MiniPyP Documentation

Release 0.2.0a1

Ryan Garber

Mar 23, 2018

Contents:

1 Quickstart **3**
- 1.1 In Code . . . 3
- 1.2 As a Daemon . . . 3

2 Configuring MiniPyP **5**
- 2.1 Example Config . . . 5
- 2.2 Basics . . . 5
- 2.3 Sites . . . 6
- 2.4 Directories . . . 6
- 2.5 Paths . . . 6
- 2.6 Error Pages . . . 6

3 Dynamic Content **7**

4 Useful Methods **9**

5 Indices and tables **13**

Note: This documentation is a work in progress, but contains (hopefully) everything you need to get started.

CHAPTER 1

Quickstart

There's a few ways to start a MiniPyP server. The default config location is `/etc/minipyp/minipyp.conf` on Mac/Linux and `%APPDATA%\MiniPyP\minipyp.conf` on Windows.

1.1 In Code

If no config parameter is given, the default location will be used.

```
from minipyp import MiniPyP

MiniPyP(config='/path/to/config').start()
```

1.2 As a Daemon

Unlike above, if the config file does not exist, a default one will be generated.

```
sudo minipyp start [-c CONFIG_PATH]
```

CHAPTER 2

Configuring MiniPyP

MiniPyP uses YAML for its configuration.

When passing a dict as the config, convert all keys like so: “MIME Types” -> “mime_types”.

2.1 Example Config

```
Host: 0.0.0.0
Port: 80
Root: /var/www/default
Timeout: 15
Sites:
- URIs:
  - mysite.com
  - www.mysite.com
  Root: /var/www/mysite
  Paths:
    "/tunnel":
      Proxy: http://127.0.0.1:8080
Directories:
  "/":
    Public: false
  "/var/www":
    Public: true
```

2.2 Basics

These options are required to start the server. They are all you need to get started, but are very limited.

Name	Description
Host	Host to listen on, use "0.0.0.0" to allow all
Port	Port to listen on
Root	Default document root
Timeout	Keep-Alive timeout in seconds, default 15

2.3 Sites

You can configure different sites by hostname.

Name	Required	Description
URIs	Yes	List of hostnames this site applies to
Root	No	Document root for this site
Paths	No	Path options (see *Paths* below)

2.4 Directories

You can specify options for any directory on the server. The directory path can be a string or a regular expression.

Name	Description
Public	If false, trying to view this directory will give a 403
Static	If true, the server will not use catchall files or allow extensions to be omitted
Indexing	If false, the server will not give a list of files in the directory
Headers	Dictionary of headers to set
Clear Headers	If true, headers set by parent directories will be ignored
Don't Handle	List of filetypes to serve as-is without running them
Allow Options	List of options to allow the user to override with .minipyp.yml (not yet implemented)

2.5 Paths

You can specify options for any request path globally or inside a Site. The path can be a string or a regular expression.

Name	Description
Proxy	URL to serve as a reverse-proxy

2.6 Error Pages

You can specify custom error pages globally or inside a Directory. You should provide one of the following options.

Name	Description
HTML	Pre-made HTML to serve, {uri} will be replaced with the request URI
File	File to serve, e.g. '/var/www/errors/404.py'

CHAPTER 3

Dynamic Content

You may use Handlers to serve any type of dynamic content, but only a Python handler is provided by default.

Here's an example index.py that displays the client's IP and the current date.

```
from datetime.datetime import now

def render(server, request):
    return 'Your IP: ' + server.peer[0] + '\nCurrent date: ' + str(now())
```

Server object:

Attribute	Description
peer	Client info, e.g. ['127.0.0.1', 25565]

Request object:

Attribute	Description
protocol	Protocol version, e.g. "HTTP/1.1"
scheme	HTTP scheme, e.g. "https"
method	HTTP method, e.g. "POST"
headers	Request headers, e.g. {'User-Agent': 'Client/1.0'} (case-insensitive)
host	Request hostname, e.g. "mysite.com"
path	Request path, e.g. "/file.txt"
uri	Full request path, e.g. "/file.txt?query=string"
query_string	Query string, e.g. "query=string"
query	Parsed query string, e.g. {'query': ['string']}
body	Request body (ASCII)
post	Parsed request body, e.g. {'post': ['param']}
site	Site config, e.g. {'uris': ['mysite.com', 'www.mysite.com'], . . . }
root	Document root as given in config
file	Absolute path of requested file (document root + request path)

CHAPTER 4

Useful Methods

If you started your server in Python code, the following methods are available to you.

class minipyp.**MiniPyP**(*config*)

add_site(*site: dict*)
Add a site after initialization.

Parameters **site** – the same dict as in the configuration

get_directory(*dir: str*)
Get the options for any given directory, defaulting if no options were set.

Parameters **dir** – OS-specific directory path

Returns dict

get_error_page(*code: int*, *directory: str = None*)
Get the error page for any HTTP code.

Parameters

- **code** – HTTP status
- **directory** – OS-specific path to get custom error pages for

Returns dict

get_handler(*extension: str*)
Get a file handler by the extension.

Parameters **extension** – the file extension (e.g. 'py')

Returns Handler object or None

get_mime_type(*extension*)
Get the MIME type for any file extension.

Parameters **extension** – the file extension

Returns MIME type or None if one is not set

get_path (*path: str*, *site: dict = None*)
Get the options for any given path, defaulting if no options were set. If a site is given, it will be applied after any global options for the path.

Parameters

- **path** – URI (e.g. '/mypath')
- **site** – site object (see MiniPyP.get_site)

Returns dict

get_site (*host: str*)
Get a site by its hostname.

Parameters **host** – the hostname to look for

Returns dict or None

load_config (*config=None*)
Load new config values from the provided dict. If None, and a previously loaded file exists, it will be reloaded.

Parameters **config** – the new config

reload_config ()
Reloads the configuration file from disk.

set_directory (*dir: str*, *options: dict = None*)
Set the options for a directory. If options is None, all options will be removed. Otherwise, any options not provided will stay as-is in the configuration.

Parameters

- **dir** – OS-specific directory path
- **options** – the options to set

set_error_page (*code: int*, *page: dict*)
Set the error page for any HTTP code. Page should include either *html* for plain HTML or *file* to render a file instead.

Parameters

- **code** – HTTP status
- **page** – dict

set_handler (*extension: str*, *mime_type: str*, *handler: minipyp.minipyp.Handler*)
Add a file handler after initialization.

Parameters

- **extension** – the file extension (e.g. 'py')
- **mime_type** – the MIME type to serve with this file
- **handler** – a Handler subclass (see Handler docs)

set_mime_type (*mime_type*, **extensions*)
Set the MIME type of a file extension.

Parameters

- **mime_type** – MIME type (e.g. 'application/html'
- **extensions** – File extensions (e.g. 'py', 'pyc', . . .)

set_path(*path: str*, *options: dict = None*)

Set the options for a path. If options is None, all options will be removed. Otherwise, any options not provided will stay as-is in the configuration.

Parameters

- **path** – URI path
- **options** – the options to set

start()

Start the server.

stop()

Stop the server.

static test_config(*config: dict*, *part: str = None*)

Test the provided configuration, or a part of it.

Parameters

- **config** – dict of config
- **part** – part of the config to test (or None)

Raise ConfigError if invalid

write_config(*to: str = None*)

Writes the current configuration to the config file. If *to* is None (default), the config will be written to the real config file or a ConfigError will be thrown. If *to* is False, nothing will be written.

Parameters **to** – the file to write to

Returns YAML-encoded configuration

CHAPTER 5

Indices and tables

- genindex
- search

Index

A

add_site() (minipyp.MiniPyP method), 9

G

get_directory() (minipyp.MiniPyP method), 9
get_error_page() (minipyp.MiniPyP method), 9
get_handler() (minipyp.MiniPyP method), 9
get_mime_type() (minipyp.MiniPyP method), 9
get_path() (minipyp.MiniPyP method), 10
get_site() (minipyp.MiniPyP method), 10

L

load_config() (minipyp.MiniPyP method), 10

M

MiniPyP (class in minipyp), 9

R

reload_config() (minipyp.MiniPyP method), 10

S

set_directory() (minipyp.MiniPyP method), 10
set_error_page() (minipyp.MiniPyP method), 10
set_handler() (minipyp.MiniPyP method), 10
set_mime_type() (minipyp.MiniPyP method), 10
set_path() (minipyp.MiniPyP method), 10
start() (minipyp.MiniPyP method), 11
stop() (minipyp.MiniPyP method), 11

T

test_config() (minipyp.MiniPyP static method), 11

W

write_config() (minipyp.MiniPyP method), 11