

# Building Protocol Libraries The Right Way

Hi!



# Hewlett Packard Enterprise

# Python HTTP

- Hyper
  - h2, hpack, priority, etc.
- Requests + urllib3
- PyCA (PyOpenSSL)

Why All Your Libraries  
Are Garbage, And All  
But One Of Mine Are  
Too

Python HTTP

Rocks

- Requests/urllib3/HTTPie
- gunicorn
- aiohttp
- Twisted
- Tornado

~0% Shared  
Code



# Code Re-Use



1/0

- httplib: state machine and parser mixed with I/O
- aiohttp/Tornado/Twisted: state machine and parser use concurrency primitives

Choice of I/O limits  
choice of libraries

So What?

Wasted Effort

Reduces  
Experimentation

# Duplicate Bugs



Limits

Optimisation

Avoiding  
This Problem

Don't Do I/O

# Alternative API

```
events = handle_data(in_bytes)
```

```
out_bytes = perform_action()
```

Not All-Purpose

# Build Wrappers

Why Bother?

===== test session starts =====

platform darwin -- Python 2.7.11, pytest-2.8.5, py-1.4.31, pluggy-0.3.1

rootdir: /Users/cory/Documents/Python/hyper-h2, inifile:

plugins: hypothesis-1.19.0, cov-2.2.0, xdist-1.13.1

gw0 [513] / gw1 [513] / gw2 [513] / gw3 [513]

scheduling tests via LoadScheduling

.....  
.....  
.....  
.....  
.....  
.....

----- coverage: platform darwin, python 2.7.11-final-0 -----

Name	Stmts	Miss	Branch	BrPart	Cover	Missing
------	-------	------	--------	--------	-------	---------

-----

h2/__init__.py	2	0	0	0	100%	
h2/connection.py	425	0	96	0	100%	
h2/errors.py	17	0	0	0	100%	
h2/events.py	95	0	4	0	100%	
h2/exceptions.py	41	0	0	0	100%	
h2/frame_buffer.py	55	0	18	0	100%	
h2/settings.py	74	0	20	0	100%	
h2/stream.py	286	0	46	0	100%	
h2/utilities.py	45	0	24	0	100%	

-----

TOTAL	1040	0	208	0	100%	
-------	------	---	-----	---	------	--

===== 513 passed in 6.69 seconds =====



I/O Is  
Application-  
Specific

# Build A Toolbox

# Example

- hyper-h2, a pure Python HTTP/2 protocol stack.
- <http://python-hyper.org/h2>

Help Us

Thanks!

Questions?