

---

# **Tarbell Documentation**

*Release 0.9-beta1*

## **News Apps and David Eads**

November 19, 2013



---

# Contents

---

<b>1</b>	<b>Using Tarbell</b>	<b>3</b>
1.1	Tarbell tutorial . . . . .	3
1.2	Install Tarbell . . . . .	11
1.3	Create or install a project . . . . .	16
1.4	Building projects . . . . .	17
1.5	Using Google spreadsheets . . . . .	20
1.6	Publishing . . . . .	21
1.7	Managing projects . . . . .	21
<b>2</b>	<b>Reference</b>	<b>23</b>
2.1	Contributing . . . . .	23
2.2	Developing base templates . . . . .	23
2.3	Command line reference . . . . .	23
2.4	Configuration reference . . . . .	24



Tarbell is a simple content management system that makes it easy to collaboratively build and publish static websites. Tarbell is designed to help smart, technically inclined people learn to build beautiful websites and publish them with ease.

Read the Tarbell tutorial to get started.

Tarbell is named after Ida Tarbell, a distinguished muckraking journalist whose 1904 *The History of the Standard Oil Company* is a masterpiece of investigative reporting. [Read more about her on Wikipedia.](#)

Need help? [Join the email list.](#)



---

# Using Tarbell

---

## 1.1 Tarbell tutorial

Let's build a website about celebrated Chicago journalist Ethel Payne!

### 1.1.1 Install Tarbell

First you need Tarbell. Fair warning, you're going to have to run these commands in a terminal:

```
pip install tarbell==0.9b1
```

(Don't know how to install pip? <resource-tk> can help!)

### 1.1.2 Configure Tarbell

Got it? Now configure Tarbell:

```
tarbell configure
```

For our tutorial, say no to configuring Amazon and Google docs. It's optional!:

```
Configuring Tarbell. Press ctrl-c to bail out!
```

```
Would you like to create a Tarbell configuration in /Users/davideads/.tarbell? [Y/n] Y
```

```
Would you like to use Google spreadsheets [Y/n]? n
```

```
No worries! Don't forget you'll need to configure your context variables in each project's config.py
```

```
- Done configuring Google spreadsheets.
```

```
Would you like to set up Amazon S3? [Y/n] n
```

```
- Not configuring Amazon S3.
```

```
What is your Tarbell projects path? [Default: /Users/davideads/tarbell, 'none' to skip]
```

```
Directory exists!
```

Projects path is /Users/davideads/tarbell

- Done setting up projects path.
- + Adding Basic Bootstrap 3 template (<https://github.com/newsapps/tarbell-template>)
- + Adding Searchable map template (<https://github.com/eads/tarbell-map-template>)

- Done configuring project templates.

Creating /Users/davideads/.tarbell/settings.yaml

- Done configuring Tarbell. Type `tarbell` for help.

### 1.1.3 Create a new project

Now that you've got Tarbell configured, create a new project:

```
tarbell newproject
```

You'll need to answer a few questions:

```
tarbell newproject
```

What is the project's short directory name? (e.g. my\_project) ethelpayne

Where would you like to create this project? [/Users/davideads/tarbell/ethelpayne]

What is the project's full title? (e.g. My awesome project) Ethel Payne: A life in journalism

Pick a template

```
[1] Basic Bootstrap 3 template
    https://github.com/newsapps/tarbell-template
```

```
[2] Searchable map template
    https://github.com/eads/tarbell-map-template
```

Which template would you like to use? [1] 1

- Cloning <https://github.com/newsapps/tarbell-template> to /Users/davideads/tarbell/ethelpayne

Copying configuration file

Copying `_base/_spreadsheet.xlsx` to `tarbell.py`'s `DEFAULT_CONTEXT`

- Creating `tarbell.py` project configuration file

- Done copying configuration file

Setting up git remote repositories

- Renaming master to `update_project_template`

- Add and commit `tarbell.py`

What is the URL of your project repository? (e.g. `git@github.com:eads/myproject.git`, leave blank to s

- Not setting up remote repository. Use your own version control!



All done! To preview your new project, type:

```
tarbell switch ethelpayne
```

You got this!

### 1.1.4 Previewing your project locally

Well, you heard the machine, you got this. Run the switch command to fire up a preview server:

```
tarbell switch ethelpayne
```

After running switch, a local server will run on your system.

```
Switching to ethelpayne
Edit this project's templates at /Users/davideads/tarbell/ethelpayne
Running preview server...
```

```
Press ctrl-c to stop the server
* Running on http://127.0.0.1:5000/
* Restarting with reloader
```

Now visit <http://127.0.0.1:5000/> in a browser.

Ethel Payne: A life in journalism   Example link 1   Example link 2   Example dropdown ▾   Example form   Submit

---

### Development mode

You can modify this file by editing `/Users/davideads/tarbell/tarbelly2/index.html` .

This block will not publish when `tarbell publish` is invoked. You can disable it with `{% block preview %}{% endblock %}` in your project's index.html file.

Get rid of this sample content by creating `/Users/davideads/tarbell/tarbelly2/index.html` . Take a look at `/Users/davideads/tarbell/tarbelly2/_base/index.html` to see source code for the default page.

## Values

The key is `headline` and the value `Test headline`.

Print this headline in your code with something like `<strong>{{ headline }}</strong>` .

## Data

### Access data with a loop:

Use the worksheet name to access data.

```
{% for row in data %}
<p>
  <strong>{{ row.column1 }}</strong>:
  {{ row.column2 }}
</p>
{% endfor %}
```

Output:

```
row1, column1: row1, column2
```

```
row2, column1: row2, column2
```

### Access data by key

Requires a column named `key`.

```
<p>
  <strong>key1, column 1</strong>:
  {{ keyed_data.key1.column1 }}
</p>
<p>
  <strong>key2, column 2</strong>:
  {{ keyed_data.key2.column2 }}
</p>
```

Output:

```
key1, column 1: key1, column1
```

```
key2, column 2: key2, column2
```

### 1.1.5 Add some context variables

First, set some project data in `/path/to/project/tarbell_config.py` (in this case `/Users/davideads/tarbell/ethelpayne/tarbell_config.py`). Open the file in your favorite editor. It should look like this:

```
# -*- coding: utf-8 -*-

"""
Tarbell project configuration
"""

# Short project name
NAME = "ethelpayne"

# Descriptive title of project
TITLE = "Ethel Payne: A life in journalism"

# Google spreadsheet key
#SPREADSHEET_KEY = "None"

# Create JSON data at ./data.json, disabled by default
# CREATE_JSON = True

# S3 bucket configuration
S3_BUCKETS = {
    # Provide target -> s3 url pairs, such as:
    # "mytarget": "s3://mys3url.bucket.url/some/path"
    "staging": "s3://projects.beta.coolorg.net/tarbelly2/",
    "production": "s3://projects.coolorg.net/tarbelly2/",
}

# Repository this project is based on (used for updates)
TEMPLATE_REPO_URL = "https://github.com/newsapps/tarbell-template"

# Default template variables
DEFAULT_CONTEXT = {
    'data': [
        { 'column1': u'row1, column1',
          'column2': u'row1, column2' },
        { 'column1': u'row2, column1',
          'column2': u'row2, column2' }],
    'google_analytics_id': None,
    'headline': u'Test headline',
    'keyed_data': {
        'key1': {
            'column1': u'key1, column1',
            'column2': u'key1, column2',
            'key': u'key1' },
        'key2': {
            'column1': u'key2, column1',
            'column2': u'key2, column2',
            'key': u'key2' }},
    'name': u'ethelpayne',
    'opengraph_description': None,
    'opengraph_image': None,
    'title': u'Ethel Payne: A life in journalism',
    'twitter_description': None
}
```

Edit the last section to include a new variable:

```

DEFAULT_CONTEXT = {
    # ...
    'title': u'Ethel Payne: A life in journalism',
    'twitter_description': None,
    'quote': ("I stick to my firm, unshakeable belief that the black "
              "press is an advocacy press, and that I, as a part of that "
              "press, can't afford the luxury of being unbiased ... when "
              "it come to issues that really affect my people, and I plead "
              "guilty, because I think that I am an instrument of change."),
    'quote_author': u'Ethel Payne',
}

```

Now copy `_base/index.html` to your project's root directory. It should look like:

```

{% extends "_base.html" %}

{% block content %}
<div class="container">
  <p><em>Get rid of this sample content by creating
    <code>{{ PROJECT_PATH }}/{{ PATH }}/code.</code>.</em> Take a look at
    <code>{{ PROJECT_PATH }}/_base/index.html</code> to see source
    code for the default page.
  </p>

  <h1>Values</h1>

  <div class="jumbotron">
    <p>The key is <code>headline</code> and the value <strong>{{ headline }}</strong>.</p>
    <p>Print this headline in your code with something like <code>{{ "<strong>{{ headline }}</strong>"</code>
  </div>

  <h1>Data</h1>

  <div class="row">
    <div class="col-md-6">
      <h2>Access data with a loop:</h2>
      <p><em>Use the worksheet name to access data.</em></p>

      <pre>
{{ "{% for row in data %}" }}
{{ "<p>" }}
{{ "  <strong>{{ row.column1 }}</strong>:" }}
{{ "  {{ row.column2 }}" }}
{{ "</p>" }}
{{ "{% endfor %}" }}
</pre>

      <p>Output:</p>

      {% for row in data %}
      <p>
        <strong>{{ row.column1 }}</strong>
        {{ row.column2 }}
      </p>
      {% endfor %}
    </div>
    <div class="col-md-6">
      <h2>Access data by key</h2>
      <p><em>Requires a column named <code>key</code>.</em></p>

```

```
<pre>
{{ "<p>" }}
{{ " <strong>key1, column 1</strong>:" }}
{{ "  {{ keyed_data.key1.column1 }}" }}
{{ "</p>" }}
{{ "<p>" }}
{{ " <strong>key2, column 2</strong>:" }}
{{ "  {{ keyed_data.key2.column2 }}" }}
{{ "</p>" }}
</pre>
```

```
<p>Output:</p>
<p>
  <strong>key1, column 1</strong>:
  {{ keyed_data.key1.column1 }}
</p>
<p>
  <strong>key2, column 2</strong>:
  {{ keyed_data.key2.column2 }}</p>
</div>
</div>
{% endblock content %}
```

Let's put your quote right at the top. Add a snippet right after `<div class="container">`:

```
{% block content %}
<div class="container">

  {# Ethel Payne quote #}
  <div class="jumbotron">
    <h2>"{{ quote }}"</h2>
    <small>{{ quote_author }}"</small>
  </div>

  {# ... #}

</div>
{% endblock content %}
```

Reload the server at <http://127.0.0.1:5000> in your web browser to see your changes!

Ethel Payne: A life in journalism

Example link 1

Example link 2

Example dropdown ▾

Example form

Submit

## Development mode

You can modify this file by editing `/Users/davideads/tarbell/tarbelly2/index.html`.

This block will not publish when `tarbell publish` is invoked. You can disable it with `{% block preview %}{% endblock %}` in your project's `index.html` file.

"I stick to my firm, unshakeable belief that the black press is an advocacy press, and that I, as a part of that press, can't afford the luxury of being unbiased ... when it come to issues that really affect my people, and I plead guilty, because I think that I am an instrument of change."

Ethel Payne

Get rid of this sample content by creating `/Users/davideads/tarbell/tarbelly2/index.html`. Take a look at `/Users/davideads/tarbell/tarbelly2/_base/index.html` to see source code for the default page.

## Values

The key is `headline` and the value `Test headline`.

Print this headline in your code with something like `<strong>{{ headline }}</strong>`.

## Data

### Access data with a loop:

Use the worksheet name to access data.

```
{% for row in data %}
<p>
  <strong>{{ row.column1 }}</strong>:
  {{ row.column2 }}
</p>
{% endfor %}
```

Output:

**row1, column1:** row1, column2

**row2, column1:** row2, column2

### Access data by key

Requires a column named `key`.

```
<p>
  <strong>key1, column 1</strong>:
  {{ keyed_data.key1.column1 }}
</p>
<p>
  <strong>key2, column 2</strong>:
  {{ keyed_data.key2.column2 }}
</p>
```

Output:

**key1, column 1:** key1, column1

**key2, column 2:** key2, column2

### 1.1.6 Building a page

The default `index.html` sure has a lot of stuff in it. Let's replace it with a simplified version:

```
{% extends "_base.html" %}

{% block content %}
<div class="container">
  {# All your content, Bootstrap columns... #}
</div>
{% endblock content %}
```

```
{% block css %}
{{ super() }} {# Calls base css block to include Bootstrap and base css #}
<script src="css/style.css"></script>
{% endblock scripts %}

{% block scripts %}
{{ super() }} {# Calls base scripts block to include jQuery and Bootstrap #}
<script src="//cdnjs.cloudflare.com/ajax/libs/moment.js/2.2.1/moment.min.js"></script>
{% endblock scripts %}
```

There are a couple of key points to note in this simple template...

- This is a Jinja template – you can do anything that Jinja templates can do. [Learn more about Jinja](#).
- The template extends the template defined in `_base/_base.html`. All files in `_base` are available as if they were on the project root. Base templates provide a way to keep boilerplate code to a minimum. Build a base template for your team or organization
- The base template defines *blocks* to be customized.
- Comments can use Jinja syntax (`{# ... #}`)
- To get the default block behavior *and* extend a block, use `{{ super() }}`.

Now start editing the content block and scripts blocks. We're just doing some silly stuff here to show off how you can start using `DEFAULT_CONTEXT` and some of the common blocks:

```
{% extends "_base.html" %}

{% block content %}
<div id="#content" class="container">
  <table class="table">
    <tbody>
      {% for row in data %}
      <tr>
        <td>{{ row.column1 }}</td>
        <td>{{ row.column2 }}</td>
      </tr>
      {% endfor %}
    </tbody>
  </table>
</div>
{% endblock content %}

{% block css %}
{{ super() }} {# Calls base css block to include Bootstrap and base css #}
<script src="css/style.css"></script>
{% endblock scripts %}

{% block scripts %}
{{ super() }} {# Calls base scripts block to include jQuery and Bootstrap #}
<script src="//cdnjs.cloudflare.com/ajax/libs/moment.js/2.2.1/moment.min.js"></script>
<script type="text/javascript">
  $(document).ready(function() {
    $('#content').append(moment().format("MMMM Do YYYY, h:mm:ss"));
  });
</script>
{% endblock scripts %}
```

### 1.1.7 Customizing page elements

You'll notice there's a nav bar at the top of the page. Instead of overriding a block, complex elements like the nav can be included in the base template as template snippets. In `_base` you'll see a file called `_nav.html`. Copy that file to your root directory and start editing. Your version of the `_nav.html` will take precedence over the version defined in `_base.html`.

## 1.2 Install Tarbell

### 1.2.1 Install Tarbell with `pip install tarbell==0.9b1`

```
pip install tarbell==0.9b1
```

*You must specify the version for compatibility with pip 1.4+ until the Tarbell 0.9 stable release.*

### 1.2.2 Configure Tarbell with `tarbell configure`

The `tarbell configure` command will set up your Tarbell settings:

```
tarbell configure
```

Please consider setting up Google spreadsheet access for collaborative data editing and Amazon S3 settings for easy publishing.

### 1.2.3 Configure Google spreadsheet access (optional)


In order to allow Tarbell to create new Google Spreadsheets, you'll need to download a `client_secrets.json` file to access the Google Drive API. You can share this file with collaborators and within your organization, but do not share this file anywhere public.

Log in to the [Google API Developer Console](#) and create a new project:



### Start using the Google APIs console

to manage your API usage



Creating an **APIs project** will let you:

- Use Google APIs **beyond anonymous limits**.
- **Monitor** API usage and **control** API access.
- **Share** API management with a team.

**Create project...**

Now click the “Services” tab and enable Google Drive API.

Tarbell

All (61)
Active (1)
Inactive (60)
Google Cloud Platform

Overview

Services

Team

API Access

Reports

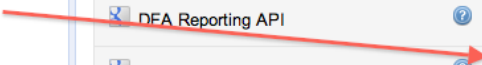
Quotas

#### All services

Select services for the project.

Service	Status	Notes
Ad Exchange Buyer API	<input type="checkbox"/> OFF	Courtesy limit: 1,000 requests/day
Ad Exchange Seller API	<input type="checkbox"/> OFF	Courtesy limit: 10,000 requests/day
Admin SDK	<input type="checkbox"/> OFF	Courtesy limit: 150,000 requests/day
AdSense Host API	<a href="#">Request access...</a>	Courtesy limit: 100,000 requests/day
AdSense Management API	<input type="checkbox"/> OFF	Courtesy limit: 10,000 requests/day
Analytics API	<input type="checkbox"/> OFF	Courtesy limit: 50,000 requests/day
Audit API	<input type="checkbox"/> OFF	Courtesy limit: 10,000 requests/day
BigQuery API	<input type="checkbox"/> OFF	Courtesy limit: 10,000 requests/day • <a href="#">Pricing</a>
Blogger API v3	<a href="#">Request access...</a>	Courtesy limit: 10,000 requests/day
Books API	<input type="checkbox"/> OFF	Courtesy limit: 1,000 requests/day
Calendar API	<input type="checkbox"/> OFF	Courtesy limit: 10,000 requests/day
Custom Search API	<input type="checkbox"/> OFF	Courtesy limit: 100 requests/day • <a href="#">Pricing</a>
DFA Reporting API	<input type="checkbox"/> OFF	Courtesy limit: 10,000 requests/day
Drive API	<input checked="" type="checkbox"/> ON	Courtesy limit: 10,000,000 requests/day

Enable Drive API



Click the “API Access” tab to create a client ID:



**Google apis**

API Project

- Overview
- Services
- Team
- API Access**
- Reports
- Quotas

### API Access

To prevent abuse, Google places limits on API requests. Using a valid OAuth token or API key allows you to exceed anonymous limits by connecting requests back to your project.

#### Authorized API Access

OAuth 2.0 allows users to share specific data with you (for example, contact lists) while keeping their usernames, passwords, and other information private. A single project may contain up to 20 client IDs. [Learn more](#)

**Create an OAuth 2.0 client ID...**

Click the big old button

#### Simple API Access

Use API keys to identify your project when you do not need to access user data. [Learn more](#)

**Key for browser apps (with referers)**

API key:

Referers: Any referer allowed

Activated on: May 25, 2013 2:38 PM

Activated by: davideads@gmail.com – you

Generate new key...  
Edit allowed referers...  
Delete key...

Create new Server key... Create new Browser key... Create new Android key... Create new iOS key...

Add some project details. These don't really matter:

**Google apis**

API Project

- Overview
- Services
- Team
- API Access**
- Reports
- Quotas

### API Access

To prevent abuse, Google places limits on API requests. Using a valid OAuth token or API key allows you to exceed anonymous limits

#### Create Client ID

Branding Information  
The following information will be shown to users whenever you request access to their private data using your new client ID.

Product name: Tarbell

Google account: davideads@gmail.com - you  
Link your project to this account's profile and reputation.

Product logo: http://example.com/example\_logo.png

Update  
Max size: 120x60 pixels

Home Page URL:

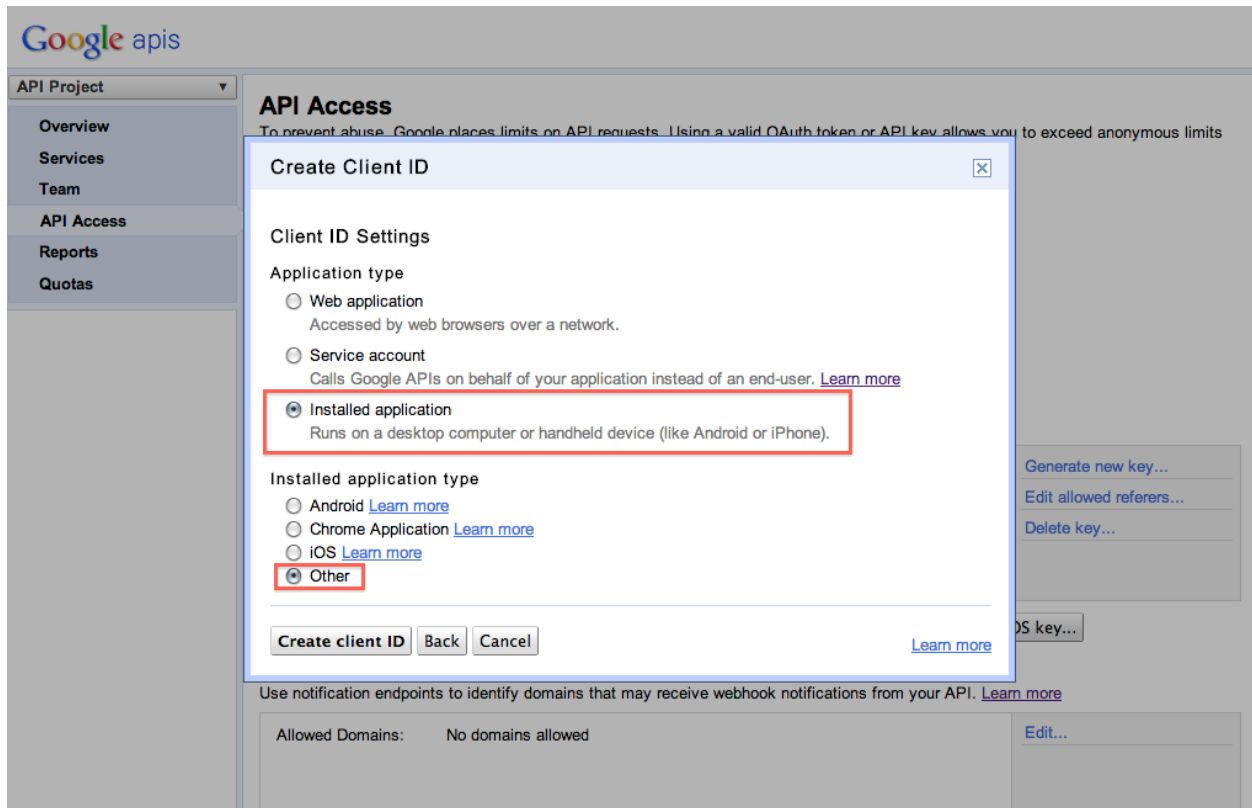
Next Cancel [Learn more](#)

Use notification endpoints to identify domains that may receive webhook notifications from your API. [Learn more](#)

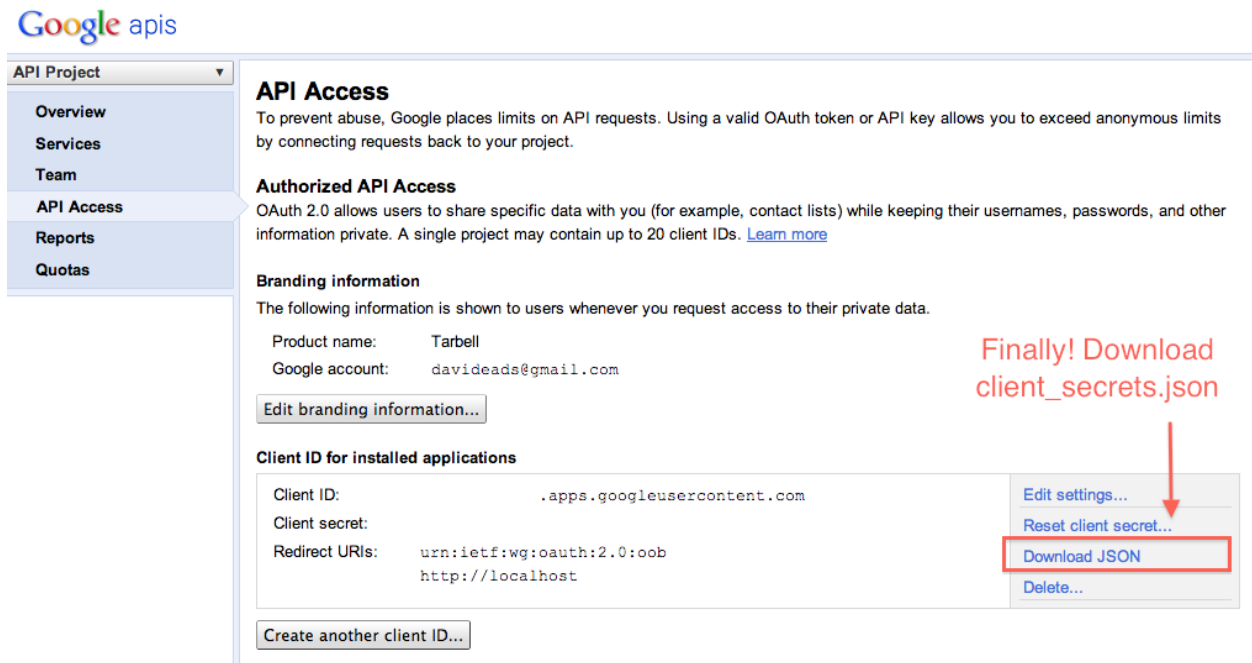
Allowed Domains: No domains allowed [Edit...](#)

Generate new key...  
Edit allowed referers...  
Delete key...  
OS key...

This is the important screen. Select “installed app” and “other”:



Whew! Now you can download the `client_secrets.json` file:



Copy `client_secrets.json` to `~/tarbell/client_secrets.json` or specify the download location when running `tarbell configure`.

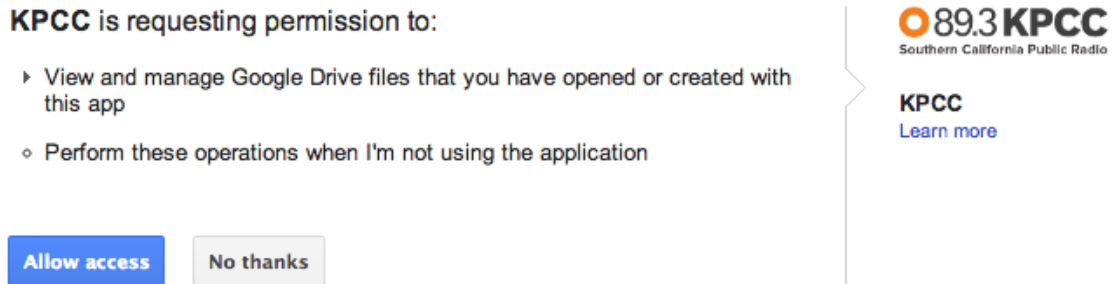
The first time a Tarbell command needs access to a Google spreadsheet, you'll be prompted to authenticate:

Go to the following link in your browser:

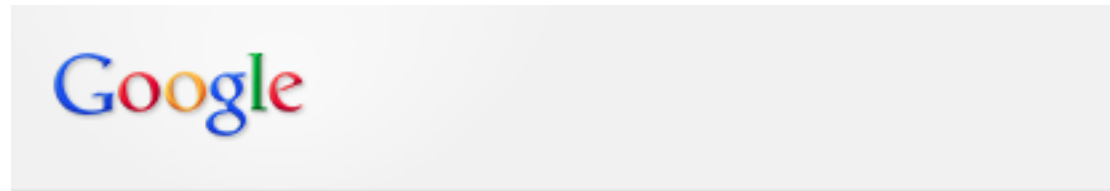
<https://accounts.google.com/o/oauth2/auth?scope=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fdrive&>

Enter verification code:

Follow the link:



You should receive a confirmation code:



Please copy this code, switch to your application and paste it there:

`4/rouLqZuRBplmRRQREZBXO_vauzRw.4oeATajeFpccmm!`

Enter it. If it works, you'll see:

Authentication successful.

Now you can access and create Google spreadsheets to use with Tarbell projects.

## 1.2.4 Configuring Amazon S3

Learn how to [set up Amazon S3](#) in the Small Data Journalism guide.

Configure `s3cmd`:

```
s3cmd --configure
```

`tarbell configure` will do this for you:

```
Would you like to set up Amazon S3? [Y/n] y
```

```
Calling s3cmd --configure
```

```
<s3cmd output, be sure to answer y at the end>
```

What is your default staging bucket? (e.g. s3://apps.beta.myorg.com/, leave blank to skip) s3://proj

What is your default production bucket? (e.g. s3://apps.myorg.com/, leave blank to skip) s3://project

- Done configuring Amazon S3.

## 1.3 Create or install a project

### 1.3.1 Create a new project with *tarbell newproject <projectname>*

Run:

```
tarbell newproject mynewproject
```

You'll be asked a few questions:

Creating nellie-bly

Where would you like to create this project? [/Users/davideads/tarbell/nellie-bly]

What is the project's full title? (e.g. My awesome project) The Story of Nellie Bly

Pick a template

```
[1] Basic Bootstrap 3 template
    https://github.com/newsapps/tarbell-template
```

```
[2] Searchable map template
    https://github.com/eads/tarbell-map-template
```

Which template would you like to use? [1] 1

- Cloning <https://github.com/newsapps/tarbell-template> to /Users/davideads/tarbell/nellie-bly

Copying configuration file

Copying `_base/_spreadsheet.xlsx` to `tarbell.py`'s `DEFAULT_CONTEXT`

- Creating `tarbell.py` project configuration file

- Done copying configuration file

Setting up git remote repositories

- Renaming master to `update_project_template`

- Add and commit `tarbell.py`

What is the URL of your project repository? (e.g. `git@github.com:eads/myproject.git`, leave blank to s

Creating new remote 'origin' to track `git@github.com:eads/nellie-bly.git`.

Warning: Don't forget! It's up to you to create this remote and push to it.

All done! To preview your new project, type:

```
tarbell switch nellie-bly
```

You got this!

You can now open up ‘ /Users/davideads/tarbell/nellie-bly ‘ and start editing the “index.html” file.

### 1.3.2 Install an existing project with *tarbell install* <repository-url>

The project must include a *tarbell.py* file and be able to be cloned with Git.

## 1.4 Building projects

### 1.4.1 Editing templates

Tarbell projects consist of simple HTML pages that may use Jinja templating features.

If you create a file in your project directory called *chapter1.html*, you’ll be able to preview the file at <http://localhost:5000/chapter1.html> and publish to the same file. This file can be straight up HTML, or it can inherit from a base template.

Files and directories that start with an underscore ( `_` ) or a dot ( `.` ) will not be rendered by the preview server or included in the generated static HTML.

### 1.4.2 Understanding the base template

Base templates live in your projects *\_base* directory, and use Jinja templating features to make your life easier. Develop base templates to use for projects that need to share boilerplate code like advertising, analytics, and common page elements. Tarbell projects are intended to inherit from base templates.

Here’s a simple *\_base/\_base.html*:

```
<html>
  <head>
    <title>{{ title }}</title>
    {% block css %}
    <link rel="stylesheet" type="text/css" href="css/style.css" />
    {% endblock css %}
  </head>
  <body>
    {% block content %}{% endblock content %}
  </body>
</html>
```

To inherit from this template extend the base template in *index.html* or other project files you create. Now, all your *index.html* needs to contain is:

```
{% block content %}
<h1>{{ title }} </h1>
{{ content|markdown }}
{% endblock content %}
```

You might notice we’re using the `markdown` filter. Base templates also define filters. See building base templates for more.

If a base template defines a static file or template (e.g. `_base/style.css`), it will be available relative to the project's base path (e.g. `http://127.0.0.1:5000/style.css`). If a project defines a file with the same name, the project's version will be used instead.

See the basic Tarbell template for a simple implementation of a base template.

### 1.4.3 Configuring projects

Project configuration is kept in the `tarbell.py` file in your project's base directory:

```
# -*- coding: utf-8 -*-

"""
Tarbell project configuration
"""

# Short project name
NAME = "nellie-bly"

# Descriptive title of project
TITLE = "The Story of Nellie Bly"

# Google spreadsheet key
# SPREADSHEET_KEY = "0Ak3IIavLYTovdC1qMFo5UDEwcUhQZmdZbkk4WW1sYUE"

# S3 bucket configuration
S3_BUCKETS = {
    "staging": "s3://projects.beta.myorg.tld/profiles/nellie-bly/",
    "production": "s3://projects.myorg.tld/profiles/nellie-bly/",
}

# Repository this project is based on (used for updates)
TEMPLATE_REPO_URL = "https://github.com/newsapps/tarbell-template"

# Default template variables
DEFAULT_CONTEXT = {
    'name': 'nellie-bly',
    'title': 'The Story of Nellie Bly'
}
```

`TITLE` and `NAME` are required and describe the project.

If specified, `SPREADSHEET_KEY` will be used as data source if Google Spreadsheets is configured.

If specified, `S3_BUCKETS` should be a Python dict consisting of `targetname->'targeturl'` pairs.

If specified, `TEMPLATE_REPO_URL` will be used to pull in updates to the base template.

If specified, `DEFAULT_CONTEXT` will provide context variables to the template. The default context is dictionary of `key->'value'` pairs to provide to the template. The `value` may be any Python object that can be represented as a Jinja template variable.

### 1.4.4 Using context variables

Template data comes from Google spreadsheets or `tarbell.py`'s `DEFAULT_CONTEXT`.

This simple `DEFAULT_CONTEXT` shows many of the key template features:

```

DEFAULT_CONTEXT = {
    'name': 'nellie-bly',
    'title': 'The Story of Nellie Bly',
    'font_size': '20px',
    # Nested dictionary
    'photos': {
        'intro': {
            'url': 'img/bly01.jpg',
            'caption': 'A caption',
        }
    },
    # Nested list
    'timeline': [
        {'year': '1902', 'description': 'Description...'},
        {'year': '1907', 'description': 'Description...'},
        {'year': '1909', 'description': 'Description...'},
    ],
}

```

To print the title in your template, use `{{ title }}`:

```
<h1>{{ title }}</h1>
```

Address a nested dictionary:

```

<aside>{{ photos.intro.caption }}</aside>
```

Access a list of data:

```
<ul>
  {% for year in timeline %}
  <li><strong>{{ year }}</strong>: {{ description }}</li>
  {% endfor %}
</ul>
```

### 1.4.5 Where can context variables be used?

Context variables can be used in HTML, CSS, and Javascript files. If the text file causes a Jinja template error (which can happen if the file has Jinja-like markers), the file will be served as static and the preview server will log an error.

This means that CSS and Javascript files may include variables. *style.css* might include:

```
#content { font-size: {{ font_size }}; }
```

Similarly, a Javascript file could include:

```
var data = {{ photos|tojson }}
console.log(photos.intro.url);
```

Use this feature with care! Missing variables could easily break your CSS or Javascript.

## 1.5 Using Google spreadsheets

### 1.5.1 The *values* worksheet

The values worksheet must have “key” and “value” columns. These key-value pairs will be provided as global variables to templates. So if there’s a row with a key column value of “foo” and a value of “bar”, `{{ foo }}` in a template will print bar.

Take this sample worksheet:

key	value
title	Project title
intro	Project intro

A *values* worksheet that contains this data provides the `{{ title }}` and `{{ intro }}` variables to the template.

Use them in your templates:

```
<h2>{{ title }}</h2>
<p class="intro">{{ intro }}</p>
```

### 1.5.2 Named worksheets

Other worksheets can hold any kind of data support by Google spreadsheets. These variables can be accessed by their worksheet name.

If there is no *key* column in the worksheet, the worksheet can be accessed as a list. Imagine a spreadsheet named *cars* with these values:

model	mpg
Civic	25.9
Accord	28.1
Element	24.6

You can access these variables in your spreadsheet with a loop:

```
{% for car in cars %}
  <h3>{{ car.model }}</h3>
  <p>MPG: {{ car.mpg }}</p>
{% endfor %}
```

If a column named *key* does exist, elements may be accessed by key. Imagine a spreadsheet named *companies* with these values:

key	name	country
ford	Ford	U.S.A.
honda	Honda	Japan
volvo	Volvo	Sweden

You can access these variables by their key name:

```
<p>{% manufacturers.ford.name %} is from {% manufacturers.ford.country %}</p>
```

### 1.5.3 Worksheet, column, and key names are slugified

Spaces and dashes are replaced with underscores (\_). Non alphanumeric characters are removed. Case is preserved.

Examples of names that will be transformed:



- *My Worksheet* becomes *My\_Worksheet*
- *My key* becomes *My\_key*
- *my-Column* becomes *my\_Column*

Names that will not be transformed:

- *MyColumn* remains *MyColumn*
- *mycolumn* remains *mycolumn*
- *my\_column* remains *my\_column*

### 1.5.4 Worksheets, columns, and keys names preceded by `_` (underscore) are ignored

Precede any worksheet name, column name, or key with an underscore to hide it from your templates and JSON data.

## 1.6 Publishing

1.6.1 Manually publish projects with *tarbell generate <output\_dir>*

1.6.2 Publish projects with *tarbell publish <staging/production/target>*

1.6.3 Remove projects with *tarbell unpublish <staging/production/target>*

## 1.7 Managing projects

1.7.1 Show all projects with *tarbell list*

1.7.2 Switch to a project with *tarbell switch <projectname>*

1.7.3 Run preview server with *tarbell serve*

1.7.4 Update a project's base template with *tarbell update*



---

# Reference

---

## 2.1 Contributing

- Github repository: <https://github.com/newsapps/flask-tarbell>

## 2.2 Developing base templates

### 2.2.1 Fork a base template or create your own

Base templates should consist of `_base/base.py` and any additional files.

### 2.2.2 Defining filters and context processors in `_base/base.py`

## 2.3 Command line reference

### 2.3.1 *tarbell configure*

### 2.3.2 *tarbell newproject*

### 2.3.3 *tarbell serve*

Requires current directory to be a Tarbell project.

### 2.3.4 *tarbell publish*

Requires current directory to be a Tarbell project.

### 2.3.5 *tarbell unpublish*

Requires current directory to be a Tarbell project.

### 2.3.6 *tarbell update*

Requires current directory to be a Tarbell project.

### 2.3.7 *tarbell generate*

Requires current directory to be a Tarbell project.

### 2.3.8 *tarbell switch*

### 2.3.9 *tarbell list*

### 2.3.10 *tarbell install*

### 2.3.11 *tarbell install-template*

## 2.4 Configuration reference

### 2.4.1 Tarbell settings (*~/.tarbell/settings.yaml*)

The settings file uses a simple YAML-based format:

```
google_account: googleaccount@gmail.com
projects_path: /Users/davideads/tarbell
project_templates:
- name: Basic Bootstrap 3 template
  url: https://github.com/newsapps/tarbell-template
- name: Searchable map template
  url: https://github.com/eads/tarbell-map-template
s3_buckets:
  production: s3://projects.coolorg.net/
  staging: s3://projects.beta.coolorg.net/
```

### 2.4.2 Google SDK client secrets (*~/.tarbell/client\_secrets.json*)

Place the client secrets file in *~/.tarbell*.