Fujian

Release 1.1.1

September 02, 2016

Contents

1 Handler Classes 1

2 Helper Classes 3

3 Helper Functions 5

CHAPTER 1

Handler Classes

class `fujian.__main__.`**`FujianHandler`**(*application*, *request*, ***kwargs*)
Connect with clients via HTTP.

`get`()
Reply with an empty response body. Basically this is a ping request.

`post`()
Execute Python code submitted in the request body, and return results of the computation.

Note: The global `fujian_return` variable is set to a zero-length string before any code is executed, so it is guaranteed not to contain data from a previous request.

Response Body

If the response code is 200, it's a JSON object with three members: `stdout`, `stderr`, and `return`. If the response code is 400 (meaning there was an unhandled exception) the object also contains a `traceback` member.

All of these are strings. The `stdout` and `stderr` members are the contents of the corresponding stdio streams. The `return` member is the value stored in the global `fujian_return` variable at the end of the call. If present, `traceback` contains the traceback of the most recent unhandled exception.

`set_default_headers`()
Set the "Server" and "Access-Control-Allow-Origin" response headers.

class `fujian.__main__.`**`FujianWebSocketHandler`**(**args*, ***kwargs*)
Connect with clients via WebSocket.

`check_origin`(*origin*)
If supplied, Tornado uses this method to allow checking the Origin request header, to verify that the request is indeed coming from somewhere we are okay with. For us that means localhost, with or without HTTPS.

`is_open`()
Determine whether the WebSocket connection is currently open.

If there is no WebSocket currently open, any calls to `write_message()` will raise a `tornado.websocket.WebSocketClosedError`.

`on_close`()
Set the local flag to know the connection is closed.

`on_message`(*message*)
Execute Python code submitted in a WebSocket message.

This works much like *post()* except this method will not necessarily produce a response to the client. If the code writes to `stdout` or `stderr`, or sets the global `fujian_return` variable, a JSON response will be sent to the client in the same way as *post()*. If the code execution raises an unhandled exception, a `traceback` member will be included, in the same way as *post()*.

If the code does not raise an unhandled exception, write to `stdout` or `stderr`, or set the global `fujian_return` variable, no message will be sent to the client about the success or failure of code execution.

Furthermore, and quite unlike a connection to *FujianHandler*, messages may be sent to the client without first being requested. This is caused by any code that calls `write_message()` on the global `FUJIAN_WS` object installed by *FujianWebSocketHandler*.

open(***kwargs*)

Set the flag that avoids delaying small messages to save bandwidth. Since Fujian is intended only for use on `localhost`, any delay would be detrimental to the user experience, and morever there is no reason to save bandwidth.

Also set the local flag to know the connection is open.

CHAPTER 2

Helper Classes

class fujian.__main__.**StdoutHandler**

This is a replacement for `sys.stdout()` and `sys.stderr()` that collects its output for retrieval with `get()`.

This class supports the required `write()` method for an output stream, but no output is emitted under any circumstances except as returned from the `get()` method.

get()

Retrieve all data submitted to this `StdoutHandler` with `write()`.

Returns The buffered data.

Return type str

write(*write_this*)

Append `write_this` to this `StdoutHandler` instance's data buffer.

Parameters **write_this** (*str*) – Data to append.

CHAPTER 3

Helper Functions

fujian.__main__.**make_new_stdout**()
Make a new stdout and stderr, with the request's exec_globals, for a single request.

fujian.__main__.**execute_some_python**(*code*)
Execute some Python code in the "exec_globals" namespace.

Parameters **code** (*str*) – The Python code to execute.

Returns A dictionary with "stdout", "stderr", "return", and possibly "traceback" keys.

Return type dict

The dictionary returned contains the values written to stdout and stderr during the code execution. If a value is written to the global `fujian_return` variable, that is returned as the value of the "return" key. If the code raises an unhandled exception, the traceback appears is the value of the "traceback" key.

Note: All values in the dictionary are guaranteed to be the Unicode string type appropriate to the Python version in use.

---

fujian.__main__.**get_from_stdout**()
Get what was written to stdout, with the request's exec_globals, in this request.

fujian.__main__.**get_from_stderr**()
Get what was written to stderr, with the request's exec_globals, in this request.

fujian.__main__.**get_traceback**()
Get a traceback of the most recent exception raised in the subinterpreter.

fujian.__main__.**myprint**(*this*)
Prints "this" using the original stdout, even when it's been replaced. For use in debugging `fujian` itself.

C

check_origin() (fujian.__main__.FujianWebSocketHandler method), 1

E

execute_some_python() (in module fujian.__main__), 5

F

FujianHandler (class in fujian.__main__), 1
FujianWebSocketHandler (class in fujian.__main__), 1

G

get() (fujian.__main__.FujianHandler method), 1
get() (fujian.__main__.StdoutHandler method), 3
get_from_stderr() (in module fujian.__main__), 5
get_from_stdout() (in module fujian.__main__), 5
get_traceback() (in module fujian.__main__), 5

I

is_open() (fujian.__main__.FujianWebSocketHandler method), 1

M

make_new_stdout() (in module fujian.__main__), 5
myprint() (in module fujian.__main__), 5

O

on_close() (fujian.__main__.FujianWebSocketHandler method), 1
on_message() (fujian.__main__.FujianWebSocketHandler method), 1
open() (fujian.__main__.FujianWebSocketHandler method), 2

P

post() (fujian.__main__.FujianHandler method), 1

S

set_default_headers() (fujian.__main__.FujianHandler method), 1
StdoutHandler (class in fujian.__main__), 3

W

write() (fujian.__main__.StdoutHandler method), 3