

**Massively increase your productivity on
personal projects with comprehensive
documentation and automated tests**

Simon Willison, DjangoCon US 2022

Talk notes are linked from [github.com/simonw](#)

... aka ...

Simon Willison, DjangoCon US 2022

Talk notes are linked from github.com/simonw

Coping strategies for the serial project hoarder

Simon Willison, DjangoCon US 2022

Talk notes are linked from github.com/simonw



ID: ffg630c99190





Search projects



Help

Sponsors

Log in

Register



Simon Willison

simonw

Joined Oct 26, 2017

Statistics

View statistics for simonw's projects via [Libraries.io](#), or by using [our public dataset on Google BigQuery](#)

185 projects

shot-scrapers

Last released Oct 15, 2022

A command-line utility for taking automated screenshots of websites

asgi-gzip

Last released Oct 13, 2022

gzip middleware for ASGI applications, extracted from Starlette

datasette-public

Last released Oct 7, 2022

Make specific Datasette tables visible to the public

datasette-sentry

Last released Oct 6, 2022

Datasette plugin for configuring Sentry

datasette-search-all

Last released Oct 4, 2022

Datasette plugin for searching all searchable tables at once

datasette-publish-fly

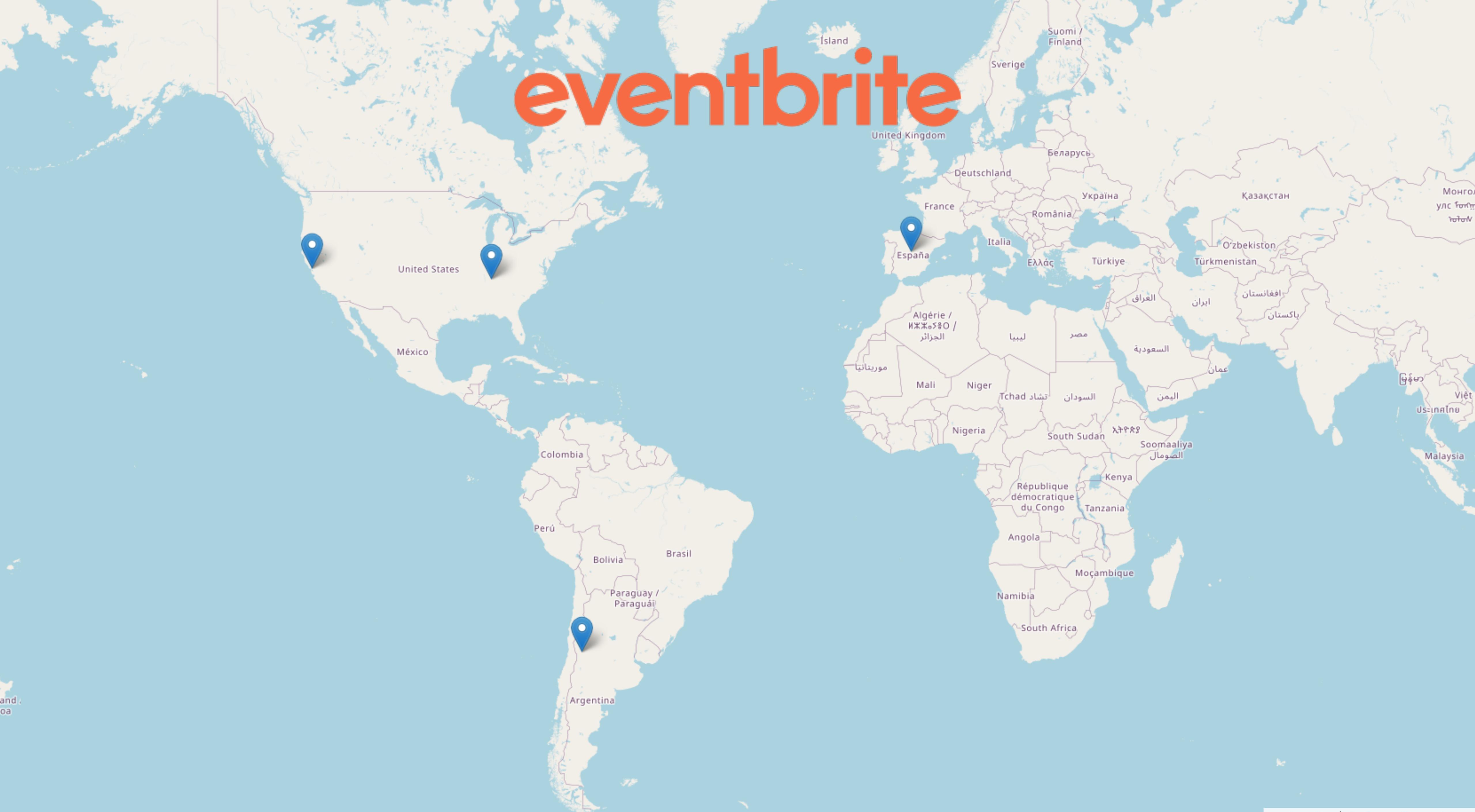
Last released Sep 29, 2022

Datasette plugin for publishing data using Fly

datasette-pretty-json

Last released Sep 28, 2022

eventbrite



The perfect commit

Implementation + Tests + Documentation

and a link to an issue thread

Async support for prepare_jinja2_environment, closes #1809

Loading branch information.

[Browse files](#) simonw committed on Sep 16

1 parent 2ebcff... commit ddc999ad1296e8c69cffede3e367dda059b8adad

 Showing 10 changed files with 76 additions and 9 deletions.[Split](#)[Unified](#) Filter changed files datasette app.py utils testing.py docs plugin_hooks.rst testing_plugins.rst tests fixtures.py plugins my_plugin.py my_plugin_2.py test_internals_datasette_c... test_plugins.py test_routes.py 22 datasette/app.py

@@ -208,6 +208,7 @@ def __init__(

208 208 crossdb=False,

209 209 nolock=False,

210 210):

211 211 + self._startup_invoked = False

212 212 assert config_dir is None or isinstance(

213 213 config_dir, Path

214 214), "config_dir= should be a pathlib.Path"

@@ -344,9 +345,6 @@ def __init__(

344 345 self.jinja_env.filters["quote_plus"] = urllib.parse.quote_plus

345 346 self.jinja_env.filters["escape_sqlite"] = escape_sqlite

346 347 self.jinja_env.filters["to_css_class"] = to_css_class

347 348 # pylint: disable=no-member

348 349 pm.hook.prepare_jinja2_environment(env=self.jinja_env, datasette=self)

349 350 -

self._register_renderers()

self._permission_checks = collections.deque(maxlen=200)

self._root_token = secrets.token_hex(32)

@@ -389,8 +387,16 @@ def urls(self):

389 387 returnUrls(self)

390 388 -

391 389 async def invoke_startup(self):

390 390 + # This must be called for Datasette to be in a usable state

391 391 + if self._startup_invoked:

392 392 + return

393 393 + for hook in pm.hook.prepare_jinja2_environment(

394 394 + env=self.jinja_env, datasette=self

395 395 +):

396 396 + await await_me_maybe(hook)

392 397 for hook in pm.hook.startup(datasette=self):

393 398 await await_me_maybe(hook)

399 399 + self._startup_invoked = True

394 400 -

395 401 def sign(self, value, namespace="default"):

396 402 return URLSafeSerializer(self._secret, namespace).dumps(value)

@@ -933,6 +939,8 @@ def _register_renderers(self):

Implementation

It should just do one thing
("thing" here is deliberately vague!)

Tests

Prove that the implementation works

Pass if the new implementation
is correct, fail otherwise

Every project should start with a test

assert $1 + 1 == 2$ is fine!

**Adding tests to an existing test suite is SO MUCH
less work than starting a new test suite from scratch**

Cookiecutter repo templates

`simonw/python-lib`

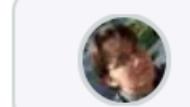
`simonw/click-app`

`simonw/datasette-plugin`

Create a new repository from python-lib-template-repository

The new repository will start with the same files and folders as [simonw/python-lib-template-repository](#).

Owner *



simonw ▾

Repository name *

my-new-python-library



Great repository names are short and memorable. Need inspiration? How about [verbose-waddle](#)?

Description (optional)

It does something really exciting

 **Public**

Anyone on the internet can see this repository. You choose who can commit.

 **Private**

You choose who can see and commit to this repository.

Include all branches

Copy all branches from simonw/python-lib-template-repository and not just main.

 You are creating a public repository in your personal account.

Create repository from template

Documentation

Same repository as the code!

Document changes that impact external developers

Update the docs in the same commit as the change

Catch missing documentation updates PR / code review

Bonus trick:
documentation unit tests

```
(datasette) datasette % pytest tests/test_docs.py -k plugin -v
=====
test session starts =====
platform darwin -- Python 3.10.3, pytest-7.1.3, pluggy-1.0.0 -- /Users/simon/.local/share/virtualenvs/datasette-AWNrQs95/bin/python
cachedir: .pytest_cache
SQLite: 3.37.0
rootdir: /Users/simon/Dropbox/Development/datasette, configfile: pytest.ini
plugins: anyio-3.6.1, xdist-2.5.0, forked-1.4.0, asyncio-0.19.0, timeout-2.1.0
asyncio: mode=strict
collected 81 items / 55 deselected / 26 selected

tests/test_docs.py::test_plugin_hooks_are_documented[actor_from_request] PASSED [ 3%]
tests/test_docs.py::test_plugin_hooks_are_documented[asgi_wrapper] PASSED [ 7%]
tests/test_docs.py::test_plugin_hooks_are_documented[canned_queries] PASSED [ 11%]
tests/test_docs.py::test_plugin_hooks_are_documented[database_actions] PASSED [ 15%]
tests/test_docs.py::test_plugin_hooks_are_documented[extra_body_script] PASSED [ 19%]
tests/test_docs.py::test_plugin_hooks_are_documented[extra_css_urls] PASSED [ 23%]
tests/test_docs.py::test_plugin_hooks_are_documented[extra_js_urls] PASSED [ 26%]
tests/test_docs.py::test_plugin_hooks_are_documented[extra_template_vars] PASSED [ 30%]
tests/test_docs.py::test_plugin_hooks_are_documented[filters_from_request] PASSED [ 34%]
tests/test_docs.py::test_plugin_hooks_are_documented[forbidden] PASSED [ 38%]
tests/test_docs.py::test_plugin_hooks_are_documented[get_metadata] PASSED [ 42%]
tests/test_docs.py::test_plugin_hooks_are_documented[handle_exception] PASSED [ 46%]
tests/test_docs.py::test_plugin_hooks_are_documented[menu_links] PASSED [ 50%]
tests/test_docs.py::test_plugin_hooks_are_documented[permission_allowed] PASSED [ 53%]
tests/test_docs.py::test_plugin_hooks_are_documented[prepare_connection] PASSED [ 57%]
tests/test_docs.py::test_plugin_hooks_are_documented[prepare_jinja2_environment] PASSED [ 61%]
tests/test_docs.py::test_plugin_hooks_are_documented[publish_subcommand] PASSED [ 65%]
tests/test_docs.py::test_plugin_hooks_are_documented[register_commands] PASSED [ 69%]
tests/test_docs.py::test_plugin_hooks_are_documented[register_facet_classes] PASSED [ 73%]
tests/test_docs.py::test_plugin_hooks_are_documented[register_magic_parameters] PASSED [ 76%]
tests/test_docs.py::test_plugin_hooks_are_documented[register_output_renderer] PASSED [ 80%]
tests/test_docs.py::test_plugin_hooks_are_documented[register_routes] PASSED [ 84%]
tests/test_docs.py::test_plugin_hooks_are_documented[render_cell] PASSED [ 88%]
tests/test_docs.py::test_plugin_hooks_are_documented[skip_csrf] PASSED [ 92%]
tests/test_docs.py::test_plugin_hooks_are_documented[startup] PASSED [ 96%]
tests/test_docs.py::test_plugin_hooks_are_documented[table_actions] PASSED [100%]

===== 26 passed, 55 deselected in 1.00s =====
(datasette) datasette %
```

```
41 @pytest.mark.parametrize(
42     "plugin", [name for name in dir(app.pm.hook) if not name.startswith("_")]
43 )
44 def test_plugin_hooks_are_documented(plugin, plugin_hooks_content):
45     headings = get_headings(plugin_hooks_content, "-")
46     assert plugin in headings
47     hook_caller = getattr(app.pm.hook, plugin)
48     arg_names = [a for a in hook_caller.spec.argnames if a != "__multicall__"]
49     # Check for plugin_name(arg1, arg2, arg3)
50     expected = f"{{plugin}}({', '.join(arg_names)})"
51     assert (
52         expected in plugin_hooks_content
53     ), f"Missing from plugin hook documentation: {expected}"
```

Everything links to
an issue thread

simonw / datasette Public

Code Issues Pull requests Discussions Actions Wiki Security Insights

prepare_jinja2_environment() hook should take datasette argument

#1809

Closed simonw opened this issue on Sep 14 · 11 comments

Red arrow pointing to the "11 comments" text.

11 comments

simonw commented on Sep 14 · edited

That plugin hook's current signature is:

```
datasette/datasette/hooks.py  
Lines 28 to 30 in 6104254  
28     @hookspec  
29     def prepare_jinja2_environment(env):  
30         """Modify Jinja2 template environment e.g. register custom template tags"""
```

As a result in the first alpha release of datasette-edit-templates I had to include this horrific hack:
https://github.com/simonw/datasette-edit-templates/blob/087f6a6cabc20020f2b0524f11aa3a7836320848/datasette_edit_templates/__init__.py#L72-L75

```
@hookimpl  
def prepare_jinja2_environment(env):  
    # TODO: This should ideally take datasette, but that's not an argument yet  
    datasette = inspect.currentframe().f_back.f_back.f_back.f_locals["self"]
```

simonw added enhancement plugins labels on Sep 14

simonw commented on Sep 14

It should also implement the await_me_maybe pattern so you can return an async function from it.

simonw commented on Sep 14

await_me_maybe might be tricky though, because the only place that hook is executed is here:

```
datasette/datasette/app.py  
Line 348 in 8430c3b  
348     pm.hook.prepare_jinja2_environment(env=self.jinja_env)
```

Assignees
No one assigned

Labels
enhancement plugins

Projects
None yet

Milestone
No milestone

Development
No branches or pull requests

1 participant

Red arrow pointing to the "11 comments" text.

What goes in an issue?

- Background: the reasons for the change
- State of play before-hand: embed existing code, link to existing docs
- Links to things! Documentation, inspiration, clues found on StackOverflow
- Code snippets illustrating potential designs and false-starts
- **Decisions.** What did you consider? What did you decide?
- Screenshots. Animated screenshots even better!
- When you close it: a link to the updated documentation and demo

Temporal documentation

It's timestamped and contextual

**You don't need to commit to keeping it up-to-date in the future
(but you can add more comments if you like)**

Issue driven development

Don't remember *anything*

You can back to a project in six months
and pick up right where you left off

Laboratory notebooks

en.wikipedia.org/wiki/Leonardo_da_Vinci



Figure out how to deploy Datasette to AWS Lambda using function URLs and Mangum #6

[New issue](#)[Closed](#)

simonw opened this issue 29 days ago · 65 comments



simonw commented 29 days ago · edited

Owner ...

Initial research and proof of concept for the `datasette-publish-lambda` plugin.simonw changed the title **One more try at Mangum-Datasette** Have another go at deploying Datasette to Lambda with Mangum 29 days ago

simonw commented 29 days ago

Owner ...

YouTube video from three months ago about deploying fast api on lambda <https://youtu.be/RGIM4JfsSk0>

Wow that video genuinely has everything I need to know - including how to make the zip file and how to deploy (via the console) and how to add a function URL

<https://github.com/pixegami/fastapi-tutorial>

```
from mangum import Mangum
app = FastAPI()
handler = Mangum(app)
```

```
pip install -t lib -r requirements.txt
```

```
(cd lib; zip ..../lambda_function.zip -r .)
```

```
zip lambda_function.zip -u main.py
zip lambda_function.zip -u books.json
```

Looks like you need cloudfront if you want a custom domain or subdomain though:

<https://medium.com/@walid.karray/configuring-a-custom-domain-for-aws-lambda-function-url-with-serverless-framework-c0d78abdc253>

(I bet cloudflare is easier)

Limit is still 50 MB (zipped, for direct upload) 250 MB (unzipped)

I want "datasette publish lambda"

I'm really tempted to have the `datasette-publish-lambda` GitHub repo build a release which is a zip file that has everything it needs - then the plugin itself downloads (and caches) the latest releases zip file, adds the SQLite DB and metadata and maybe some templates and plugins and static files to that zip file, adds any --install plugins to it and then ships the result

Assignees

No one assigned

Labels

research

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

1 participant



simonw/public-notes/issues

Tell people what
you did!

(It's so easy to skip this step)

Release notes (with dates)

Expand your definition of “done”
to include writing about what you did

Twitter threads

(Embed images + links + videos)

Get a blog

GUILT

is the enemy of projects

Avoid side projects with user accounts

If it has user accounts it's not a
side-project, it's an unpaid job

If your project is tested and documented,
you have nothing to feel guilty about

(That's what I tell myself anyway!)

Thank you

simonwillison.net

twitter.com/simonw

github.com/simonw