

# API «Директ Кредит»

## Скрипты, необходимые для интеграции сервиса

### Для production

```
<link rel="stylesheet" href="//dcapi.direct-credit.ru/style.css" type="text/css">
<script src="//dcapi.direct-credit.ru/JsHttpRequest.js" type="text/javascript"></script>
<script src="//dcapi.direct-credit.ru/dc.js" charset="utf-8" type="text/javascript"></script>
```

### Для тестирования

```
<link rel="stylesheet" href="//dcapi.direct-credit.ru/style.css" type="text/css">
<script src="http://dev-dc.direct-credit.ru/api/JsHttpRequest.js" type="text/javascript"></script>
<script src="http://dev-dc.direct-credit.ru/api/testapi.js" charset="utf-8" type="text/javascript"></script>
```

Данный код добавляется между тегами `<head> ... </head>` на страницах, где используется сервис кредитования.  
**Без них ничего работать не будет!**

Конструкция `'//'` перед адресом скриптов позволяет использовать их как по **http-протоколу**, так и по защищенному протоколу **https**. Данная конструкция позволяет автоматически подставлять либо **http**, либо **https**, в зависимости от страницы, на которой подключены скрипты.

Например, Вы добавили скрипты на странице со списком товаров `http://mysite.ru/catalog/`, которая доступна по протоколу **http**, скрипты будут подключены так же по **http**, а оформление заказа, предположим, у вас находится в разделе `https://mysite.ru/order/`, который доступен по протоколу **https**. Скрипты будут подключены по **https**.

Все функции из API «Директ Кредит» должны вызываться в элементе **body**.

Все параметры, передающиеся в функцию, должны быть строкового типа (**'string'**)

## Основные параметры API

Основные параметры можно прописать так же между тегами `<head> ... </head>`. Они используются во всех функциях сервиса.

```
<script type="text/javascript">
  var partnerID = "0000";
  var debug = true;
</script>
```

Свой **partnerID**, являющийся обязательной переменной, **необходимо** узнать у менеджера Директ Кредит перед тем, как приступить к интеграции сервиса.

## Расчет ежемесячных платежей для товаров

Для расчета ежемесячных платежей необходимы следующие данные по товарам:

- ✓ **ID товара** (Элемента) – параметр: **productID**
- ✓ **Стоимость товара** – параметр: **productPrice**
- ✓ **Количество товара** (можно не указывать. По умолчанию «1») – параметр: **productCount**
- ✓ **Категория/тип товара** – параметр: **productType**

Для получения расчета ежемесячных платежей необходимо собрать в массив данные по товарам:

```
var arrProducts = new Array();
arrProducts[0] = { id : productID[1], price: productPrice[1], count: productCount[1], type: productType[1] };
arrProducts[1] = { id : productID[2], price: productPrice[2], count: productCount[1], type: productType[2] };
arrProducts[2] = { id : productID[3], price: productPrice[3], count: productCount[1], type: productType[3] };
```

Предположим, что у нас есть три товара:

### 1. Ноутбук Lenovo ThinkPad X1 Carbon Touch

- 1.1 ID товара (**productID**): 2515
- 1.2 Цена (**productPrice**): 65 990 руб.
- 1.3 Количество (**productCount**): 1 шт.
- 1.4 Тип товара (**productType**): «Ноутбуки»

### 2. Мобильный телефон Apple iPhone 5 Black 32Gb

- 2.1 ID товара (**productID**): 743
- 2.2 Цена (**productPrice**): 24 680 руб.
- 2.3 Количество (**productCount**): 1 шт.
- 2.4 Тип товара (**productType**): «Мобильные телефоны»

### 3. Микроволновая печь Gorenje GM025DGE

- 3.1. ID товара (**productID**): 1306
- 3.2. Цена (**productPrice**): 5 910.15 руб.
- 3.3. Количество (**productCount**): 2 шт.
- 3.4. Тип товара (**productType**): «Бытовая техника»

Наш код будет выглядеть следующим образом:

```
var arrProducts = new Array();
arrProducts[0] = { id: '2515', price: '65990', count: '1', type: 'Ноутбуки' };
arrProducts[1] = { id: '743', price: '24680', count: '1', type: 'Мобильные телефоны' };
arrProducts[2] = { id: '1306', price: '5910.15', count: '2', type: 'Бытовая техника' };
```

\* Разделителем копеек в цене (параметр: **price**) может быть только точка, например, '1023.15'. Пробелов быть не должно!

Если какой-то товар покупается в количестве нескольких штук, необходимо в массиве использовать необязательное поле **count** для указания количества. Если его не указать, считается, что count: '1'

Перед началом работы, для успешной интеграции и корректного подбора кредитной программы банка, необходимо обязательно заполнить таблицу соответствия типов товаров, с передаваемым типом товара с сайта. Обратитесь к вашему менеджеру Директ Кредит для настройки соответствия таблиц.

После того, как мы сформировали массив данных по товарам, можем запросить расчет ежемесячных платежей по всем позициям.

```
DCLoans(partnerID, 'getPayment', { products : arrProducts }, function(result){
    //result будет содержать массив товаров с рассчитанными ежемесячными платежами на основании подключенных
у вас в системе «Директ Кредит» кредитных программ банков.
}, debug);
```

Последний параметр в функции – **debug**, может принимать значение **false** или **true**.

Необходим для вывода отладки в консоли браузеров **Google Chrome** и **Mozilla Firefox**.

**result** будет содержать ответ следующей структуры:

1. **result.status** (Статус ответа, может быть **true** или **false**)
2. **result.payments** (Массив ежемесячных платежей по каждому из товаров)
  - 2.1. **result.payments** [ID товара].id (ID товара, который передавался в массив товаров для расчета ежемесячных платежей)
  - 2.2. **result.payments** [ID товара].payment (Ежемесячный платеж по конкретному товару)
  - 2.3. **result.payments** [ID товара].downPayment (Первоначальный взнос по конкретному товару в процентах)
3. **result.allProducts** (Ежемесячный платеж для всех товаров в сумме)
4. **result.downPayment** (Минимально возможный первоначальный взнос для всех товаров в сумме в процентах)

В случае каких-либо ошибок, **result.status** будет содержать **false**, а в **result.error** будет содержать текст ошибки. Остальных параметров не будет!

# API для корзины интернет-магазина

В корзине интернет-магазина, на шаге выбора способа оплаты, предлагается сделать **новый способ оплаты**: «В кредит», при выборе которого необходимо производить поочередно следующие действия:

1. Очистить кредитную корзину, на случай, если ранее были добавлены товары, но после клиент изменил свой заказ.
2. Добавить товары, которые в данный момент находятся в корзине.
3. Перейти к оформлению заявок на кредиты в подключенных банках

Для начала необходимо сформировать массив товаров, такой же, как для расчёта ежемесячных платежей, указав при этом **дополнительным параметром** название товара (`productName`):

```
var arrProducts = new Array();
arrProducts[0] = {id: productID[1], price: productPrice[1], count: productCount[1], type: productType[1], name: productName[1]};
arrProducts[1] = {id: productID[2], price: productPrice[2], count: productCount[2], type: productType[2], name: productName[2]}; arrProducts[2] = {id: productID[3], price:
productPrice[3], count: productCount[3], type: productType[3], name: productName[3]};
```

Предположим, что у нас есть три товара:

## 1. Ноутбук Lenovo ThinkPad X1 Carbon Touch

- 1.1. ID товара (`productID`): 2515
- 1.2. Цена (`productPrice`): 65 990 руб.
- 1.3. Количество (`productCount`): 1 шт.
- 1.4. Тип товара (`productType`): «Ноутбуки»

## 2. Мобильный телефон Apple iPhone 5 Black 32Gb

- 2.1. ID товара (`productID`): 743
- 2.2. Цена (`productPrice`): 24 680 руб.
- 2.3. Количество (`productCount`): 1 шт.
- 2.4. Тип товара (`productType`): «Мобильные телефоны»

## 3. Микроволновая печь Gorenje GMO25DGE

- 3.1. ID товара (`productID`): 1306
- 3.2. Цена (`productPrice`): 5 910.15 руб.
- 3.3. Количество (`productCount`): 2 шт.
- 3.4. Тип товара (`productType`): «Бытовая техника»

Наш код будет выглядеть так:

```
var arrProducts = new Array();
arrProducts[0] = {id: '2515', price: '65990', count: '1', type: 'Ноутбуки', name: 'Lenovo ThinkPad X1 Carbon Touch' };
arrProducts[1] = {id: '743', price: '24680', count: '1', type: 'Мобильные телефоны', name: 'Apple iPhone 5 Black 32Gb' };
arrProducts[2] = {id: '1306', price: '5910.15', count: '2', type: 'Бытовая техника', name: 'Gorenje GMO25DGE' };
```

\*Разделителем копеек в цене (параметр: `price`) может быть только точка, например, '1023.15'. Пробелов быть не должно!

После того, как мы сформировали массив данных по товарам, можем сделать запрос на добавление товаров, предварительно очистив корзину:

```
DCLoans(partnerID, 'delProduct', false, function(result){
    if (result.status == true) {
        DCLoans(partnerID, 'addProduct', { products : arrProducts }, function(result){
            //result.status может быть либо true, либо false
        }, debug);
    }
}, debug);
```

1. Первая функция ('`delProduct`') очищает кредитную корзину от ранее добавленных товаров.
2. Вторая функция ('`addProduct`') добавляет собранные товары в кредитную корзину.

**result** в обеих функциях будет содержать ответ следующей структуры:

– **result.status** (Статус ответа, может быть **true** или **false**).

В случае каких-либо ошибок, **result.status** будет содержать **false**. А в **result.error** будет содержать текст ошибки.

Если окно для оформления кредитных заявок нужно вызвать тут же, то необходимо по результатам функции **addProducts** вызвать функцию **saveOrder**:

```
DCLoans(partnerID, 'delProduct', false, function(result){
  if (result.status == true) {
    DCLoans(partnerID, 'addProduct', { products : arrProducts }, function(result){
      if (result.status == true) {
        DCLoans(partnerID, 'saveOrder', { order : '123', phone: '9161234567' }, function(result){
          //result.status может быть либо false, либо всплывет окошко для оформление кредитных заявок
          }, debug);
        }
      }, debug);
    }
  }, debug);
}, debug);
```

Последняя функция **'saveOrder'** создает заказ в базе данных сервиса и вызывает окно кредитной анкеты.

В случае каких-либо ошибок **result** вернет следующие параметры:

**result.status: false**

**result.error:** будет содержать текст ошибки

Последний параметр в функции – **debug**, может принимать значение **false** или **true**.

Необходим для вывода отладки в консоли браузеров **Google Chrome** и **Mozilla Firefox**.

Если окно для оформления кредитных заявок нужно вызывать на следующем шаге:

```
DCLoans(partnerID, 'saveOrder', { order : '123', phone: '9161234567' }, function(result){
  //result.status может быть либо false, либо всплывет окошко для оформление кредитных заявок
}, debug);
```

В метод **saveOrder** передаются **обязательные** параметры **order, phone, codeTT**:

1. **order** – содержит номер заказа интернет-магазина
2. **phone** – данный параметр содержит номер телефона клиента, передаваемый в кредитную анкету из формы заказа интернет-магазина.
3. **codeTT** (Идентификатор торговой точки интернет-магазина. Используется в случае, если у интернет-магазина несколько точек выдачи товара, либо магазин работает в нескольких регионах. Можно указать любой удобный числовой идентификатор магазина, будь то код региона или внутренний номер магазина). Может принимать как числовые (77), так и буквенные (msk) значения.

**!** Передача номера мобильного телефона клиента из формы заказа магазина в кредитную анкету «Директ Кредит» должна осуществляться в определенном формате: **'9161234567' ! (\*)**

Это очень важный пункт в интеграции «Директ Кредит». Необходимо для того, чтобы заявки, оформление которых было прервано на первом шаге заполнения анкеты, не терялись. Сотрудники call-центра «Директ Кредит» прозвонят такие заявки и предложат помощь в заполнении анкеты по телефону, доведя тем самым вашего клиента до получения кредитного решения.

Так же метод **saveOrder** может содержать дополнительные параметры:

1. **loanTerm** (Срок кредитования в месяцах, по умолчанию в фильтре при оформлении заявок на кредиты)
2. **downPayment** – (Первоначальный взнос в %, по умолчанию в фильтре при оформлении заявок на кредиты)
3. **program** - ID Кредитного продукта в соответствии с кодами продуктов от банков
4. **firstName** - Имя клиента (Подставляется в анкету)
5. **email** - E-mail клиента (Подставляется в анкету)
6. **insurance** - Страхование

Возможные варианты:

- L - Страхование жизни заемщика
- W - Страхование заемщика от потери работы
- L/W - Оба вида страхования заемщика

## 7. **promo** - Признак промо-продукта.

Возможные варианты:

- 006
- 0010
- 0012
- 0024
- 0036

Список может быть расширен

### Пример использования параметров:

```
DCLoans(partnerID, 'saveOrder', { order: '123', phone: '9161234567', codeTT: '123123',  
loanTerm: '24', downPayment: '10',  
errorMsg: 'При попытке оформить заявку на получение кредита произошла ошибка. Попробуйте обновить страницу.' }, function(result){  
//result.status может быть либо false, либо появляется виджет для оформления кредитных заявок  
}, debug);
```

Дополнительные параметры не обязательны и используются на усмотрение партнера.

## Функция получения статуса по заявке

Для получения статуса оформления заявки вашего клиента существует следующая callback функция:  
**DCCheckStatus(result)**

Необходимо объявить функцию **DCCheckStatus** на своей стороне.

Если функция объявлена, то она будет вызвана после закрытия анкеты кредитования и в параметр **result** будет передано одно из нижеперечисленных значений:

result = 0 Пользователь вышел не оформив заявку до конца. Наш call-центр таких клиентов прозванивает

result = 1 Предварительное одобрение

result = 3 (В кредите отказано)

result = 4 (Отказ клиента)

result = 5 Отправлен на верификацию