значения за последние 30 дней
        {"AccountNo", NumberInvoice},
        {"From", "20000101" },
        {"To", "21000101" },
        {"Status", "1" }
    };

    var signature = SignatureHelper.Compute(requestParams, string.Empty, "get-
list-invoices");

    var url = AppSettings.ServiceUrl + "invoices?token=" + AppSettings.Token +
"&signature=" + signature + "&From=20000101&To=21000101&AccountNo=" +
NumberInvoice + "&Status=1";
    using (var client = new WebClient())
    {
        client.Encoding = AppSettings.DefaultEncoding;
        var data = client.DownloadString(url);

        Console.WriteLine(data);
    }
}

```


Пример кода на PHP

```
i $baseUrl = "https://sandbox-api.express-pay.by/v1/";

// Отправка GET запроса
function sendRequestGET($url) {
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_POST, 0);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    $response = curl_exec($ch);
    curl_close($ch);

    return $response;
}

// Просмотр списка счетов по параметрам
function getListInvoices($token, $fromDate = "", $toDate = "", $status = "",
    $accountNo = "") {
    $url = $baseUrl . "invoices?token=" . $token . "&From=" . $fromDate . "&To=" .
    $toDate . "&AccountNo=" . $accountNo . "&Status=" . $status;

    // Использование цифровой подписи
    // $requestParams = array(
    //     "Token" => $token,
    //     //Параметры фильтра являются опциональными, по умолчанию возвращает
    значения за последние 30 дней
    //     "AccountNo" => $accountNo,
    //     "From" => $fromDate,
    //     "To" => $toDate,
    //     "Status" => $status
    // );

    // $signature = computeSignature($requestParams, "", "get-list-invoices");
    // $url .= "&signature=" . $signature;

    return sendRequestGET($url);
}

// Пример запуска
echo getListInvoices("a75b74cbcfe446509e8ee874f421bd64");
```

Выставление счета

Метод выставляет новый счет.

i Метод: POST

Адрес вызова: <https://api.express-pay.by/v1/invoices?token={Token}>

Входные параметры

| ПОЛЕ | ТИП | ОПИСАНИЕ |
|-------------------|---------------|--|
| Token* | String(32) | API-ключ производителя услуг |
| AccountNo* | String(30) | Номер лицевого счета |
| Amount* | Decimal(19,2) | Сумма счета на оплату. Разделителем дробной и целой части является символ запятой |
| Currency* | Integer | Код валюты |
| Expiration | String(8) | Дата истечения срока действия выставлена счета на оплату. Формат - ууууММdd |
| Info | String(1024) | Назначение платежа |
| Surname | String(30) | Фамилия |
| FirstName | String(30) | Имя |
| Patronymic | String(30) | Отчество |
| FirstName | String(30) | Имя |
| Patronymic | String(30) | Отчество |
| City | String(30) | Город |
| Street | String(30) | Улица |
| House | String(18) | Дом |
| Building | String(10) | Корпус |
| Apartment | String(10) | Квартира |
| IsNameEditable | Integer | При оплате разрешено изменять ФИО плательщика 0 – нет, 1 – да |
| IsAddressEditable | Integer | При оплате разрешено изменять адрес плательщика 0 – нет, 1 – да |

| | | |
|-------------------|-------------|--|
| IsAmountEditable | Integer | При оплате разрешено изменять сумму оплаты 0 – нет, 1 – да |
| EmailNotification | String(255) | Адрес электронной почты, на который будет отправлено уведомление о выставлении счета |
| SmsPhone | String(13) | Номер мобильного телефона, на который будет отправлено SMS-сообщение о выставлении счета |

* - обязательные поля

Выходные параметры

| ПОЛЕ | ТИП | ОПИСАНИЕ |
|-----------|---------|-------------|
| InvoiceNo | Integer | Номер счета |

Пример кода на С#

```

i public static void AddInvoice()
{
    var requestParams = new Dictionary<string, string>
    {
        {"Token", AppSettings.Token},
        {"AccountNo", "123456"},
        {"Amount", "10"},
        {"Currency", "974"},
        {"Expiration", "20160505"},
        {"Info", "info"},
        {"Surname", "Ivanov"},
        {"FirstName", "Ivan"},
        {"Patronymic", "Ivanovich"},
        {"City", "Minsk"},
        {"Street", "Frunze"},
        {"Building", "2"},
        {"Apartment", "10"},
        {"IsNameEditable", "0"},
        {"IsAddressEditable", "0"},
        {"IsAmountEditable", "0"},
        {"EmailNotification", "ivanov@gmail.com"},
        {"SmsPhone", "+375291234567"},
    };

    var signature = SignatureHelper.Compute(requestParams, string.Empty, "add-
invoice");

    using (var client = new WebClient())
    {
        var url = AppSettings.ServiceUrl + "invoices?token=" + AppSettings.Token +
"&signature=" + signature;
        var response = client.UploadValues(url, new NameValueCollection
        {
            {"AccountNo", "123456"},
            {"Amount", "10"},
            {"Currency", "974"},
            {"Expiration", "20160505"},
            {"Info", "info"},
            {"Surname", "Ivanov"},
            {"FirstName", "Ivan"},
        }
    )
}

```

```

        {"Patronymic", "Ivanovich"},
        {"City", "Minsk"},
        {"Street", "Frunze"},
        {"Building", "2"},
        {"Apartment", "10"},
        {"IsNameEditable", "0"},
        {"IsAddressEditable", "0"},
        {"IsAmountEditable", "0"},
        {"EmailNotification", "ivanov@gmail.com"},
        {"SmsPhone", "+375291234567"},
    });

    var data = AppSettings.DefaultEncoding.GetString(response);
    Console.WriteLine(data);
}
}

```

Пример кода на PHP

```

i $baseUrl = "https://sandbox-api.express-pay.by/v1/";

// Отправка POST запроса
function sendRequestPOST($url, $params) {
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_POSTFIELDS, http_build_query($params));
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    $response = curl_exec($ch);
    curl_close($ch);

    return $response;
}

// Выставление счета
function addInvoice($token, $numberAccount, $amount, $currency, $expiration = "",
    $info = "", $surname = "", $firstName = "", $patronymic = "", $city = "", $street
    = "", $building = "", $apartment = "", $isNameEditable = "",
    $isAddressEditable = "", $isAmountEditable = "", $emailNotification = "",
    $smsPhone = "") {

    $url = $baseUrl . "invoices?token=" . $token;

    $requestParams = array(
        "Token" => $token,
        "AccountNo" => $numberAccount,
        "Amount" => $amount,
        "Currency" => $currency,
        "Expiration" => $expiration,
        "Info" => $info,
        "Surname" => $surname,
        "FirstName" => $firstName,
        "Patronymic" => $patronymic,
        "City" => $city,
        "Street" => $street,
        "Building" => $building,
    );
}

```

```
        "Apartment" => $apartment,
        "IsNameEditable" => $isNameEditable,
        "IsAddressEditable" => $isAddressEditable,
        "IsAmountEditable" => $isAmountEditable,
        "EmailNotification" => $emailNotification,
        "SmsPhone" => $smsPhone
    );

    // Использование цифровой подписи
    // $signature = computeSignature($requestParams, "", "add-invoice");
    // $url .= "&signature=" . $signature;

    return sendRequestPOST($url, $requestParams);
}

// Пример запуска
echo addInvoice("a75b74cbcf446509e8ee874f421bd64", 1, 10, 974);
```

Детальная информация о счете

Метод возвращает детальную информацию о существующем счете.

i Метод: GET

Адрес вызова: `http://api.express-pay.by/v1/invoices/{InvoiceNo}?token={Token}`

Входные параметры

| ПОЛЕ | ТИП | ОПИСАНИЕ |
|------------|------------|------------------------------|
| Token* | String(32) | API-ключ производителя услуг |
| InvoiceNo* | Integer | Номер счета на оплату |

* - обязательные поля

Выходные параметры

| ПОЛЕ | ТИП | ОПИСАНИЕ |
|------------|---------------|--|
| Status | Integer | Статус счета на оплату: 1 - Ожидает оплату 2 - Просрочен 3 - Оплачен 4 - Оплачен частично 5 - Отменен |
| Created | String(14) | Дата/Время выставления счета. Формат - ууууММддННммсс |
| Expiration | String(8) | Дата истечения срока действия выставлена счета на оплату. Формат - ууууММдд |
| Amount | Decimal(19,2) | Сумма счета на оплату |
| Currency | Integer | Код валюты |
| Info | String(1024) | Назначение платежа |
| Surname | String(30) | Фамилия |
| FirstName | String(30) | Имя |
| Patronymic | String(30) | Отчество |
| City | String(30) | Город |
| Street | String(30) | Улица |

| | | |
|-------------------|------------|--|
| House | String(18) | Дом |
| Building | String(10) | Корпус |
| Apartment | String(10) | Квартира |
| IsNameEditable | Integer | При оплате разрешено изменять ФИО плательщика 0 – нет, 1 – да |
| IsAddressEditable | Integer | При оплате разрешено изменять адрес плательщика 0 – нет, 1 – да |
| IsAmountEditable | Integer | При оплате разрешено изменять сумму оплаты 0 – нет, 1 – да |

Пример кода на C#

```

i private const string NumberInvoice = "1";

public static void GetDetailsInvoice()
{
    var requestParams = new Dictionary<string, string>
    {
        {"Token", AppSettings.Token},
        {"Id", NumberInvoice}
    };

    var signature = SignatureHelper.Compute(requestParams, string.Empty, "get-
details-invoice");

    using (var client = new WebClient())
    {
        client.Encoding = AppSettings.DefaultEncoding;
        var data =
            client.DownloadString(AppSettings.ServiceUrl + "invoices/" +
NumberInvoice + "?token=" + AppSettings.Token + "&signature=" +
signature);
        Console.WriteLine(data);
    }
}

```

Пример кода на PHP

```

i $baseUrl = "https://sandbox-api.express-pay.by/v1/";

// Отправка GET запроса
function sendRequestGET($url) {
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_POST, 0);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    $response = curl_exec($ch);
    curl_close($ch);

    return $response;
}

```

```
// Детальная информация о счете
function getDetailsInvoice($numberInvoice, $token) {
    $url = $baseUrl . "invoices/" . $numberInvoice . "?token=" . $token;

    // Использование цифровой подписи
    // $requestParams = array(
    //     "Token" => $token,
    //     "Id" => $numberInvoice
    // );

    // $signature = computeSignature($requestParams, "", "get-details-invoice");
    // $url .= "&signature=" . $signature;

    return sendRequestGET($url);
}

// Пример запуска
echo getDetailsInvoice(1, "a75b74cbcf446509e8ee874f421bd64");
```


Статус счета

Метод возвращает текущий статус счета.

i Метод: GET

Адрес вызова: `https://api.express-pay.by/v1/invoices/{InvoiceNo}/status?token={Token}`

Входные параметры

| ПОЛЕ | ТИП | ОПИСАНИЕ |
|------------|------------|------------------------------|
| Token* | String(32) | API-ключ производителя услуг |
| InvoiceNo* | Integer | Номер счета на оплату |

* - обязательные поля

Выходные параметры

| ПОЛЕ | ТИП | ОПИСАНИЕ |
|--------|---------|--|
| Status | Integer | Статус счета на оплату: 1 - Ожидает оплату 2 - Просрочен 3 - Оплачен 4 - Оплачен частично 5 - Отменен |

Пример кода на C#

```
i private const string NumberInvoice = "1";

public static void GetStatusInvoice()
{
    var requestParams = new Dictionary<string, string>
    {
        {"Token", AppSettings.Token},
        {"InvoiceId", NumberInvoice}
    };

    var signature = SignatureHelper.Compute(requestParams, string.Empty, "status-
invoice");
    var url = AppSettings.ServiceUrl + "invoices/" + NumberInvoice +
"/status?token=" + AppSettings.Token + "&signature=" + signature;
    using (var client = new WebClient())
    {
        client.Encoding = AppSettings.DefaultEncoding;
        var data = client.DownloadString(url);
        Console.WriteLine(data);
    }
}
```

```
}
```

Пример кода на PHP

```
i $baseUrl = "https://sandbox-api.express-pay.by/v1/";

// Отправка GET запроса
function sendRequestGET($url) {
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_POST, 0);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    $response = curl_exec($ch);
    curl_close($ch);

    return $response;
}

// Статус счета
function statusInvoice($numberInvoice, $token) {
    $url = $baseUrl . "invoices/" . $numberInvoice . "/status?token=" . $token;

    // Использование цифровой подписи
    // $requestParams = array(
    //     "Token" => $token,
    //     "Id" => $numberInvoice
    // );

    // $signature = computeSignature($requestParams, "", "status-invoice");
    // $url .= "&signature=" . $signature;

    return sendRequestGET($url);
}

// Пример запуска
echo statusInvoice(1, "a75b74cbcfe446509e8ee874f421bd64");
```

Отменить счет

Метод отменяет счет к оплате. Операция возможна только для счетов со статусом "Ожидает оплату".

i Метод: DELETE

Адрес вызова: `https://api.express-pay.by/v1/invoices/{InvoiceNo}?token={Token}`

Входные параметры

| ПОЛЕ | ТИП | ОПИСАНИЕ |
|------------|------------|------------------------------|
| Token* | String(32) | API-ключ производителя услуг |
| InvoiceNo* | Integer | Номер счета на оплату |

* - обязательные поля

Пример кода на C#

```
i private const string NumberInvoice = "1";

public static void CancelInvoice()
{
    var requestParams = new Dictionary<string, string>
    {
        {"Token", AppSettings.Token},
        {"Id", NumberInvoice}
    };

    var signature = SignatureHelper.Compute(requestParams, string.Empty, "cancel-
invoice");
    var url = AppSettings.ServiceUrl + "invoices/" + NumberInvoice + "?token=" +
AppSettings.Token + "&signature=" + signature;
    using (var client = new WebClient())
    {
        client.Encoding = AppSettings.DefaultEncoding;
        var data = client.UploadString(url, "DELETE", string.Empty);
        Console.WriteLine(data);
    }
}
```

Пример кода на PHP

```
i $baseUrl = "https://sandbox-api.express-pay.by/v1/";

// Отправка DELETE запроса
function sendRequestDELETE($url) {
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
```

```

curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "DELETE");
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
$response = curl_exec($ch);
curl_close($ch);

return $response;
}

// Отменить счет
function cancelInvoice($numberInvoice, $token) {
    $url = $baseUrl . "invoices/" . $numberInvoice . "?token=" . $token;

    // Использование цифровой подписи
    // $requestParams = array(
    //     "Token" => $token,
    //     "Id" => $numberInvoice
    // );

    // $signature = computeSignature($requestParams, "", "get-details-invoice");
    // $url .= "&signature=" . $signature;

    return sendRequestDELETE($url);
}

// Пример запуска
echo cancelInvoice(1, "a75b74cbcf446509e8ee874f421bd64");

```

Получить список оплат

Метод возвращает список полученных оплат. Если входные параметры запроса не заданы, то метод возвращает список полученных оплат за последние 30 дней.

i Метод: GET

Адрес вызова: <https://api.express-pay.by/v1/payments?token={Token}>

Входные параметры

| ПОЛЕ | ТИП | ОПИСАНИЕ |
|-----------|----------------------|--------------------------------|
| Token* | String(32) | API-ключ производителя услуг |
| From | String(8) - yyyyMMdd | Дата оплаты с – начало периода |
| To | String(8) - yyyyMMdd | Дата оплаты по – конец периода |
| AccountNo | String(30) | Номер лицевого счета |

* - обязательные поля

Выходные параметры

| ПОЛЕ | ТИП | ОПИСАНИЕ |
|------------|---------------|---|
| PaymentNo | Integer | Номер платежа |
| AccountNo | String(30) | Номер лицевого счета |
| Created | String(14) | Дата/Время платежа. Формат - ууууMMddHHmmss |
| Amount | Decimal(19,2) | Сумма платежа |
| Currency | Integer | Код валюты |
| Info | String(1024) | Назначение платежа |
| Surname | String(30) | Фамилия |
| FirstName | String(30) | Имя |
| Patronymic | String(30) | Отчество |
| City | String(30) | Город |
| Street | String(30) | Улица |

| | | |
|-----------|------------|----------|
| House | String(18) | Дом |
| Building | String(10) | Корпус |
| Apartment | String(10) | Квартира |

Пример кода на С#

```

i private const string NumberPayment = "1022";

public static void GetListPayments()
{
    var requestParams = new Dictionary<string, string>
    {
        {"Token", AppSettings.Token},

        // Параметры фильтра являются опциональными. По умолчанию возвращает
        // значения за последние 30 дней
        {"AccountNo", NumberPayment},
        {"From", "20000101" },
        {"To", "21000101" }
    };

    var signature = SignatureHelper.Compute(requestParams, string.Empty, "get-
list-payments");

    var url = AppSettings.ServiceUrl + "payments?token=" + AppSettings.Token +
"&signature=" + signature + "&From=20000101&To=21000101&AccountNo=" +
NumberPayment;

    using (var client = new WebClient())
    {
        client.Encoding = AppSettings.DefaultEncoding;
        var data = client.DownloadString(url);

        Console.WriteLine(data);
    }
}

```

Пример кода на PHP

```

i $baseUrl = "https://sandbox-api.express-pay.by/v1/";

// Отправка GET запроса
function sendRequestGET($url) {
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_POST, 0);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    $response = curl_exec($ch);
    curl_close($ch);

    return $response;
}

```

```

// Получить список оплат
function getListPayments($token, $fromDate = "", $toDate = "", $numberPayment =
"") {
    $url = $baseUrl . "payments?token=" . $token;

    // Использование цифровой подписи
    // $requestParams = array(
    //     "Token" => $token,
    //     //Параметры фильтра являются опциональными, по умолчанию возвращает
значения за последние 30 дней
    //     "AccountNo" => $numberPayment,
    //     "From" => $fromDate,
    //     "To" => $toDate,
    // );

    // $signature = computeSignature($requestParams, "", "get-list-payments");
    // $url .= "&signature=" . $signature;

    return sendRequestGET($url);
}

// Пример запуска
echo getListPayments("a75b74cbcfe446509e8ee874f421bd64");

```

Детальная информация об оплате

Метод возвращает детальную информацию о полученной оплате.

i Метод: GET

Адрес вызова: <https://api.express-pay.by/v1/payments/{PaymentNo}?token={Token}>

Входные параметры

| ПОЛЕ | ТИП | ОПИСАНИЕ |
|------------|------------|------------------------------|
| Token* | String(32) | API-ключ производителя услуг |
| PaymentNo* | Integer | Номер платежа |

* - обязательные поля

Выходные параметры

| ПОЛЕ | ТИП | ОПИСАНИЕ |
|------------|---------------|---|
| AccountNo | String(30) | Номер лицевого счета |
| Created | String(14) | Дата/Время платежа. Формат - ууууMMddHHmmss |
| Amount | Decimal(19,2) | Сумма платежа |
| Currency | Integer | Код валюты |
| Info | String(1024) | Назначение платежа |
| Surname | String(30) | Фамилия |
| FirstName | String(30) | Имя |
| Patronymic | String(30) | Отчество |
| City | String(30) | Город |
| Street | String(30) | Улица |
| House | String(18) | Дом |
| Building | String(10) | Корпус |
| Apartment | String(10) | Квартира |

Пример кода на C#

```
i private const string NumberPayment = "1022";

public static void GetDetailsPayment()
{
    var requestParams = new Dictionary<string, string>
    {
        {"Token", AppSettings.Token},
        {"Id", NumberPayment}
    };

    var signature = SignatureHelper.Compute(requestParams, string.Empty, "get-
details-payment");

    using (var client = new WebClient())
    {
        client.Encoding = AppSettings.DefaultEncoding;
        var data =
            client.DownloadString(AppSettings.ServiceUrl + "payments/" +
NumberPayment + "?token=" + AppSettings.Token + "&signature=" +
signature);

        Console.WriteLine(data);
    }
}
```

Пример кода на PHP

```
i $baseUrl = "https://sandbox-api.express-pay.by/v1/";

// Отправка GET запроса
function sendRequestGET($url) {
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_POST, 0);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    $response = curl_exec($ch);
    curl_close($ch);

    return $response;
}

// Детальная информация об оплате
function getDetailPayment($token, $numberPayment) {
    $url = $baseUrl . "payments/" . $numberPayment . "?token=" . $token;

    // Использование цифровой подписи
    // $requestParams = array(
    //     "Token" => $token,
    //     "PaymentNo" => $numberPayment
    // );

    // $signature = computeSignature($requestParams, "", "get-details-payment");
    // $url .= "&signature=" . $signature;

    return sendRequestGET($url);
}
```

```
}  
  
// Примеры запуска  
echo getDetailPayment("a75b74cbcfe446509e8ee874f421bd64", 1);
```

ИНТЕРНЕТ-ЭКВАЙРИНГ

Выставление счета по карте

Метод выставляет новый счет для оплаты через банковскую карту.

i Метод: POST

Адрес вызова: `https://api.express-pay.by/v1/cardinvoices?token={Token}`

Входные параметры

| ПОЛЕ | ТИП | ОПИСАНИЕ |
|--------------------|---------------|---|
| Token * | String(32) | API-ключ производителя услуг |
| AccountNo * | String(30) | Номер лицевого счета |
| Expiration | String(8) | Дата истечения срока действия выставленного счета на оплату. Формат - ууууММдд |
| Amount * | Decimal(19,2) | Сумма счета на оплату. Сумма счета должна быть не менее 1,00 BYN. Разделителем дробной и целой части является символ запятой. |
| Currency * | Integer | Код валюты |
| Info * | String(1024) | Назначение платежа |
| ReturnUrl * | String(512) | Адрес для перенаправления пользователя в случае успешной оплаты |
| FailUrl * | String(512) | Адрес для перенаправления пользователя в случае неуспешной оплаты |
| Language | String(2) | Язык в кодировке ISO 639-1. Если не указан, будет использован язык по умолчанию |
| SessionTimeoutSecs | Integer | Продолжительность сессии в секундах. |
| ExpirationDate | String(32) | Время жизни заказа. Формат ууууММддННммсс. Если этот параметр не передается в запросе, то для определения времени жизни сессии используется SessionTimeoutSecs. |
| | | |

* - обязательные поля

Выходные параметры

| ПОЛЕ | ТИП | ОПИСАНИЕ |
|---------------|-------------|---|
| CardInvoiceId | Integer | Номер счета по карте |
| ErrorCode | Integer | Номер ошибки. Отсутствует, если ошибки нет |
| ErrorMessage | String(512) | Сообщение об ошибке. Отсутствует, если ошибки нет |

Пример кода на С#

```
i public static void AddCardInvoice()
{
    var requestParams = new Dictionary
    {
        {"Token", AppSettings.Token},
        {"AccountNo", "123456"},
        {"Expiration", "20161224"},
        {"Amount", "10"},
        {"Currency", "933"},
        {"Info", "info"},
        {"ReturnUrl", "https://example.com/success"},
        {"FailUrl", "https://example.com/fail"},
        {"Language", "ru"},
        {"SessionTimeoutSecs", "2000"},
        {"ExpirationDate", "20161224235001"},
    };
    var signature = SignatureHelper.Compute(requestParams, string.Empty, "add-
card-invoice");
    using (var client = new WebClient())
    {
        var url = AppSettings.ServiceUrl + "cardinvoices?token=" +
AppSettings.Token + "&signature=" + signature;
        var response = client.UploadValues(url, new NameValueCollection
        {
            {"Token", AppSettings.Token},
            {"AccountNo", "123456"},
            {"Expiration", "20161224"},
            {"Amount", "10"},
            {"Currency", "933"},
            {"Info", "info"},
            {"ReturnUrl", "https://example.com/success"},
            {"FailUrl", "https://example.com/fail"},
            {"Language", "ru"},
            {"SessionTimeoutSecs", 2000},
            {"ExpirationDate", "20161224235001"},
        });
        var data = AppSettings.DefaultEncoding.GetString(response);
        Console.WriteLine(data);
    }
}
```

Пример кода на PHP

```
i $baseUrl = "https://api.express-pay.by/v1/";

// Отправка POST запроса
function sendRequestPOST($url, $params) {
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_POSTFIELDS, http_build_query($params));
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    $response = curl_exec($ch);
    curl_close($ch);
    return $response;
}

//Выставление счета по карте
function addInvoiceByCard($token, $accountNo, $amount, $currency, $info,
$returnUrl, $failUrl, $expiration="", $language="", $pageView="",
$sessionTimeoutSecs="", $expirationDate="") {
    $url = $baseUrl . "cardinvoices?token=" . $token;
    $requestParams = array(
        "Token" => $token,
        "AccountNo" => $accountNo,
        "Expiration" => $expiration,
        "Amount" => $amount,
        "Currency" => $currency,
        "Info" => $info,
        "ReturnUrl" => $returnUrl,
        "FailUrl" => $failUrl,
        "Language" => $language,
        "SessionTimeoutSecs" => $sessionTimeoutSecs,
        "ExpirationDate" => $expirationDate
    );

    // Использование цифровой подписи
    // $signature = computeSignature($requestParams, "", "add-card-invoice");
    // $url .= "&signature=" . $signature;
    return sendRequestPOST($url, $requestParams);
}

// Пример использования
echo addInvoiceByCard("a75b74cbcfe446509e8ee874f421bd64", "123456", "20161224",
10, 933, "info", "https://example.com/success", "https://example.com/fail", "ru",
2000, "20161224235001");
```

Форма оплаты

Метод инициирует оплату по банковской карте и возвращает адрес для формы ввода данных банковской карты. Для оплаты выставленного счета необходимо передать в метод CardInvoiceNo, полученный при вызове метода “Выставление счета по карте”.

После ввода данных банковской карты и успешного проведения платежа пользователь будет перенаправлен на адрес returnUrl, указанный при выставлении счета по карте. В случае ошибки проведения платежа пользователь будет перенаправлен на адрес failurl.

i Метод: GET

Адрес вызова: `https://api.express-pay.by/v1/cardinvoices/{CardInvoiceNo}/payment?token={Token}`

Входные параметры

| ПОЛЕ | ТИП | ОПИСАНИЕ |
|-----------------|------------|--|
| Token * | String(32) | API-ключ производителя услуг |
| CardInvoiceId * | Integer | Номер счета по карте, полученный с помощью метода “Выставление счета по карте” |

* - обязательные поля

Выходные параметры

| ПОЛЕ | ТИП | ОПИСАНИЕ |
|--------------|-------------|---|
| FormUrl | String(512) | Адрес страницы ввода данных банковской карты |
| ErrorCode | Integer | Номер ошибки. Отсутствует, если ошибки нет |
| ErrorMessage | String(512) | Сообщение об ошибке. Отсутствует, если ошибки нет |

Пример кода на C#

```
i private const string CardNumberInvoice = "1674";
public static void PaymentCardInvoice()
{
    var requestParams = new Dictionary
    {
        {"Token", AppSettings.Token},
        {"CardInvoiceId", CardNumberInvoice}
    };
    var signature = SignatureHelper.Compute(requestParams, string.Empty, "card-
invoice-form");
    using (var client = new WebClient())
    {
```

```

    var url = AppSettings.ServiceUrl + "cardinvoices/" + CardNumberInvoice +
"/payment?token="+AppSettings.Token+"&signature="+signature;
    using (var client = new WebClient())
    {
        client.Encoding = AppSettings.DefaultEncoding;
        var data = client.DownloadString(url);
        Console.WriteLine(data);
    }
}
}

```

Пример кода на PHP

```

i private static $baseUrl = "https://api.express-pay.by/v1/";
// Отправка GET запроса
private static function sendRequestGET($url) {
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_POST, 0);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    $response = curl_exec($ch);
    curl_close($ch);
    return $response;
}
// Форма оплаты
function getPaymentForm($cardNumberInvoice) {
    $url = $baseUrl . "cardinvoices/" . $cardNumberInvoice . "/payment";
    // Использование цифровой подписи
    // $requestParams = array(
    //     "Token" => $token,
    //     "CardInvoiceId" => $cardInvoiceNo
    // );
    // $signature = computeSignature($requestParams, "", "card-invoice-form");
    // $url .= "&signature=" . $signature;
    return sendRequestGET($url);
}
//Пример использования
echo getPaymentFormat("1674");

```

Статус счета по карте

Метод возвращает статус счета по карте.

i Метод: GET

Адрес вызова: `https://api.express-pay.by/v1/cardinvoices/{CardInvoiceNo}/status?token={Token}`

Входные параметры

| ПОЛЕ | ТИП | ОПИСАНИЕ |
|-----------------|------------|---|
| Token * | String(32) | API-ключ производителя услуг |
| CardInvoiceId * | Integer | Номер счета по карте, полученный с помощью метода "Выставление счета по карте" |
| Language | String(2) | Язык в кодировке ISO 639-1. Если не указан, считается, что язык – русский. Сообщение об ошибке будет возвращено именно на этом языке. |

* - обязательные поля

Выходные параметры

| ПОЛЕ | ТИП | ОПИСАНИЕ |
|-------------------|-------------|---|
| Amount | Integer | Сумма оплаты |
| CardInvoiceStatus | Integer | Состояние оплаты: 0 - Счет зарегистрирован, но не оплачен 1 - Предавторизованная сумма захолдирована (для двухстадийных платежей) 2 - Проведена полная авторизация суммы счета 3 - Авторизация отменена 4 - По транзакции была проведена операция возврата 5 - Инициирована авторизация через ACS банка-эмитента 6 - Авторизация отклонена |
| ErrorCode | Integer | Номер ошибки. Отсутствует, если ошибки нет |
| ErrorMessage | String(512) | Сообщение об ошибке. Отсутствует, если ошибки нет |

Пример кода на C#

```
i private const string CardNumberInvoice = "1674";  
public static void StatusCardInvoice()  
{  
    var requestParams = new Dictionary  
    {
```



```

        {"Token", AppSettings.Token},
        {"CardInvoiceId", CardNumberInvoice},
        {"Language", "ru"},
    };
    var signature = SignatureHelper.Compute(requestParams, string.Empty, "status-
card-invoice");
    using (var client = new WebClient())
    {
        var url = AppSettings.ServiceUrl + "cardinvoices/" + CardNumberInvoice +
"/status?token=" + AppSettings.Token + "&signature=" + signature;
        using (var client = new WebClient())
        {
            client.Encoding = AppSettings.DefaultEncoding;
            var data = client.DownloadString(url);
            Console.WriteLine(data);
        }
    }
}

```

Пример кода на PHP

```

i private static $baseUrl = "https://api.express-pay.by/v1/";
// Отправка GET запроса
private static function sendRequestGET($url) {
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_POST, 0);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    $response = curl_exec($ch);
    curl_close($ch);
    return $response;
}
// Статус счета по карте
function getStatusCardInvoice($token, $cardInvoiceNo, $language="") {
    $url = $baseUrl . "cardinvoices/" . $cardInvoiceNo . "/status?token=" .
$token;
    // Использование цифровой подписи
    // $requestParams = array(
    //     "Token" => $token,
    //     "CardInvoiceId" => $cardInvoiceNo,
    //     "Language" => "ru"
    // );
    // $signature = computeSignature($requestParams, "", "status-card-invoice");
    // $url .= "&signature=" . $signature;
    return sendRequestGET($url);
}
//Пример использования
echo getStatusCardInvoice("a75b74cbcf446509e8ee874f421bd64", "1674");

```

Отмена счета по карте

Метод отменяет оплату счета по карте. Использование операции отмены возможно до 23:59:59 дня проведения операции (не позднее). Для разрешения использования данной операции в анкете реквизитов услуги ЕРИП в разделе "Предоставление возможности плательщику отменить платеж" необходимо выставить опцию "Разрешить до перечисления денежных средств банком".

i Метод: POST

Адрес вызова: `https://api.express-pay.by/v1/cardinvoices/{CardInvoiceNo}/reverse?token={Token}`

Входные параметры

| ПОЛЕ | ТИП | ОПИСАНИЕ |
|-----------------|------------|--|
| Token * | String(32) | API-ключ производителя услуг |
| CardInvoiceId * | Integer | Номер счета по карте, полученный с помощью метода "Выставление счета по карте" |

* - обязательные поля

Выходные параметры

| ПОЛЕ | ТИП | ОПИСАНИЕ |
|--------------|-------------|---|
| ErrorCode | Integer | Номер ошибки. Отсутствует, если ошибки нет |
| ErrorMessage | String(512) | Сообщение об ошибке. Отсутствует, если ошибки нет |

Пример кода на C#

```
i private const string CardNumberInvoice = "1674";
public static void ReverseCardInvoice()
{
    var requestParams = new Dictionary
    {
        {"Token", AppSettings.Token},
        {"CardInvoiceId", CardNumberInvoice}
    };
    var signature = SignatureHelper.Compute(requestParams, string.Empty, "reverse-card-invoice");
    using (var client = new WebClient())
    {
        var url = AppSettings.ServiceUrl + "cardinvoices/" + CardNumberInvoice +
            "/reverse?token=" + AppSettings.Token + "&signature=" + signature;
        using (var client = new WebClient())
        {
            client.Encoding = AppSettings.DefaultEncoding;
            var data = client.DownloadString(url);
            Console.WriteLine(data);
        }
    }
}
```

```
}  
}  
}
```

Пример кода на PHP

```
i $baseUrl = "https://api.express-pay.by/v1/";  
// Отправка POST запроса  
function sendRequestPOST($url, $params) {  
    $ch = curl_init();  
    curl_setopt($ch, CURLOPT_URL, $url);  
    curl_setopt($ch, CURLOPT_POST, 1);  
    curl_setopt($ch, CURLOPT_POSTFIELDS, http_build_query($params));  
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);  
    $response = curl_exec($ch);  
    curl_close($ch);  
    return $response;  
}  
// Отмена счета по карте  
function reverseCardInvoice($token, $cardInvoiceNo) {  
    $url = $baseUrl . "cardinvoices/" . $cardInvoiceNo . "/reverse?token=" .  
    $token;  
  
    $requestParams = array(  
        "Token" => $token,  
        "CardInvoiceId" => $cardInvoiceNo  
    );  
  
    // Использование цифровой подписи  
    // $signature = computeSignature($requestParams, "", "reverse-card-invoice");  
    // $url .= "&signature=" . $signature;  
    return sendRequestPOST($url, $requestParams);  
}  
//Пример использования  
echo reverseCardInvoice("a75b74cbcfe446509e8ee874f421bd64", "1674");
```

УВЕДОМЛЕНИЯ НА САЙТ

Уведомления на сайт предназначены для вызова специального адреса сайта производителя услуг. При использовании такого способа получения уведомлений нет необходимости постоянно опрашивать поступившие платежи через API. Такой способ получения уведомлений снижает нагрузку на серверы и увеличивает скорость доставки уведомлений о платежах.

Вызов адреса сайта происходит в момент совершения платежа в адрес производителя услуг или при изменении статуса счета на оплату. Вызываемый адрес сайта указывается в настройках личного кабинета.

Для этого необходимо выполнить следующие операции:

1. Перейти в раздел настройки (пункт меню “Настройка”);
2. Перейти на вкладку “Услуги”
3. В списке услуг для нужной услуги перейти на “Уведомления”;
4. Установить опцию “Получать уведомления об оплате на URL”;
5. Указать адрес вызова сайта производителя услуг в поле “URL для уведомлений” вида **https://mydomain.by/express-pay/notification** (см. рисунок).

Настройки

Личный кабинет > Настройки > Услуги > Уведомления

Уведомления

Включить доставку уведомлений по SMS

Получать уведомления по SMS на

Включить доставку уведомлений по электронной почте

Получать уведомления по электронной почте на

Получать уведомления об оплате на URL

URL для уведомлений

Применять цифровую подпись ?

Секретное слово ?

Таймаут ожидания ответа секунд

Тестовое уведомление на URL

Сохранить **Отменить**

Рисунок 2 – Настройка адреса адрес сайт для доставки уведомления на сайт

Во всех запросах передаются обязательные параметры:

- *Data* – информация о платеже в JSON-формате (поступление нового платежа, отмена платежа, изменение статуса счета);
- *Signature={hash}* – цифровая подпись запроса, которая формируется на основании секретного слова (задается в личном кабинете) и передаваемых данных. Данный параметр передается в случае установки опции “Применять цифровую подпись”.

Сайт производителя услуг должен ответить HTTP-кодом 200 в случае успешной обработки запроса. Если сайт производителя услуг вернет код ответа отличный от 200 или истечет таймаут ожидания ответа, то будет выполнена повторная попытка доставки уведомления. Для обеспечения гарантированной доставки уведомления выполняется 3 дополнительных попытки доставки уведомлений. Первая попытка доставки выполняется через 3 минуты, вторая через 30 минут, третья через 90 минут.

В случае не успешных попыток доставки всех 4-х уведомлений по платежу на адрес электронной почты производителя услуг направляется письмо с отчетом об ошибочных доставках.

При включении опции “Применять цифровую подпись” будет формироваться цифровая подпись для передаваемых данных, которая позволяет контролировать целостность передаваемых данных (защита от MITM-атак). Цифровая подпись основана на алгоритме HMAC-SHA1, который описан в стандарте [RFC 2104](#).

Параметр “Таймаут ожидания ответа” устанавливает время, которое сервер “Экспресс Платежи” будет ожидать ответа от сайта производителя услуг. Если в течение данного времени сервер производителя услуг не вернет ответ с HTTP-кодом 200, то попытка доставки уведомления считается неудачной.

Параметры уведомления о платежах

| ПОЛЕ | ТИП | ОПИСАНИЕ |
|-----------|------------|--|
| CmdType | Integer | Тип уведомления: 1 - Поступление нового платежа 2 - Отмена платежа 3 - Изменение статуса счета |
| Status | Integer | Статус счета на оплату: 1 - Ожидает оплату 2 - Просрочен 3 - Оплачен 4 - Оплачен частично 5 - Отменен Оptionальное поле. Возвращается только в уведомления типа 3. |
| AccountNo | String(30) | Номер лицевого счета |

| | | |
|-----------|---------------|---|
| InvoiceNo | String(30) | Номер измененного счета в системе "Экспресс Платежи" |
| PaymentNo | Integer | Номер платежа |
| Amount | Decimal(19,2) | Сумма оплаты (разделитель между целой и дробной частью ",") |
| Created | String(14) | Дата оплаты (Формат - yyyyMMddHHmmss) |
| Service | String(258) | Название услуги |
| Payer | String(258) | ФИО плательщика |
| Address | String(258) | Адрес плательщика |

Пример передаваемых данных при получении нового платежа:

```

i
{
  "CmdType":1,
  "PaymentNo":1082,
  "AccountNo":1024,
  "Amount":"20000",
  "Created":"20160217122109",
  "Service":"mydomain.by",
  "Payer": "",
  "Address":""
}

```

Пример передаваемых данных при отмене платежа:

```

i
{
  "CmdType":2,
  "PaymentNo":1082,
  "AccountNo":1024,
  "Amount":"20000",
  "Created":"20160217122203",
}

```

```
"Service": "mydomain.by",
"Payer": "",
"Address": ""
}
```

Пример передаваемых данных при изменении статуса счета:

```
i
{
  "CmdType": 3,
  "Status": 3,
  "AccountNo": "147221",
  "InvoiceNo": 17645,
  "Amount": "16",
  "Created": "20161130125859",
  "Service": "10ballov.by",
  "Payer": "",
  "Address": ""
}
```

При нажатии кнопки “Тестовое уведомление на URL” выполняется доставка тестового уведомления о платеже на адрес производителя услуг, по которому можно проверить прием платежа сайтом производителя услуг.

Пример кода на C#

```
i namespace EripNotify.Models
{
    public class EripNotify
    {
        public string Data { get; set; }
        public string Signature { get; set; }
    }
}
namespace EripNotify.Controllers
{
    public class EripNotifyController : ApiController
    {
        private static readonly Encoding HashEncoding = Encoding.UTF8;
        private string _secretWord = "secretWord";
        private bool _useSignature = false;
        [HttpPost]
        public IHttpActionResult Test(Models.EripNotify model)
        {
            if (_useSignature)
```

```

    {
        // Проверяем цифровую подпись
        if (model.Signature != ComputeSignature(model.Data, _secretWord))
        {
            // NOTE: Добавить обработку ошибки
            // ...
            return InternalServerError();
        }
    }
    // Преобразуем из JSON в Object
    var obj = JObject.Parse(model.Data);
    // NOTE: Выполняем действия с полученным объектом
    // ...
    return Ok();
}
// Функция генерации и проверки цифровой подписи
public string ComputeSignature(string json, string secretWord)
{
    var hmac = string.IsNullOrEmpty(secretWord) ?
        new HMACSHA1(HashEncoding.GetBytes(string.Empty)) :
        new HMACSHA1(HashEncoding.GetBytes(secretWord));
    var hash = hmac.ComputeHash(HashEncoding.GetBytes(json));
    var result = new StringBuilder();
    foreach (var item in hash)
        result.Append(item.ToString("X2"));
    return result.ToString();
}
}
}
}

```

Пример кода на PHP

```

i // Секретное слово указывается в настройках личного кабинета
$secretWord = "secretWord";
// Использование цифровой подписи указывается в настройках личного кабинета
$isUseSignature = true;
// Обработка POST запроса
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $data = $_POST['Data'];
    $signature = $_POST['Signature'];

    // Проверяем использование цифровой подписи
    if($isUseSignature) {
        // Проверяем цифровую подпись
        if($signature == computeSignature($data, $secretWord)) {
            // Преобразуем из JSON в Object
            $data = json_decode($data);
            // NOTE: Добавить обработку данных о полученном платеже
            // ...
            echo "success";
        } else {
            echo "fail signature";
        }
    }
} else {
    // Без использования цифровой подписи
    // Преобразуем из JSON в Object
    $data = json_decode($data);
}
}

```



```
        // NOTE: Добавить обработку данных о полученном платеже
        // ...
        echo "success";
    }
}
// Функция генерации и проверки цифровой подписи
function computeSignature($json, $secretWord) {
    $hash = NULL;

    if (empty(trim($secretWord)))
        $hash = strtoupper(hash_hmac('sha1', $json, ""));
    else
        $hash = strtoupper(hash_hmac('sha1', $json, $secretWord));
    return $hash;
}
```

ЦИФРОВАЯ ПОДПИСЬ

Формирование цифровой подписи для передаваемых данных обеспечивает целостность информации и гарантирует, что передаваемые данные не были изменены посторонними лицами в процессе передачи. Цифровая подпись данных осуществляется с помощью алгоритма HMAC-SHA1, который описан в стандарте [RFC 2104](#).

При использовании цифровой подписи для запросов на сервер необходимо объединить все параметры запроса в одну строку в строго определенном порядке. Для объединенной строки необходимо рассчитать цифровую подпись, которая указывается при запросе в параметре *signature*. Сервер при получении входящего запроса выполняет проверку цифровой подписи (параметра *signature*) и параметров запроса.

Пример формирования строки запроса при использовании цифровой подписи для получения списка счетов:

<http://api.express-pay.by/v1/invoices?token={Token}&signature={Hash}>.

Для усиления безопасности возможно формирование цифровой подписи с помощью секретного слова, которое известно только серверу и клиенту. Чтобы задать секретное слово, необходимо выполнить следующие операции в личном кабинете:

6. Перейти в раздел настройки (пункт меню “Настройка”);
7. Перейти на вкладку “Услуги”
8. В списке услуг для нужной услуги перейти к настройкам API;
9. Укажите секретное слово и никому его не передавайте (см. рисунок).

Настройки
Личный кабинет > Настройки > Услуги > API

API

Разрешить использование API

API ключ

Применять цифровую подпись ?

Секретное слово ?

Разрешенные IP-адреса для подключения (рекомендуется) ?

Рисунок 3 – Использование цифровой подписи и задание секретного слова

В случае успешной проверки выполняется обработка запроса. Если параметры запроса не соответствуют цифровой подписи, то отправляется уведомление об ошибке и запрос не обрабатывается сервером.

Порядок следования параметров запросов следующий:

1. Добавление счета

```
i "token",  
    "accountno",  
    "amount",  
    "currency",  
    "expiration",  
    "info",  
    "surname",  
    "firstname",  
    "patronymic",  
    "city",  
    "street",  
    "house",  
    "building",  
    "apartment",  
    "isnameeditable",  
    "isaddresseditable",  
    "isamounteditable"
```

2. Детальная информация счета

```
i "token",  
    "id"
```

3. Отменить счет

```
i "token",  
    "id"
```

4. Статус счета

```
i "token",  
    "invoiceid"
```

5.Список счетов

i "token",
"from",
"to",
"accountno",
"status"

6.Список платежей

i "token",
"from",
"to",
"accountno"

7.Детали по платежу

i "token",
"id"

8.Выставление счета по карте

i "token",
"accountno",
"expiration",
"amount",
"currency",
"info",
"returnurl",
"failurl",
"language",
"pageview",
"sessiontimeoutsecs",
"expirationdate"

9.Форма оплаты

i "token",
"cardinvoiceid"

10.Статус счета по карте

i "token",
" cardinvoiceid ",

```
"language"
```

11.Отмена счета по карте

```
i "token",  
" cardinvoiceid "
```

Примечание: Разделителем целой и дробной части в сумме (параметр amount) является запятая.

Пример кода на С#

```
i namespace ExPay.Api  
{  
    public static class AppSettings  
    {  
        public const string ServiceUrl = "https://api.express-pay.by/v1/";  
        public const string Token = "a75b74cbcf446509e8ee874f421bd63"; // API-  
        ключ  
        public static readonly Encoding DefaultEncoding = Encoding.UTF8;  
    }  
  
    public static class SignatureHelper  
    {  
        private static readonly Encoding HashEncoding = Encoding.UTF8;  
  
        // Порядок следования полей при вычислении цифровой подписи.  
        // Внимание! Для корректной работы порядок изменять нельзя  
        private static readonly Dictionary<string, string[]> Mapping = new  
        Dictionary<string, string[]>  
        {  
            {  
                "add-invoice", new[]  
                {  
                    "token",  
                    "accountno",  
                    "amount",  
                    "currency",  
                    "expiration",  
                    "info",  
                    "surname",  
                    "firstname",  
                    "patronymic",  
                    "city",  
                    "street",  
                    "house",  
                    "building",  
                    "apartment",  
                    "isnameeditable",  
                    "isaddresseditable",  
                    "isamounteditable"  
                }  
            }  
        }  
    }  
}
```

```

    },
    {
      "get-details-invoice", new[]
      {
        "token",
        "id"
      }
    },
    {
      "cancel-invoice", new[]
      {
        "token",
        "id"
      }
    },
    {
      "status-invoice", new[]
      {
        "token",
        "invoiceid"
      }
    },
    {
      "get-list-invoices", new[]
      {
        "token",
        "from",
        "to",
        "accountno",
        "status"
      }
    },
    {
      "get-list-payments", new[]
      {
        "token",
        "from",
        "to",
        "accountno"
      }
    },
    {
      "get-details-payment", new[]
      {
        "token",
        "id"
      }
    },
    {
      "add-card-invoice", new[]
      {
        "token",
        "id"
      }
    },
    {
      "card-invoice-form", new[]
      {
        "token",
        "cardinvoiceid"
      }
    }
  }
}

```

```

    },
    {
        "status-card-invoice", new[]
        {
            "token",
            "cardinvoiceid",
            "language"
        }
    },
    {
        "reverse-card-invoice", new[]
        {
            "token",
            "cardinvoiceid"
        }
    }
};

public static string Compute(Dictionary<string, string> requestParams,
string secretWord, string action)
{
    var normalizedParams = requestParams
        .ToDictionary(k => k.Key.ToLower(), v => v.Value);

    var cmdFields = Mapping[action];

    var builder = new StringBuilder();
    foreach (var cmdField in cmdFields)
    {
        if (normalizedParams.ContainsKey(cmdField))
            builder.Append(normalizedParams[cmdField]);
    }

    HMACSHA1 hmac;
    if (string.IsNullOrEmpty(secretWord))
    {
        // в алгоритме всегда должно быть задан ключ шифрования. Если исп.
        конструктор по умолчанию, то ключ генерится автоматически,
        // что нам не подходит
        hmac = new HMACSHA1(HashEncoding.GetBytes(string.Empty));
    }
    else
    {
        hmac = new HMACSHA1(HashEncoding.GetBytes(secretWord));
    }

    var hash = hmac.ComputeHash(
        HashEncoding.GetBytes(builder.ToString()));

    var result = new StringBuilder();
    foreach (var item in hash)
    {
        result.Append(item.ToString("X2"));
    }

    return result.ToString();
}
}
}

```

Пример кода на PHP

```
i // Формирование цифровой подписи
public static function computeSignature($requestParams, $secretWord, $method) {
    $normalizedParams = array_change_key_case($requestParams, CASE_LOWER);
    $mapping = array(
        "add-invoice" => array(
            "token",
            "accountno",
            "amount",
            "currency",
            "expiration",
            "info",
            "surname",
            "firstname",
            "patronymic",
            "city",
            "street",
            "house",
            "building",
            "apartment",
            "isnameeditable",
            "isaddresseditable",
            "isamounteditable"),
        "get-details-invoice" => array(
            "token",
            "id"),
        "cancel-invoice" => array(
            "token",
            "id"),
        "status-invoice" => array(
            "token",
            "invoiceid"),
        "get-list-invoices" => array(
            "token",
            "from",
            "to",
            "accountno",
            "status"),
        "get-list-payments" => array(
            "token",
            "from",
            "to",
            "accountno"),
        "get-details-payment" => array(
            "token",
            "id"),
        "add-card-invoice" => array(
            "token",
            "id"),
        "card-invoice-form" => array(
            "token",
            "cardinvoiceid"),
        "status-card-invoice" => array(
            "token",
            "cardinvoiceid",
            "language"),
        "reverse-card-invoice" => array(
```



```
        "token",
        "cardinvoiceid")
);

$apiMethod = $mapping[$method];
$result = "";

foreach ($apiMethod as $item){
    $result .= $normalizedParams[$item];
}

$hash = strtoupper(hash_hmac('sha1', $result, $secretWord));

return $hash;
}
```

ТЕСТОВЫЙ СТЕНД

При тестировании API вместо адреса <https://api.express-pay.by> необходимо использовать адрес тестового стенда <https://sandbox-api.express-pay.by>. В качестве тестовых данных используйте данные, приведенные ниже.

i Адрес вызова: <https://sandbox-api.express-pay.by/v1/>

Тестовые API ключи (токены)

| НОМЕР УСЛУГИ | API КЛЮЧ | РАЗРЕШИТЬ ИСПОЛЬЗОВАНИЕ API | ПРИМЕНЯТЬ ЦИФРОВУЮ ПОДПИСЬ | СЕКРЕТНОЕ СЛОВО |
|--------------|----------------------------------|-----------------------------|----------------------------|------------------------|
| 1 | a75b74cbcfe446509e8ee874f421bd63 | Нет | Нет | |
| 2 | a75b74cbcfe446509e8ee874f421bd64 | Да | Нет | |
| 3 | a75b74cbcfe446509e8ee874f421bd65 | Да | Да | |
| 4 | a75b74cbcfe446509e8ee874f421bd66 | Да | Да | sandbox.express-pay.by |

Тестовые счета на оплату

| НОМЕР УСЛУГИ | API КЛЮЧ | НОМЕР СЧЕТА НА ОПЛАТУ | НОМЕР ЛИЦЕВОГО СЧЕТА | ДАТА ВЫСТАВЛЕНИЯ СЧЕТА | СТАТУС |
|--------------|----------------------------------|-----------------------|----------------------|------------------------|------------------|
| 3 | a75b74cbcfe446509e8ee874f421bd65 | 1 | 10 | 01-01-2015 12:00 | Ожидает оплаты |
| 3 | a75b74cbcfe446509e8ee874f421bd65 | 2 | 20 | 01-02-2015 12:00 | Оплачен |
| 3 | a75b74cbcfe446509e8ee874f421bd65 | 3 | 30 | 01-03-2015 12:00 | Оплачен частично |
| 3 | a75b74cbcfe446509e8ee874f421bd65 | 4 | 40 | 01-04-2015 12:00 | Ожидает оплаты |
| 3 | a75b74cbcfe446509e8ee874f421bd65 | 5 | 50 | 01-05-2015 12:00 | Оплачен частично |

| | | | | | |
|---|----------------------------------|----|-----|---------------------|---------------------|
| 3 | a75b74cbcfe446509e8ee874f421bd65 | 6 | 60 | 01-06-2015 12:00 | Отменен |
| 2 | a75b74cbcfe446509e8ee874f421bd64 | 7 | 70 | 01-01-2015 12:00 | Ожидает оплаты |
| 2 | a75b74cbcfe446509e8ee874f421bd64 | 8 | 80 | 01-02-2015 12:00 | Оплачен |
| 2 | a75b74cbcfe446509e8ee874f421bd64 | 9 | 90 | 01-03-2015 12:00 | Оплачен частично |
| 2 | a75b74cbcfe446509e8ee874f421bd64 | 10 | 100 | 01-04-2015 12:00 | Ожидает оплаты |
| 2 | a75b74cbcfe446509e8ee874f421bd64 | 11 | 110 | 01-05-2015 12:00 | Оплачен частично |
| 2 | a75b74cbcfe446509e8ee874f421bd64 | 12 | 120 | 01-06-2015 12:00 | Отменен |

Тестовые платежи

| НОМЕР УСЛУГИ | API КЛЮЧ | НОМЕР ПЛАТЕЖА | НОМЕР ЛИЦЕВОГО СЧЕТА | ДАТА ПЛАТЕЖА |
|-----------------|----------------------------------|------------------|----------------------------|---------------------|
| 3 | a75b74cbcfe446509e8ee874f421bd65 | 1 | 10 | 01-01-2015 12:00 |
| 3 | a75b74cbcfe446509e8ee874f421bd65 | 2 | 20 | 01-02-2015 12:00 |
| 3 | a75b74cbcfe446509e8ee874f421bd65 | 3 | 30 | 01-03-2015 12:00 |
| 2 | a75b74cbcfe446509e8ee874f421bd64 | 4 | 40 | 01-04-2015 12:00 |
| 2 | a75b74cbcfe446509e8ee874f421bd64 | 5 | 50 | 01-05-2015 12:00 |
| 2 | a75b74cbcfe446509e8ee874f421bd64 | 6 | 60 | 01-06-2015 12:00 |

Тестовые счета на оплату банковской картой

| API КЛЮЧ | НОМЕР СЧЕТА НА ОПЛАТУ | СТАТУС СЧЕТА ПО БАНКОВСКОЙ КАРТЕ | ЗАПРЕТ СТОРНИРОВАНИЯ |
|----------------------------------|-----------------------------|--|-------------------------|
| a75b74cbcfe446509e8ee874f421bd64 | 100 | Счет зарегистрирован, но не оплачен | Да |

| | | | |
|----------------------------------|-----|---|-----|
| a75b74cbcfe446509e8ee874f421bd64 | 101 | Предавторизованная сумма захолдирована (для двухстадийных платежей) | Да |
| a75b74cbcfe446509e8ee874f421bd64 | 102 | Проведена полная авторизация суммы счета | Нет |
| a75b74cbcfe446509e8ee874f421bd64 | 103 | Авторизация отменена | Нет |
| a75b74cbcfe446509e8ee874f421bd64 | 104 | По транзакции была проведена операция возврата | Нет |
| a75b74cbcfe446509e8ee874f421bd64 | 105 | Инициирована авторизация через ACS банка-эмитента | Да |
| a75b74cbcfe446509e8ee874f421bd64 | 106 | Авторизация отклонена | Да |
| a75b74cbcfe446509e8ee874f421bd66 | 100 | Счет зарегистрирован, но не оплачен | Да |
| a75b74cbcfe446509e8ee874f421bd66 | 101 | Предавторизованная сумма захолдирована (для двухстадийных платежей) | Да |
| a75b74cbcfe446509e8ee874f421bd66 | 102 | Проведена полная авторизация суммы счета | Нет |
| a75b74cbcfe446509e8ee874f421bd66 | 103 | Авторизация отменена | Нет |
| a75b74cbcfe446509e8ee874f421bd66 | 104 | По транзакции была проведена операция возврата | Нет |
| a75b74cbcfe446509e8ee874f421bd66 | 105 | Инициирована авторизация через ACS банка-эмитента | Да |
| a75b74cbcfe446509e8ee874f421bd66 | 106 | Авторизация отклонена | Да |

СКАЧАТЬ ИСХОДНЫЕ КОДЫ ПРИМЕРОВ

- [Скачать пример исходных кодов на C#](#)
- [Скачать пример исходных кодов на PHP](#)