Connection instructions

Requirements

You need to have an account in Capitalist.net to connect the payment system to your website. If you still do not have an account, register at https://capitalist.net/reg.

Shop making

Before filling out the form, be ready to upload the verification file to your website (see the second step).

In the main menu select "**Add Store**" or go to https://capitalist.net/merchants/add.

Step 1: Input of a name, domain, and secret key

- A store name can contain up to 60 symbols. It will be visible for users.

- The domain name of your site.

Step 2: Confirmation of the domain ownership

Follow the confirmation instructions on the page, upload the file to your website and click "**Confirm**".

Step 3: Detailed configuration of the store

- Fill in the description, select a category and upload a logo.

- In the settings section fill in Success URL, Fail URL, Status URL, and Interaction URL. (see picture2)

- Interaction URL will receive automatic notifications about paid orders.

- Save and wait for the moderator's approval.

Integration

So, you have registered and created a store, now you need to complete an important step - to prepare your site for interaction with the Capitalist API and set up the work.

Payment Request Form

This HTML form is the key one. With its help you can send a customer for a payment. It provides the basic parameters required for this purpose. As you can see from the example, you can pass the query parameter lang=en or ru to select the interface language.

The method="get" is also available, so you can pass these parameters in a query string.

Request Example

```
<form name="payment" method="post" action="https://capitalist.net/merchant/pay?lang=en"
accept-charset="UTF-8"><input type="hidden" name="merchantid" value="70" />

<input type="hidden" name="number" value="123981" />

<input type="hidden" name="amount" value="15.23" />

<input type="hidden" name="currency" value="USD" />

<input type="hidden" name="description" value="My description" />

<input type="hidden" name="sign" value="5ae3dbfdbac751f44355dedf5db59d02" />

</form>
```

Query parameters

Key	Name	Format	Example	Description
merchantid	Merchant ID	/^\d+$/	70	Store identifier.
number	Payment No.	/^[A-Z\d\-]{1,42}$/i	123981	Order number. It is saved in the Capitalist billing. It allows you to identify an order in the system, and also link it with orders in your billing. It checks originality for a specific **Merchant ID**. It may contain numbers, Latin letters, hyphen.
amount	Amount	/^[\d\.]+$/	15.23	Order sum. No more than two symbols after comma for USD, RUR, EUR, USDT. No more than 8 symbols after comma for BTC. Minimum: 20 RUR, 0.1 USD/EUR, 0.001 BTC.

				Minimum USDT (Trc20): 5 USD/USDT Minimum USDT (Erc20): 300 USD/USDT
currency	Currency	/^(USD|EUR|RUR|BTC|USDT|USDTt|USDT-TRC20|USDT-ERC20)$/	USD	Order currency. Depending on it, a corresponding account will be replenished that is set in the settings of the shop. To enable the receiving of bitcoins, please contact the service support. In this case, the order currency must be BTC. USDT (Erc20) currency codes: • **USDT** or **USDT-ERC20** USDT (Trc20) currency codes: • **USDTt** или **USDT-TRC20** Payment method "USDT (Erc20)" is available for orders with amount >= 300 USDT/USDTt/USD Payment method "USDT (Trc20)" is available for orders with amount >= 5 USDT/USDTt/USD
description	Description	/^.{1,150}$/	Hallo.	Payment description.
sign	Sign			Digital signature. See digital signature formation.
Optional parameters opt*				
opt_email	Preselected email			The user's e-mail is required in the second step.
opt_payway	Preselected payway			Payment method code. The code list is listed at the end of this document.
opt_pc	Preferred payment currency			The preferred currency in the second step (if it is available and if opt_payway is passed, the order sum will be displayed in this currency at the second step). Decorative parameter.

opt_lock_payway	Lock payway.			It prohibits a user from changing the payment method. Available value: • "1" - changing the payment method is forbidden • "0" (default) - you can change
opt_cb_method		get or post		A user's returning to success/fail/status url will take place with the passing of the parameters by the selected method. • "get" • "post" (default)
opt_client_id		Line ^.{0,64}$		Some identifier of your client

Generation of digital signature

A digital signature is formed by combining the values of all required form parameters. Parameters are concatenated alphabetically, i.e. for the form of sending a payment request in the following order (alphabetically): **amount**, **currency**, **description**, **merchantid**, **number, opt_email**. Concatenation goes through the ":" character. If optional parameters **opt_*** are not passed, they do not take part in a signature

The value obtained after combining the parameters is hashed by the **HMAC-MD5** method, where the "**secret key**" of the store is used as the secret key. (The description of HMAC on Wikipedia: https://ru.wikipedia.org/wiki/HMAC).

As a result, the **sign** parameter passes a hashcode in a lowercase hexadecimal encoding.

The example of EDS(electronic digital signature) formation in PHP:

See at the end of this document.

Order creation API

To create an order or view creation errors, you can make a request (GET or POST) to https://capitalist.net/merchant/payGate/createorder.

The request parameters are the same as in the "Payment request form" item.

Request example

The example shows a POST request with a JSON body (Content-Type: application-json). It is also acceptable to pass parameters in the GET query string.

```
### Order creation API
POST https://capitalist.net/merchant/payGate/createorder
Accept: application/json
Content-Type: application/json

{
  "merchantid": 163,
  "amount": 50,
  "number": "1657014969",
  "description": "test order creation",
  "currency": "USD",
  "sign": "mytestsign"
}

# Fail response sample
{
  "errors": {
    "sign": "Wrong signature."
  }
}

# Success response sample
{
  "order": {
    "description": "My demo payment.",
    "amount": 50,
    "currency": "USD",
    "number": "1677063939",
    "merchant": 163,
    "status": "NEW",
    "statusLocalized": "Waiting for payment",
    "date": 1677063945,
    "email": null,
    "paid": false,
    "refunded": false,
    "chargedBack": false,
    "paymentUrl": "https:\/\/capitalist.net\/merchant\/order\/163\/1677063939",
    "convertedAmount": null,
    "convertedCurrency": null
  }
}
```

Customer Return Pages

After payment the payment gateway redirects the customer to the return page. On this page Capitalist tries to determine, building on the available data, the payment status and, depending on it, redirect a customer to a particular result page on the seller's website (**Success URL**, **Fail URL**, **Status URL, Interaction URL**) respectively.

Parameters which are passed to result pages on the seller's website

When a user is redirected to the result pages, the following parameters are passed:

Key	Name	Example	Description
merchant_id	Merchant ID	70	Store identifier.
order_number	Payment No.	123981	Order number.
payment_state	Payment state	success	Payment status. (It can get values **success** or **fail**, **inprocess**).
payment_amount	Payment amount	15.23	Payment sum.
payment_currency	Payment currency	USD	Payment currency. (USD, EUR, RUR)
sign			Digital signature.

When a digital signature is created, the parameters are concatenated alphabetically, i.e. in the following order: **merchant_id**, **order_number**, **payment_amount**, **payment_currency**, **payment_state**.

Payment notification (Interaction URL)

During making a payment, Capitalist creates a request with some data on it to the interaction page (**Interaction URL**.) This request contains all the necessary data to complete the payment process on your site.

Verification of payment information

To exclude the possibility of compromising the payment notification on the interaction page, you need to implement its verification, which is given below.

Verification of a data source

In this review, you need to make sure that the payment notification is sent directly by Capitalist and is not compromised.

Data verification

Always check the parameters in the payment notification:

- Store identifier (parameter **merchant_id**)

- Order number (parameter **order_number**)

- Payment amount (parameter **payment_amount**)

- Payment currency (parameter **payment_currency**)

- Payment status (parameter **payment_state**). The value **success** must be there.

- Digital signature (parameter **sign**). See Formation of digital signatureы

Validation of data with the help of API request

To check the status of an order payment, you can run a request (GET or POST) at https://capitalist.net/merchant/payGate/checkstate.

Query parameters

Parameter	Name	Example	Description
merchant_id	Merchant ID	70	Store identifier.
order_number	Payment No.	123981	Order number.
amount	Amount	2.32	Sum of payment.
currency	Currency	USD	Order currency.
sign			Digital signature.

When a digital signature is created, the parameters are concatenated alphabetically, i.e. in the following order: **amount, currency, merchant_id, order_number**.

The supplied parameters must comply with an existing order and the signature must be correct, in this case the order status will be returned

Query Result

The result is returned in JSON format (https://ru.wikipedia.org/wiki/JSON). Here:

- **success != false** – the order status could not be checked.

- **data.order.paid = true** – an order is paid.

- **convertedAmount** and **convertedCurrency** are filled in case of auto-conversion to another currency works during replenishment. There will be the total amount credited to the merchant's account.

The example of a successful response and an order is paid

```json
{
  "success": true,
  "data": {
    "order": {
      "merchant_id": 70,
      "order_number": "1446466734",
      "description": "My purpose of payment.",
      "state": "paid",
      "paid": true,
      "payUrl": https://capitalist.net/merchant/order/70/1446466734,
      "convertedAmount": 314.06,
      "convertedCurrency": "RUR"
    }
  }
}
```

The example of a response if signature does not coincide

```json
{
  "success": false,
  "error": "The attempt to check the signature  was failed.",
  "data": {
    "errors": {
      "sign": [
        "The attempt to check the signature  was failed."
      ]
    }
  }
}
```

The example of an answer if the order cannot be found or its data does not coincide

```json
{
  "success": false,
  "error": "The attempt to check the order  was failed.",
  "data": {
    "errors": {
      "sign": [
        "The attempt to check the order  was failed."
      ]
    }
  }
}
```

Calculation of an approximate sum of the order for displaying it in a currency different from the currency of placing the order

To get an approximate sum of an order, you can run a request (GET or POST) at https://capitalist.net/merchant/payGate/paywayinfo. In processing of the response, it is necessary to check that data.order != null. (See also the parameter **opt_pc** in the payment request form.)

Query parameters

Parameter	Name	Example	Description
merchant_id	Merchant ID	70	Store identifier.
amount	Amount	71.94	Sum of payment
currency	Currency	EUR	Order currency.
opt_payway	Selected payway	mastercard	Payment method code. The code list is listed at the end of this document.
sign			Digital signature.

Request Example

https://capitalist.net/merchant/payGate/paywayinfo?merchant_id=70&amount=71.94&currency=EUR&sign=b2053cea0a768666e7aa887e4dba9b44&opt_payway=mastercard

The example of an answer

```
{
  "success": true,
  "data": {
    "target": {
      "amount": "71.94",
      "currency": "EUR"
    },
    "availablePaywayCurrencies": [
      "RUR",
      "EUR",
      "UAH",
      "USD"
    ],
    "order": {
      "merchantid": "70",
      "opt_payway": "mastercard",
      "amount": "5197.98",
      "currency": "RUR"
    }
  }
}
```

Parameters of the response

Parameter	Description
availablePaywayCurrencies	The list of available currencies in the 2nd step of payment.
order { }	In case of impossibility of calculation or unavailability of currency, data.order can get the meaning of null.
order.amount order.currency	The sum and currency for invoicing.

The example of signature (or signature verification) in PHP

```
/**

 * Method for signing data
 *
 * @param array $data is a hash array for signing (e.g. $_POST)
 * @param string $secret is the secret store key
 *
 * @return string
 */
function signData($data, $secret)
{
    unset($data['sign']);
    ksort($data, SORT_STRING);
    $str = implode(':', $data);
    return hash_hmac('md5', $str, $secret);
}
```

```php
$order = [

    'merchantid' => 85123,
    'number' => 'testorder123',
    'amount' => '12.34',
    'currency' => 'RUR',
    'description' => 'My demo payment.',
];

$order['sign'] = signData($order, 'mysecret123');

// var_dump($order):
//
// array (size=6)
//   'merchantid' => int 85123
//   'number' => string 'testorder123' (length=12)
//   'amount' => string '12.34' (length=5)
//   'currency' => string 'RUR' (length=3)
//   'description' => string 'My demo payment.' (length=16)
//   'sign' => string '8bb5946e96999f81f8958a8c79d50d4b' (length=32)
```

Order cancellation

The placed order can be canceled at any time using the handle https://capitalist.net/merchant/payGate/cancel.

If the cancellation is successful, the response code will be **200**, otherwise **422**. The response body returns json with order details.

You can only cancel an unpaid order.

Request parameters

Parameter	Description
merchantId	Merchant/shop identifier
orderNumber	Order number
signature	Signature As in other places before, all parameters (their values) of the request, with the exception of signature, are concatenated through a colon into one string. Order - in alphabetical order of parameter names; The value obtained after combining the parameters is hashed using the HMAC-MD5 method, where the "secret key" of the store is used as the secret key. That is Hash_HMAC-MD5("<merchantId>:<orderNumber>", <merchantSecret>) We pass the final string as signature

GET request with query parameters is allowed as well as **POST** request (**application/x-www-form-urlencoded**, **application/json**).

Request example

```
POST https://capitalist.net/merchant/payGate/cancel
Accept: application/json
Content-Type: application/json

{
    "merchantId": 163,
    "orderNumber": 1665130908,
    "signature": " b2053cea0a768666e7aa887e4dba9b44"
}
```

Response example

```
{
  "order": {
    "description": "My demo payment.",
    "amount": 20,
    "currency": "RUR",
    "number": "1665130908",
```

```
    "merchant": 163,
    "status": "CANCELLED",
    "statusLocalized": "CANCELLED",
    "date": 1665130909,
    "created": true,
    "email": null,
    "paid": false,
    "paymentUrl": "https:\/\/capitalist.net\/merchant\/order\/163\/1665130908"
  }
}
```

Recurrent payment for subscription

A recurring payment for an active subscription is initiated by the merchant using the following handle: https://capitalist.net/merchant/subscription/payment.

If canceled successfully, the response will be 200, otherwise 422.

Request parameters

Parameter	Example	Description
merchantId	70	Merchant ID
subscriptionId	a11163f0-463c-11ed-8b5d-7bc897aeac86	Subscription ID
timestamp	16652007860000	
signature		Signature. As in other places before, all parameters (their values) of the request, with the exception of signature, are concatenated through a colon into one string. Order - in alphabetical order of parameter names; The value obtained after combining the parameters is hashed using the HMAC-MD5 method, where the "secret key" of the store is used as the secret key. That is Hash_HMAC-MD5("<merchantId>:<subscriptionId>:<timestamp>", <merchantSecret>) We pass the final string as signature

GET request with query parameters is allowed as well as **POST** request (**application/x-www-form-urlencoded**, **application/json**).

Request example

```
POST https://capitalist.net/merchant/subscription/cancel
Accept: application/json
Content-Type: application/json

{
  "merchantId": 163,
  "subscriptionId": "a11163f0-463c-11ed-8b5d-7bc897aeac86",
  "timestamp": "16652007860000",
  "signature": "fa69d5568b8b9e2836000fa9d70abef2"
}
```

Response example

```
{
  "success": true
}
```

Subscription cancellation

An active subscription can be canceled at any time using the handle https://capitalist.net/merchant/subscription/cancel.

If canceled successfully, the response will be 200, otherwise 422.

Request parameters

Parameter	Example	Description
merchantId	70	Merchant ID
subscriptionId	a11163f0-463c-11ed-8b5d-7bc897aeac86	Subscription ID
timestamp	16652007860000	
signature		Signature. As in other places before, all parameters (their values) of the request, with the exception of signature, are concatenated through a colon into one string. Order - in alphabetical order of parameter names; The value obtained after combining the parameters is hashed using the HMAC-MD5 method, where the "secret key" of the store is used as the secret key. That is Hash_HMAC-MD5(“<merchantId>:<subscriptionId>:<timestamp>”, <merchantSecret>) We pass the final string as signature

GET request with query parameters is allowed as well as **POST** request (**application/x-www-form-urlencoded**, **application/json**).

Request example

```
POST https://capitalist.net/merchant/subscription/cancel
Accept: application/json
Content-Type: application/json

{
  "merchantId": 163,
  "subscriptionId": "a11163f0-463c-11ed-8b5d-7bc897aeac86",
  "timestamp": "16652007860000",
  "signature": "fa69d5568b8b9e2836000fa9d70abef2"
}
```

Response example

```
{
  "success": true
}
```

Subscriptions list

The list of active subscriptions in page mode can be obtained using the handle https://capitalist.net/merchant/subscription/list.

Returns the total number and a list of subscriptions of a page: {total: [int], subscriptions: {obj}}.

Request parameters

Parameter	Example	Description
merchantId	70	Merchant ID.
limit	1	Page size
offset	0	Page offset
timestamp	16652007860000	
signature		Signature. As in other places before, all parameters (their values) of the request, with the exception of signature, are concatenated through a colon into one string. Order - in alphabetical order of parameter names; The value obtained after combining the parameters is hashed using the HMAC-MD5 method, where the "secret key" of the store is used as the secret key. That is Hash_HMAC-MD5("<limit>:<merchantId>:<offset>:<subscriptionId>:<timestamp>", <merchantSecret>) We pass the final string as signature

GET request with query parameters is allowed as well as **POST** request (**application/x-www-form-urlencoded**, **application/json**).

Request example
```
POST http://localhost:8050/merchant/subscription/list
Accept: application/json
Content-Type: application/json

{
  "merchantId": 374,
  "timestamp": "16652007860000",
  "limit": 1,
  "offset": 0,
  "signature": "fa69d5568b8b9e2836000fa9d70abef2",
}
```

Response example

```json
{
  "success": true,
  "subscriptions": [
    {
      "id": "dda214c4-7a1f-11ed-8b5d-c76b09b23a0a",
      "subscriptionId": "dda214c4-7a1f-11ed-8b5d-c76b09b23a0a",
      "productId": "d19986cc-4562-11ed-8b5d-2f5a48c6e4f5",
      "product": {
        "id": "d19986cc-4562-11ed-8b5d-2f5a48c6e4f5",
        "name": "Подпииисочкаааа",
        "shortDescription": "Моя хорошая подписулечка",
        "merchantId": 374,
        "period": "month",
        "price": "10",
        "currency": "USD"
      },
      "status": "new",
      "nextUpdate": 1673470800,
      "nextUpdateFormatted": "2023-01-11 21:00:00",
      "orders": [
        "10101792"
      ],
      "email": "myclient@example.com",
      "clientId": "myclient@example.com",
      "merchantId": 374,
      "lastPayments": [],
      "lastPaymentDate": null
    }
  ],
  "total": 3
}
```

The list of payment systems codes

In the optional **opt_payway** parameter (see above) you can pass the code of the selected payment system). A user will be redirected immediately to the next payment step according to the chosen system.

The list of all codes (in JSON)

https://capitalist.net/merchant/payGate/payways?all=1

The list of codes filtered for a particular store, sum and currency

https://capitalist.net/merchant/payGate/payways?merchantid=70&amount=2.32&currency=USD

The example of an answer

```
{

  "success": true,
  "data": {
    "payways": [
      {
        "code": "capitalist",
        "name": "Capitalist.net"
      },
      {
        "code": "webmoney",
        "name": "WebMoney"
      },
      {
        "code": "yandexmoney",
        "name": «Yandex.Money»
      },
      {
        "code": "epese",
        "name": "EPESE"
      }
    ]
  }
}
```

Currency rates request

Some payment methods requires special order currency selected. For example: Bitcoin payment only available for orders in BTC currency. In order to calculate equivalent amount in BTC you should fetch currency rate via special request to **/merchant/payGate/currencyrate**.

Currency pair rate (response: JSON)

GET https://capitalist.net/merchant/payGate/currencyrate

Request options

Параметр	Имя	Пример	Описание
merchantid	Merchant ID	70	Your merchant ID.
amount	Amount	71.94	Amount.
currency	Currency	USD	Currency 1.
opt_pc	Destination currency	BTC	Currency 2.

No signature required.

Response example

```
{
    "data": {
        "sell": [
            {
                "comment": "sell USD for BTC to capitalist",
                "pair": "USD/BTC",
                "sellAmount": "500",
                "sellCurrency": "USD",
                "amount": 0.051175,
                "currency": "BTC"
            }
        ],
        "buy": [
            {
                "comment": "buy USD for BTC from capitalist",
                "pair": "USD/BTC",
                "buyAmount": "500",
                "buyCurrency": "USD",
                "amount": 0.0544,
                "currency": "BTC"
            }
        ]
    }
}
```

History of last changes

1.12 by 16.06.2023

- Removed old registration screenshots

- Removed old history of changes

1.11 by 22.02.2023

- Added order creation API (payGate/createorder)

1.10 by 12.12.2022

- Subscriptions API: subscriptions list added

1.9 by 10.10.2022

- Subscriptions API added (Cancellation and Recurrent payments)

1.8 by 07.10.2022

- Order cancellation added to API.

1.7.6 by 17.05.2022

- USDT Erc20 and Trc20 limits added:

    o payment method USDT Erc20 available for orders with amount >= 300 USD/USDT/USDTt

    o payment method USDT Trc20 available for orders with amount >= 5 USD/USDT/USDTt

…