

---

# **django-sheets Documentation**

*Release 0.1.2*

**George Whewell**

March 24, 2015



<b>1</b>	<b>Installation</b>	<b>1</b>
<b>2</b>	<b>Configuration</b>	<b>3</b>
<b>3</b>	<b>Usage</b>	<b>5</b>
3.1	Multi-sheet Documents . . . . .	6
<b>4</b>	<b>Credits</b>	<b>7</b>
4.1	Development Lead . . . . .	7
4.2	Contributors . . . . .	7
<b>5</b>	<b>History</b>	<b>9</b>
5.1	0.1.2 (2014-24-03) . . . . .	9
5.2	0.1.1 (2014-22-01) . . . . .	9
5.3	0.1.0 (2014-18-01) . . . . .	9



---

## Installation

---

To install the latest release from PyPI using pip:

```
$ pip install django-sheets
```

Add `sheets` to `INSTALLED_APPS` in `settings.py`:

```
INSTALLED_APPS = (  
    ...  
    'sheets',  
    ...  
)
```



---

## Configuration

---

Provided your Django project has a working cache backend, django-sheets will cache requests to the Google Sheets API for 5 minutes.

- Performance- making an extra HTTP request per page-view (or more) will lead to poor response times.
- Reliability- The Google Sheets API may experience failing requests- having a cached copy on your servers means this is less likely to affect you.
- Exceeding quota- By making excessive requests you risk exceeding the API Quota and having requests denied until the quota is refreshed, making your page unusable.

If you wish to disable this cache, add to your settings file:

```
SHEETS_CACHE_DISABLED = True
```

You can lower the cache timeout if you wish to sacrifice performance to lessen the chance of stale data, or extend it to improve performance and reduce server load:

```
# Set timeout to 1 hour  
SHEETS_CACHE_TIMEOUT = 3600
```





---

## Usage

---

To use django-sheets, you need to have a key to a publically-accessible Google Sheets spreadsheet. To find it, open your sheet and select *File -> Share*. Click *Get sharable link* from the dialog. You'll find the 44-character key as part of the URL.

For example, the sample link `https://docs.google.com/spreadsheets/d/1bJNR7SLqpzWJNvstNcFR4gtS-M7Bmn0` has key `1bJNR7SLqpzWJNvstNcFR4gtS-M7Bmn0D1X2lGTJpVGM`

Load the django-sheets template tags in your template:

```
{% load sheets %}
```

Assign the CSV data to a variable using the `{% csv %}` tag:

```
{% csv <key> as <variable_name> %}
```

Try it using the sample key above:

```
{% load sheets %}
{% csv "1uPsdCGUnUsf3d2xGHRGUUb7_k5IQPtBvfQY61u8Z8wE" as data %}
<table>
  <thead>
    <tr>
      {% for header in data.headers %}
        <th>{{ header }}</th>
      {% endfor %}
    </tr>
  </thead>
  <tbody>
    {% for row in data.rows %}
      <tr>
        {% for cell in row %}
          <td>{{ cell }}</td>
        {% endfor %}
      </tr>
    {% endfor %}
  </tbody>
</table>
```

View the output, you should see

Origin (English)	Name (English)	Origin (Native)	Name (Native)
Australia	Nicole Kidman	Australia	Nicole Kidman
Austria	Johann Strauss	Österreich	Johann Strauß
Belgium (Flemish)	Rene Magritte	België	René Magritte
Belgium (French)	Rene Magritte	Belgique	René Magritte
Belgium (German)	Rene Magritte	Belgien	René Magritte

## 3.1 Multi-sheet Documents

By default, `django-sheets` will fetch the first sheet in the document. To access other sheets, you can use the `gid` parameter:

```
{% csv <key> gid=<gid> as <variable_name> %}
```

To obtain the `gid` parameter, you must open the document in Google Docs, and select *Publish to Web*. From the drop-down, select the sheet you want and copy the `gid` from the link given.

**Credits**

---

## 4.1 Development Lead

- George Whewell <georgerw@gmail.com>

## 4.2 Contributors

None yet. Why not be the first?



---

## History

---

### 5.1 0.1.2 (2014-24-03)

- Added gid parameter for multi-sheet documents

### 5.2 0.1.1 (2014-22-01)

- Added .headers and .rows methods (aliases for [0] and [1:])

### 5.3 0.1.0 (2014-18-01)

- First release on PyPI.