

# API Application Manual

Topic: API Application Manual For Vending

---

Summary API Application Manual

By reading this document, you can quickly use the API, generate Vending Token, Clear Credit Token and so on.

# Catalogue

<b>1</b>	<b>FOREWORD .....</b>	<b>3</b>
1.1	PREPARE .....	3
1.2	WEB SITE: <b>TEST PLATFORM</b> .....	3
<b>2</b>	<b>VENDING API.....</b>	<b>3</b>
2.1	SEND A CLEAR CREDIT REQUEST TO THE SERVER .....	3
2.2	SEND A DIRECTLY CLEAR CREDIT REQUEST TO THE SERVER .....	4
2.3	SEND A CLEAR TAMPER REQUEST TO THE SERVER .....	5
2.4	SEND A DIRECTLY CLEAR TAMPER REQUEST TO THE SERVER .....	7
2.5	SEND A QUERY CUSTOMER INFORMATION RECORD REQUEST TO THE SERVER .....	8
2.6	SEND A QUERY METER ID INFORMATION RECORD REQUEST TO THE SERVER .....	9
2.7	SEND A QUERY METER CREDIT RECORD REQUEST TO THE SERVER .....	9
2.8	SEND A VENDING REQUEST TO THE SERVER .....	11
2.9	SEND A VENDING DIRECTLY REQUEST TO THE SERVER .....	13

## 1 FOREWORD

### 1.1 Prepare

- 1) Before calling the api, you can input basic data, such as meter ID number, price, customer information, and account opening information, into the system through our Vending management system, so that the API can be executed correctly.
- 2) The account of the Vending management system can be obtained by asking our sales staff.

### 1.2 Web Site: [Test platform](#)

Follow below several steps to get token easily by using the API, More detailed API description , Please refer to the website

<http://www.server-newv.stronpower.com/swagger>

## 2 Vending API

### 2.1 Send a Clear Credit request to the server

Request URL as follows:

<http://www.server-newv.stronpower.com/api/ClearCredit>

If we want Clear Credit. And the Post Body as follows:

```
{
  "CompanyName": "your company name",
  "UserName": "your user name",
  "PassWord": "your password",
  "CustomerID": "your customer's id , This number is your own definition",
  "METER_ID": " your customer's Meter id"
}
```

If I want to Clear Credit of meter 58100000007, and Customer Id is 0001. The vending/API account info: Company Name: stron; User Name: 0000; Pass Word: stron888.

Post Body as follows:

```
{
  "CompanyName": "stron",
  "UserName": "0000",
  "PassWord": "stron888",
  "CustomerId": "0001",
  "METER_ID": "58100000007"
}
```

Operation instance:

## ClearCredit

**POST** /api/ClearCredit

Response Class (Status 200)  
string

Response Content Type

**Parameters**

Parameter	Value	Description
clearcredit	<pre>{   "CompanyName": "stron",   "UserName": "0000",   "PassWord": "stron888",   "CustomerId": "0001",   "METER_ID": "58100000007" }</pre>	

Parameter content type:

2 **Try it out!** [Hide Response](#)

3 

```
"4515 5704 0965 8793 2417,2021-03-30 12:22:30"
```

### 2.2 Send a directly Clear Credit request to the server

Request URL as follows:

<http://www.server-newv.stronpower.com/api/ClearCreditDirectly>

If you want generate the Clear Credit Token, and don't want to register Price, District, Sales Satations and Customer information into Server. So you can use this kind of API ClearCreditDirectly.

If we want get Clear Credit Token directly. And the Post Body as follows:

```
{
  "CompanyName": "your company name",
  "UserName": "your user name",
  "PassWord": "your password",
  "CustomerId": "your customer's id , This number is your own definition",
  "Meter_ID": "58000000008"
}
```

Operation instance:

## ClearCreditDirectly

**POST** /api/ClearCreditDirectly

**Response Class (Status 200)**  
string

Response Content Type

**Parameters**

Parameter	Value	Description
clearcredit	<pre>{   "CompanyName": " ",   "UserName": " ",   "PassWord": " ",   "Meter_ID": "580000000008" }</pre>	

Parameter content type:

[Try it out!](#) [Hide Response](#)

**Response Body**

```
"5862 1514 4311 2927 6498,2022/9/7 21:09:28"
```

**Response Code**

```
200
```

### 2.3 Send a Clear Tamper request to the server

Request URL as follows:

<http://www.server-newv.stronpower.com/api/ClearTamper>

If we want Clear Tamper. And the Post Body as follows:

```
{
  "CompanyName": "your company name",
  "UserName": "your user name",
```

```
"PassWord": "your password",  
"CustomerID": "your customer's id , This number is your own definition",  
"METER_ID": " your customer's Meter id"  
}
```

If I want to Clear Tamper of meter 58100000007, and Customer Id is 0001. The vending/API account info: Company Name: stron; User Name: 0000; Pass Word: stron888.

Post Body as follows:

```
{  
  "CompanyName": "stron",  
  "UserName": "0000",  
  "PassWord": "stron888",  
  "CustomerId": "0001",  
  "METER_ID": "58100000007"  
}
```

Operation instance:

**POST** /api/ClearTamper

**Response Class (Status 200)**  
string

Response Content Type

**Parameters**

Parameter	Value	Description
cleartamper	<pre>"CompanyName": "stron", "UserName": "0000", "PassWord": "stron888", "CustomerId": "0001", "METER_ID": "58100000007" }</pre>	

Parameter content type:

**1** → **Try it out!** [Hide Response](#)

**2** → **Response Body**

**3** → **"4211 4805 4646 7549 2227,2021-03-30 12:28:28"**

## 2.4 Send a directly Clear Tamper request to the server

Request URL as follows:

<http://www.server-newv.stronpower.com/api/ClearTamperDirectly>

If you want generate the Clear Tamper Token, and don't want to register Price, District, Sales Satations and Customer information into Server. So you can use this kind of API ClearTamperDirectly.

If we want get Clear Tamper Token directly. And the Post Body as follows:

```
{
  "CompanyName": "Your company name",
  "UserName": "Your name",
  "PassWord": "Your password",
  "Meter_ID": "580000000008"
}
```

Operation instance:

### ClearTamperDirectly

**POST** /api/ClearTamperDirectly

**Response Class (Status 200)**  
string

Response Content Type

**Parameters**

Parameter	Value	Description
cleartamper	<pre>{   "CompanyName": " ",   "UserName": " ",   "PassWord": " ",   "Meter_ID": "580000000008" }</pre>	

Parameter content type:

[Hide Response](#)

**Response Body**

**Token**

`"0771 7929 3203 2169 5532.2022/9/7 21:38:41"`

## 2.5 Send a Query Customer Information record request to the server

Request URL as follows:

<http://www.server-newv.stronpower.com/api/QueryCustomerInfo>

If I want to query credit record of Customer ID: CTS-00001

Post Body as follows:

```
{
  "CompanyName": "Your company name",
  "UserName": "Your name",
  "PassWord": "Your password",
  "CustomerId": "CTS-00001"
}
```

Operation instance:

The screenshot displays an API testing interface. The top section, titled 'Parameters', contains a table with columns 'Parameter', 'Value', and 'Description'. A single parameter named 'Querycustomer' is listed, with its value being a JSON object: 

```
{
  "CompanyName": "Your company name",
  "UserName": "Your name",
  "PassWord": "Your password",
  "CustomerId": "CTS-00001"
}
```

. A red arrow points to the 'CustomerId' field in this JSON. Below the table, a dropdown menu for 'Parameter content type' is set to 'application/json', with another red arrow pointing to it. A 'Try it out!' button is visible. The bottom section, titled 'Response Body', shows the JSON response from the server: 

```
[
  {
    "Company_name": "Your company name",
    "Username": "Your name",
    "Customer_id": "CTS-00001",
    "Customer_name": "Test client 1",
    "Customer_address": "plot 328 Upanga",
    "Customer_phone": "+25512345678",
    "Meter_id": "58101195194",
    "Meter_type": "Energy Meter",
    "Total_paid": "186.00/440.00",
    "Total_unit": "18.6/22",
    "Total_times": "4",
    "Unit": "kWh",
    "Price": "10.00",
    "Price_unit": "Tshs.",
    "Rate": "0.00"
  }
]
```

Parameter	Value	Description
Querycustomer	<pre>{   "CompanyName": "Your company name",   "UserName": "Your name",   "PassWord": "Your password",   "CustomerId": "CTS-00001" }</pre>	

Parameter content type:

[Try it out!](#) [Hide Response](#)

**Response Body**

```
[
  {
    "Company_name": "Your company name",
    "Username": "Your name",
    "Customer_id": "CTS-00001",
    "Customer_name": "Test client 1",
    "Customer_address": "plot 328 Upanga",
    "Customer_phone": "+25512345678",
    "Meter_id": "58101195194",
    "Meter_type": "Energy Meter",
    "Total_paid": "186.00/440.00",
    "Total_unit": "18.6/22",
    "Total_times": "4",
    "Unit": "kWh",
    "Price": "10.00",
    "Price_unit": "Tshs.",
    "Rate": "0.00"
  }
]
```



## 2.6 Send a Query Meter ID Information record request to the server

Request URL as follows:

<http://www.server-newv.stronpower.com/api/QueryMeterInfo>

If I want to query credit record of Meter ID: 58101195178

Post Body as follows:

```
{
  "CompanyName": "Your company name",
  "UserName": "Your name",
  "PassWord": "Your password",
  "MeterId": "58101195178"
}
```

Operation instance:

The screenshot displays an API testing tool interface. At the top, the 'Parameters' section is active, showing a table with columns: Parameter, Value, Description, Parameter Type, and Data Type. A row for 'Querymeter' is shown with a JSON body in the 'Value' column. A red arrow labeled '1' points to the 'MeterId' field in the JSON. Below the table, the 'Parameter content type' is set to 'application/json'. A red arrow labeled '2' points to the 'Try it out!' button. To the right, an 'Example Value' box shows a JSON schema for the response. Below the 'Parameters' section, the 'Response Body' section is active, displaying a large JSON object with various fields including 'Company\_name', 'Username', 'Customer\_id', 'Customer\_name', 'Customer\_address', 'Customer\_phone', 'Meter\_id' (highlighted with a red box), 'Meter\_type', 'Total\_paid', 'Total\_unit', 'Total\_times', 'Unit', 'Price', 'Price\_unit', and 'Rate'.

Parameter	Value	Description	Parameter Type	Data Type
Querymeter	<pre>{   "CompanyName": "Your company name",   "UserName": "Your name",   "PassWord": "Your password",   "MeterId": "58101195178" }</pre>		body	Model Example Value

Parameter content type: application/json

Try it out! Hide Response

**Response Body**

```
[
  {
    "Company_name": "Your company name",
    "Username": "Your name",
    "Customer_id": "CTS-00001",
    "Customer_name": "AllenTest",
    "Customer_address": "Hunan",
    "Customer_phone": "188888888888888888",
    "Meter_id": "58101195178",
    "Meter_type": "Energy Meter",
    "Total_paid": "0",
    "Total_unit": "0",
    "Total_times": "0",
    "Unit": "kwh",
    "Price": "1.00",
    "Price_unit": "usd",
    "Rate": "0.00"
  }
]
```

## 2.7 Send a Query Meter credit record request to the server

Request URL as follows:

<http://www.server-newv.stronpower.com/api/QueryMeterCredit>

If I want to query credit record of Meter ID Number: 580000000024

Post Body as follows:

```
{
  "CompanyName": "Your company name",
  "UserName": "Your name",
  "PassWord": "Your password",
  "MeterId": "580000000024"
}
```

Operation instance:

## QueryMeterCredit

**POST** /api/QueryMeterCredit

**Response Class (Status 200)**  
OK

Model	Example Value
	<pre>{   "METER_ID": "string",   "COMPANY": "string",   "NET_VALUE": "string",   "VAT": "string",   "AMOUNT": "string",   "TOKEN": "string",   "GEN_EMP_ID": "string",   "GEN_TIME": "2021-09-13T05:12:16.722Z" }</pre>

Response Content Type

**Parameters**

Parameter	Value	Description
creditrecord	<div><div>1</div><div><pre>{   "CompanyName": "tron",   "UserName": "tron",   "PassWord": "tron",   "MeterId": "580000000024" }</pre></div></div>	

Parameter content type:

Try it out!

Hide Response

2

## Response Body

```
[
  {
    "CUST_ID": "ST0001",
    "METER_ID": "58000000024",
    "COMPANY": "stron",
    "NET_VALUE": "443.46",
    "VAT": "56.54",
    "AMOUNT": "500 Mts/4.0 m³",
    "TOKEN": "2446 9742 6662 1300 2591",
    "GEN_EMP_ID": "0000",
    "GEN_TIME": "2021.09.12"
  },
  {
    "CUST_ID": "ST0001",
    "METER_ID": "58000000024",
    "COMPANY": "stron",
    "NET_VALUE": "418.00",
    "VAT": "53.30",
    "AMOUNT": "471.30 Mts/3.8 m³",
    "TOKEN": "3050 3739 5826 4703 7985",
  }
]
```

## Response Code

200

## 2.8 Send a Vending request to the server

Request URL as follows:

<http://www.server-newv.stronpower.com/api/VendingMeter>

Parameter	Value	Explanation
is_vend_by_unit	true	Vending by unit
	false	Vending by money

the post Body as follows:

```
{
  "CompanyName": " your company name ",
  "UserName": " your user name ",
  "PassWord": " your password ",
  "MeterID": "string",
  "is_vend_by_unit": "true: vending unit . false: vending money ",
  "Amount": 0
}
```

For example , If I want to top up meter ID: 58100000007, the recharge by **unit**, and the amount is 1700. the post Body as follows:

```
{
  "CompanyName": " your company name ",
  "UserName": " your user name ",
  "PassWord": " your password ",
  "MeterID": "58100000007",
  "is_vend_by_unit": "true ",
  "Amount": "1700"
}
```

**Notes:** if you want vending by unit, the value of “is\_vend\_by\_unit” is true.

The screenshot shows an API testing interface. A red arrow labeled '1' points to the request body field, which contains a JSON object: `{ "CompanyName": " stron", "UserName": "0000", "PassWord": "stron888", "MeterID": "58100000007", "is_vend_by_unit": "true", "Amount": 1700 }`. Below this, the 'Parameter content type' is set to 'application/json'. A red arrow labeled '2' points to the 'Try it out!' button. Below the button, the 'Response Body' is displayed, with a red arrow labeled '3' pointing to it. The response body is a JSON object: `[ { "Company_name": " stron", "Username": "0000", "Customer_id": "0001", "Customer_name": "Allen", "Customer_address": "China-ShenZhen", "Customer_phone": "15888888888", "Meter_id": "58100000007", "Meter_type": "Energy Meter", "Total_paid": "1295896.2162162162162162162162", "Total_unit": "1700", "Unit": "kWh", "Price": "705.12", "Price_unit": "NGN", "Rate": "7.50%", "Token": "6265 0934 1918 3689 8513", "Gen_time": "2021.03.31" }`.

1

2

3

```

vending
{
  "CompanyName": " stron",
  "UserName": "0000",
  "PassWord": "stron888",
  "MeterID": "58100000007",
  "is_vend_by_unit": "true",
  "Amount": 1700
}

Parameter content type:
application/json

Try it out! Hide Response

Response Body

[
  {
    "Company_name": " stron",
    "Username": "0000",
    "Customer_id": "0001",
    "Customer_name": "Allen",
    "Customer_address": "China-ShenZhen",
    "Customer_phone": "15888888888",
    "Meter_id": "58100000007",
    "Meter_type": "Energy Meter",
    "Total_paid": "1295896.2162162162162162162162",
    "Total_unit": "1700",
    "Unit": "kWh",
    "Price": "705.12",
    "Price_unit": "NGN",
    "Rate": "7.50%",
    "Token": "6265 0934 1918 3689 8513",
    "Gen_time": "2021.03.31"
  }
]
```

## 2.9 Send a Vending directly request to the server

Request URL as follows:

<http://www.server-api.stronpower.com/api/VendingMeterDirectly>

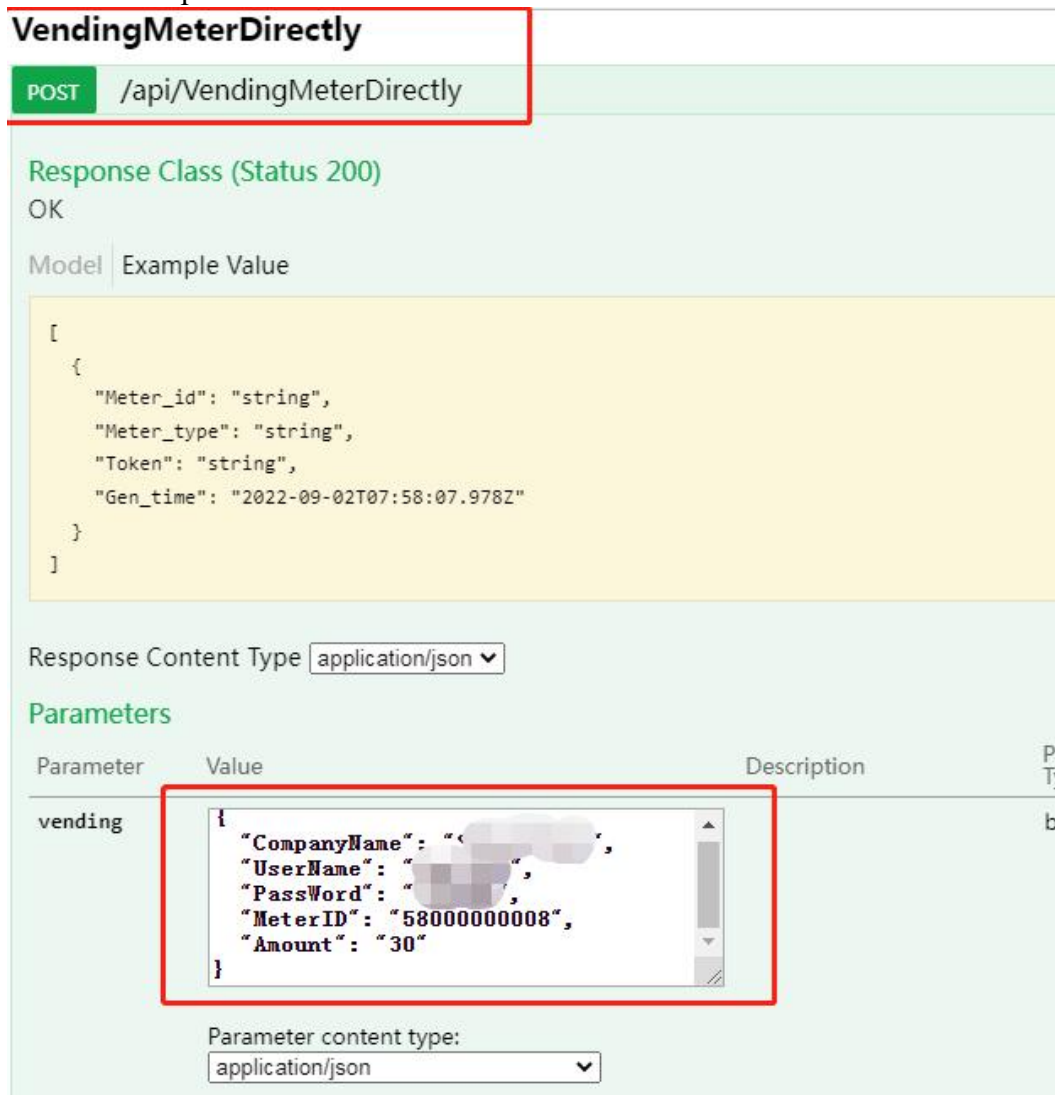
If you want generate the Token, and don't want to register Price, District, Sales Stations and Customer information into Server. So you can use this kind of API VendingMeterDirectly.

If we want generate the Token for Meter 580000000024, Meter type is Energy Meter, and recharge 30kWh.

Post Body as follows:

```
{
  "CompanyName": " your company name ",
  "UserName": " your user name ",
  "PassWord": " your password ",
  "MeterId": "580000000024",
  "Amount": "30"
}
```

Operation instance:



**VendingMeterDirectly**

**POST** /api/VendingMeterDirectly

**Response Class (Status 200)**  
OK

Model | Example Value

```
[
  {
    "Meter_id": "string",
    "Meter_type": "string",
    "Token": "string",
    "Gen_time": "2022-09-02T07:58:07.978Z"
  }
]
```

Response Content Type


**Parameters**

Parameter	Value	Description	Pa Ty
vending	<pre>{   "CompanyName": " ",   "UserName": " ",   "PassWord": " ",   "MeterID": "580000000008",   "Amount": "30" }</pre>		bo

Parameter content type:

### Response Body

```
[
  {
    "Meter_id": "580000000008",
    "Meter_type": "Energy Meter",
    "Token": "6976 8690 5101 8990 8139",
    "Gen_time": "2022.09.02"
  }
]
```



### Response Code

200