Python, Requests, & The Standard Library

Kenneth Reitz
Welcome.
Requests

HTTP for Humans

```python
>>> r = requests.get('https://api.github.com/user', auth=('user', 'pass'))
>>> r.status_code
200
>>> r.headers['content-type']
'application/json; charset=utf-8'
>>> r.encoding
'utf-8'
>>> r.text
u'{"type": "User"...'
>>> r.json
{u'private_gists': 419, u'total_private_repos': 77, ...}
```
Statement of Intent

• Establish the Requests Project’s current stance on Standard Library Inclusion.

• Use this forum up to a larger discussion on the purpose of the Standard Library.
Context
Requests Features

- Security is top-priority, as well as design.
- Interfaces optimized for automatic best-practices of SSL, Connection Pooling, Encoding, Headers, etc.
- Removes a significant amount of complexity around interacting with web services.
Requests Facts

- Requests has been downloaded ~42M times from The Cheeseshop.
- The most downloaded Python package ever.
- Standard Library inclusion is suggested often.
Requests Development

• 1 BDFL, 2 Core Contributors.

• Has been under feature-freeze for two years.

• Changes are still often made to the codebase, for various reasons.
Dependencies

- **Chardet** — Python 2/3 compatible character encoding detector.

- **Urllib3** — Python HTTP library with thread-safe connection pooling, file post support, sanity friendly, and more.

  (urllib3 receives continual improvements)
Critical infrastructure for the Python community.
Arguments For Inclusion

• Social responsibility — is it the “right thing” to do? This is the biggest driving force.

• Sustainability — having the project included in Python’s codebase may make it easier to request funding for core contributors.

• Chardet would make an excellent candidate for standard library inclusion.
Notable Observations

- Carefully curated and often updated CA Bundle — for TLS verification.
- HTTP specs and usage change significantly over time — we keep up with the changes.
- Security releases — 12 hour turnaround time.
Against Inclusion

• Requests’s biggest pitch is that it’s better than the standard library. If the standard library isn’t good enough, we can replace it.

• We would lose this ability, as well as our ability to make a quick release in the event of a security incident or spec change.
General Questions

• What is the Standard Library’s goals today?

• With the existence of —ensure—pip, is inclusion in the standard library necessary?

• —ensure—requests?
Critical infrastructure for the Python community.
Critical infrastructure for the Python community.

Too critical for standard library inclusion.