

# If you thought your energy bill rise was bad, wait until you see these ...

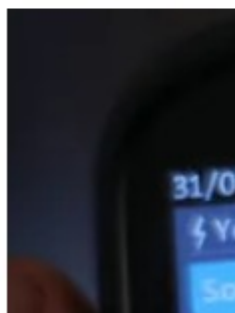
As the new tax even stamps at



## Energy bills forecast to remain above £2,000 in blow to Sunak's loan scheme

**WUOLIFE Energy Solutions**

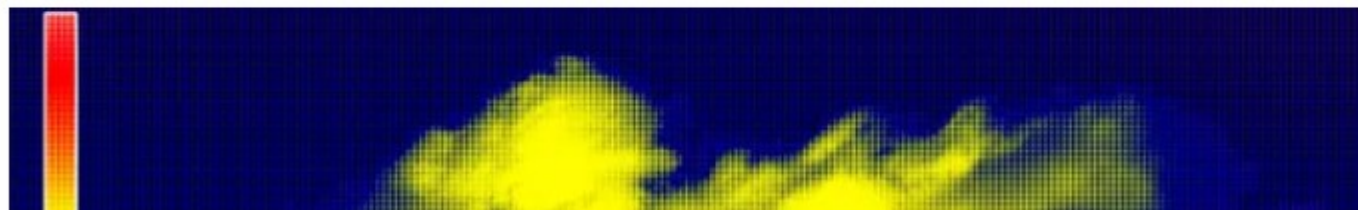
Analysts warn pressure to st



## Where Britain's journey to insulation went wrong

Conservative governments have left the UK's problem with leaky homes unresolved

- [Labour says it will insulate 2m houses in first year](#)





# The new smart home

Advanced Internet of Things platform devices



## Display and CAD combined

Wifi enabled In Home Display and consumer access device (CAD) which connects to your smart meter; use with our Bright App, APIs, or build your own interface to your data.

# Glowmarkt API documentation

## Using the Bright Application as an Individual User

Last updated: 15 Oct 2021

Version: 1.7

This document explains how to use the Glowmarkt platform APIs for an individual user who manages one or more sets of data (more sets of data is when a single user /account has multiple installations, i.e. multiple locations with smart meters).

### Prerequisites

To retrieve data from the API you need to do the following:

1. Download the Bright App (see instructions below)
2. Create an account using Bright (sign up following the App instructions)
3. Set up your data
  - a. If you have Glow hardware, set it up (connected to both the meter and the internet)
  - b. If you have a SMETS 2 smart meter, complete the verification process within the Bright App (on first time login) and wait for confirmation via the App and in an email that the verification has passed. The verification process is required whether you purchase hardware or not. (Hildebrand can retrieve delayed half hourly consumption data from the DCC because we are a DCC Other User).
4. If, in addition to the Glowmarkt API, you would like access to MQTT please do the following:
  - a. email [support@glowmarkt.com](mailto:support@glowmarkt.com) stating that you wish to use the MQTT
  - a. provide the Username you used when you created your Bright account and
  - b. the MAC ID on your Glow CAD (either the GlowStick or the IHD/CAD)



# Bright

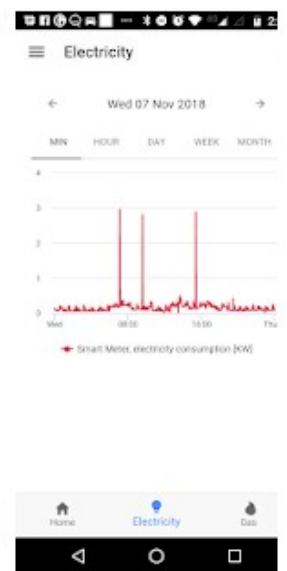
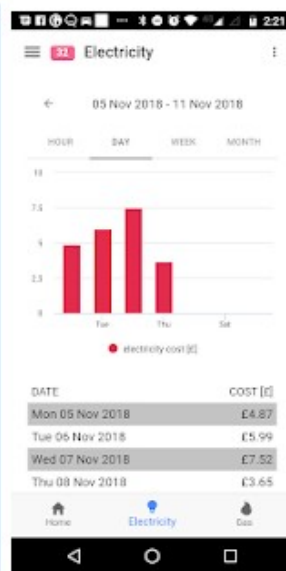
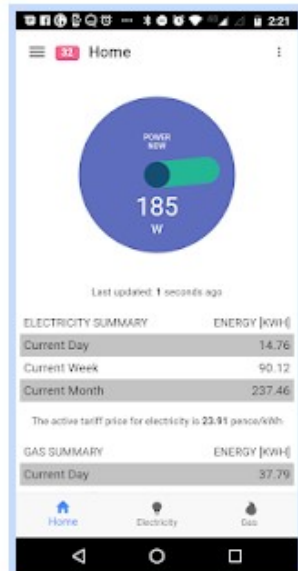
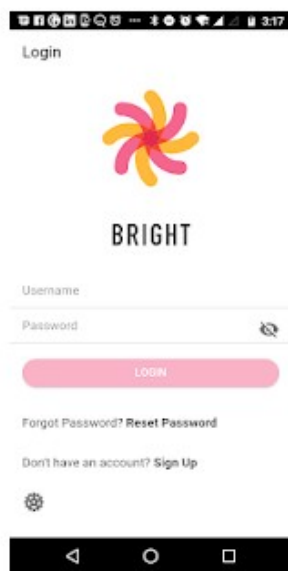
Hildebrand Technology Ltd House & Home

★★★★☆ 54

PEGI 3

This app is available for some of your devices

Installed



Monitor your energy consumption anytime, anywhere with Bright. For real-time data you need



## Step 1. Authenticate

To retrieve data from the Glowmarkt Platform a JWT token needs to be generated, this is delivered as part of the authentication process.

Authenticate as a user: provide your username and password and the applicationID listed below (it identifies you to us as an individual who seeks data retrieval functionality). Unless otherwise instructed, please use the following information:

<b>API req</b>	POST <a href="https://api.glowmarkt.com/api/v0-1/auth">https://api.glowmarkt.com/api/v0-1/auth</a>
<b>Headers</b>	Content-Type: application/json applicationId: b0f1b774-a586-4f72-9edd-27ead8aa7a8d
<b>Body</b>	<pre>{   "username": "your username",   "password": "your password" }</pre>
<b>Example cURL</b>	<pre>curl -X POST -H "Content-Type: application/json" -H "applicationId: b0f1b774-a586-4f72-9edd-27ead8aa7a8d" -d '{   "username": "your username",   "password": "your password" }' "https://api.glowmarkt.com/api/v0-1/auth"</pre>
<b>Swagger reference</b>	<a href="https://api.glowmarkt.com/api-docs/v0-1/usersys/usertypes/#/auth/usernamelogin">https://api.glowmarkt.com/api-docs/v0-1/usersys/usertypes/#/auth/usernamelogin</a>

The response will be in JSON, and will contain the JWT token. Please note that the generated token currently expires after 7 days.

Below is the API response you will receive following the above request - with an example of a JWT Token (in bold).

```
{  
  "valid": true,
```

# Convert [curl](#) commands to Python, JavaScript, PHP, R, Go, Rust, Elixir, Java, MATLAB, Dart, CFML, Ansible URI, Strest or JSON

## curl command

Examples: [GET](#) - [POST](#) - [Basic Auth](#)

```
curl -X POST -H "Content-Type: application/json" -H "applicationId: b0f1b774-a586-4f72-9edd-27ead8aa7a8d" -d |'{
"username": "your username",
"password": "your password"
}' "https://api.glowmarkt.com/api/v0-1/auth"
```

[Ansible](#) [CFML](#) [Dart](#) [Elixir](#) [Go](#) [Java](#) [JavaScript](#) [JSON](#) [Node.js](#) [MATLAB](#) [PHP](#) [Python](#) [R](#) [Rust](#) [Strest](#)

```
import requests

headers = {
    # Already added when you pass json= but not when you pass data=
    # 'Content-Type': 'application/json',
    'applicationId': 'b0f1b774-a586-4f72-9edd-27ead8aa7a8d',
}

json_data = {
    'username': 'your username',
    'password': 'your password',
}

response = requests.post('https://api.glowmarkt.com/api/v0-1/auth', headers=headers, json=json_data)

# Note: json_data will not be serialized by requests
# exactly as it was in the original request.
#data = '{\n"username": "your username",\n"password": "your password"\n}'
#response = requests.post('https://api.glowmarkt.com/api/v0-1/auth', headers=headers, data=data)
```

[Copy to clipboard](#)

```
def dump_data(username, password, kind='electricity consumption', output_filename='electricity.csv'):

    # step 1 - get authentication token
    print('getting token...')
    headers = { 'applicationId': 'b0f1b774-a586-4f72-9edd-27ead8aa7a8d' }

    json_data = {
        'username': username, 'password': password,
    }

    auth_response = requests.post('https://api.glowmarkt.com/api/v0-1/auth', headers=headers, json=json_data)
    token = auth_response.json()['token']
    print(f'...got token {token}\n\n')

    # step 2 - figure out which resources is the electricity usage
    print('getting resources...')
    headers['token'] = token

    resources_response = requests.get('http://api.glowmarkt.com/api/v0-1/virtualentity', headers=headers)
    for r in resources_response.json()[0]['resources']:
        if r['name'] == kind:
            resource_id = r['resourceId']
    print(f'.. found resource id {resource_id}')

    # step 3 - calculate start and end dates and write out data
    current_datetime = datetime.datetime.now().strftime("%Y-%m-%dT%H:%M:%S")
    start_datetime = (datetime.datetime.now() - datetime.timedelta(days = 31)).strftime("%Y-%m-%dT%H:%M:%S")

    print(f'retrieving data from {start_datetime} to {current_datetime}')
    response = requests.get(f'https://api.glowmarkt.com/api/v0-1/resource/{resource_id}/readings?from={start_datetime}')

    with open(output_filename, 'w') as output:
        output.write('date,usage\n')
        for timestamp, usage in response.json()['data']:
            d = datetime.datetime.fromtimestamp(timestamp)
            output.write(f'{d},{usage}\n')
```



```
In [22]: elec = pd.read_csv('electricity.csv', parse_dates=['date'], na_values=['None'])
elec.columns = ['date', 'electricity']
elec.head()

gas = pd.read_csv('gas.csv', parse_dates=['date'], na_values=['None'])
gas.columns = ['date', 'gas']
gas

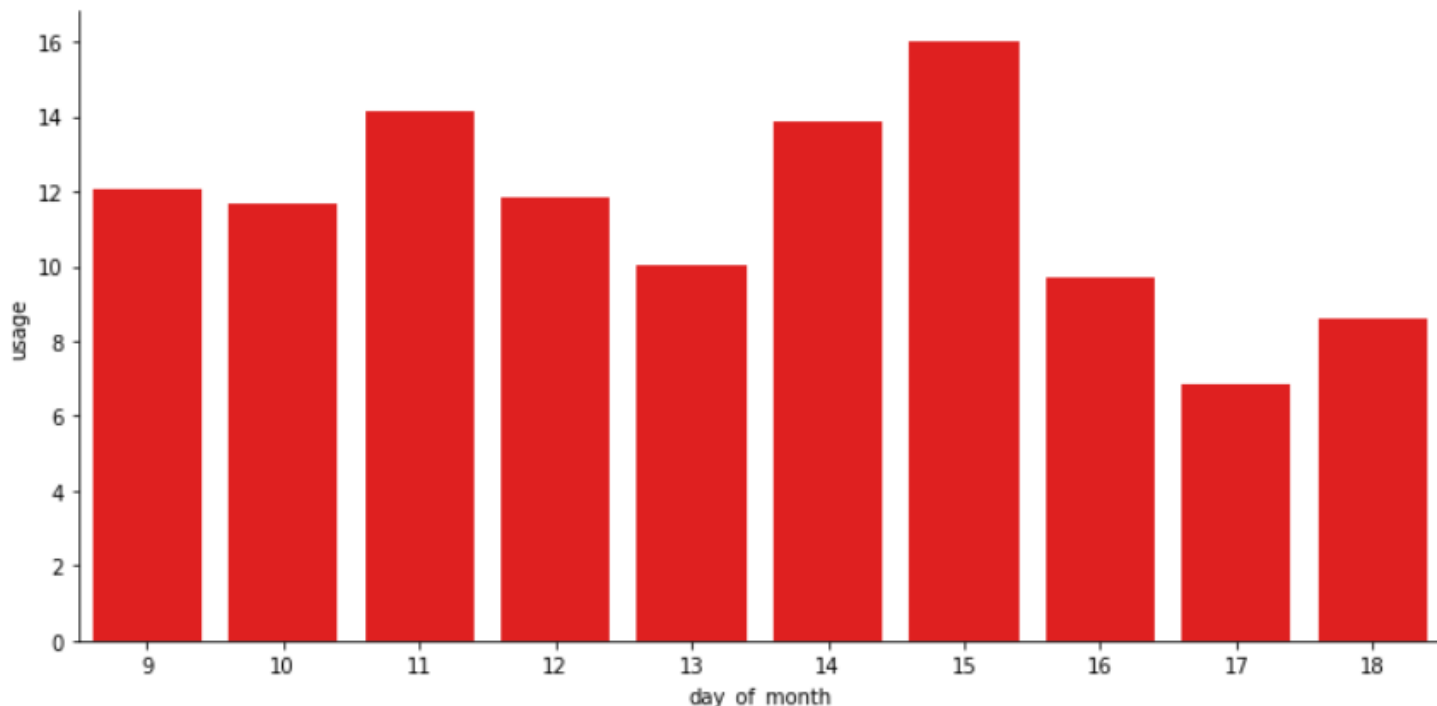
df = elec.merge(gas, left_on='date', right_on='date', how='outer').melt(id_vars=['date'], var_name='measurement')
df['day_name'] = df['date'].dt.day_name()
df['day_of_month'] = df['date'].dt.day
df['hour'] = df['date'].dt.hour
df['month'] = df['date'].dt.month_name()
df
```

Out[22]:

	date	measurement	usage	day_name	day_of_month	hour	month
0	2022-03-19 20:00:00	electricity	0.0	Saturday	19	20	March
1	2022-03-19 21:00:00	electricity	0.0	Saturday	19	21	March
2	2022-03-19 22:00:00	electricity	0.0	Saturday	19	22	March
3	2022-03-19 23:00:00	electricity	0.0	Saturday	19	23	March
4	2022-03-20 00:00:00	electricity	0.0	Sunday	20	0	March

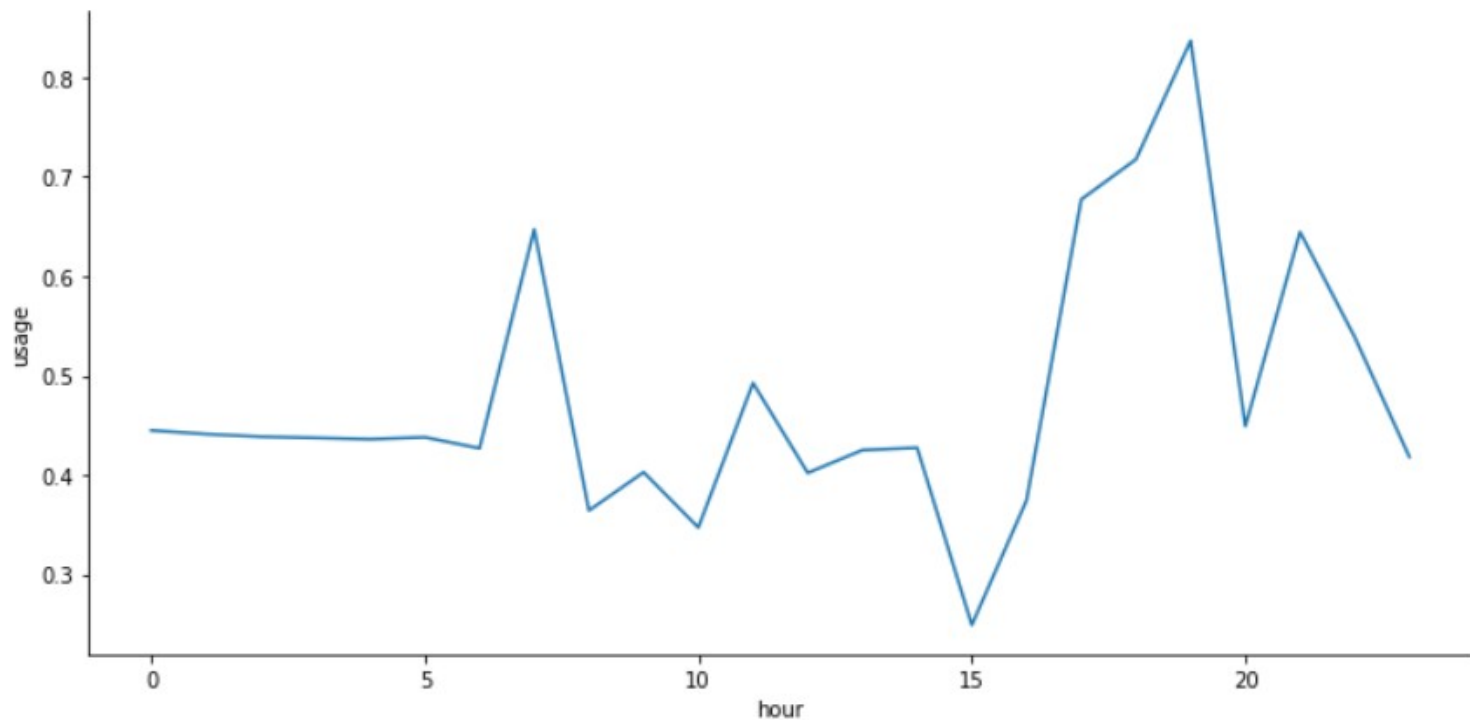
```
In [26]: sns.catplot(  
    data = just_electricity,  
    kind = 'bar',  
    x = 'day_of_month',  
    y = 'usage',  
    aspect = 2,  
    estimator=sum,  
    ci=None,  
    color='red'  
)
```

Out[26]: <seaborn.axisgrid.FacetGrid at 0x7f7ea40a8160>



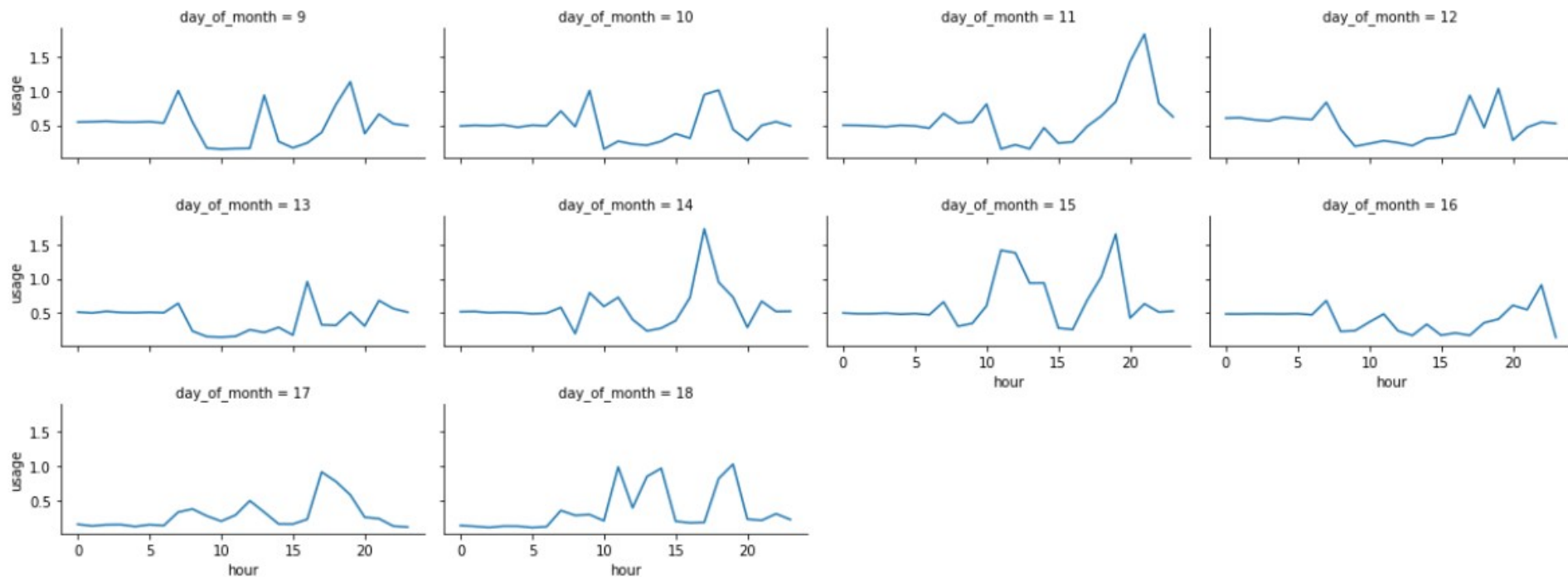
```
8]: sns.relplot(  
    data = just_electricity,  
    kind = 'line',  
    x = 'hour',  
    y = 'usage',  
    aspect = 2,  
    ci=None  
)
```

8]: <seaborn.axisgrid.FacetGrid at 0x7f7ea34409d0>



```
sns.relplot(
    data = just_electricity,
    kind = 'line',
    x = 'hour',
    y = 'usage',
    aspect = 2,
    estimator=sum,
    col = 'day_of_month',
    col_wrap = 4,
    height=2
)
```

<seaborn.axisgrid.FacetGrid at 0x7f7ea330b2e0>



```
import requests
import datetime
from sparklines import sparklines

def print_data(username, password, kind='electricity consumption', output_filename='electricity.csv'):

    # step 1 - get authentication token
    headers = { 'applicationId': 'b0f1b774-a586-4f72-9edd-27ead8aa7a8d' }

    json_data = {
        'username': username, 'password': password,
    }

    auth_response = requests.post('https://api.glowmarkt.com/api/v0-1/auth', headers=headers, json=json_data)
    token = auth_response.json()['token']

    # step 2 - figure out which resources is the electricity usage
    headers['token'] = token

    resources_response = requests.get('http://api.glowmarkt.com/api/v0-1/virtualentity', headers=headers)
    for r in resources_response.json()[0]['resources']:
        if r['name'] == kind:
            resource_id = r['resourceId']

    # step 3 - calculate start and end dates and write out data
    current_datetime = datetime.datetime.now().strftime("%Y-%m-%dT%H:%M:%S")
    start_datetime = (datetime.datetime.now() - datetime.timedelta(days = 10)).strftime("%Y-%m-%dT%H:%M:%S")

    response = requests.get(f'https://api.glowmarkt.com/api/v0-1/resource/{resource_id}/readings?from={start_datetime}')

    total = sum([d[1] for d in response.json()['data']])

    print(sparklines([d[1] for d in response.json()['data']])[0] + f' | 10 day total: {total} KWh')
```



```
martin@martin-prox14amd: ~  
File Actions Edit View Help  
martin@martin-prox14amd: ~  
██████████ | 10 day total: 103.085 KWh  
martin@martin-prox14amd:~$ █
```