

HTTP Server

Description

In this problem, you will simulate an HTTP server that handles client requests and responds with JSON content. The server will return different messages based on the requested URL and status codes. The server uses zlib compression for the responses. Your task is to implement the server to compress the JSON content before sending it to the client.

The server should respond with the following dictionary structure and you need to `json.dumps` to convert it to a JSON data:

```
{
    "status": status,
    "message": content
}
```

- Replace `status` with the appropriate status code.
- Replace `content` with the appropriate message as shown in the unit test.

Task

- Implement the server to handle requests to `/index.html` and `/hello.html`
- Return JSON responses with status codes 200, 500, and 404.
- Compress the JSON responses using `zlib` before sending them to the client.

Output (with unit test)

Testing `create_server` ...

`bind` called with: `call(('localhost', 8080))`

`listen` called with: `call(5)`

Testing `get_content` ...

test attribute passed: Hello world! is in `{"status": 200, "message": "Hello world!"}`

test attribute passed: 404 Not found is in `{"status": 404, "message": "404 Not found"}`

test attribute passed: 500 Internal Server Error is in `{"status": 500, "message": "500 Internal Server Error"}`

Testing `get_header` ...

request header: `['GET /index.html HTTP/1.1', 'Host: localhost', '', '']`

test attribute passed: `/index.html` is equal to `/index.html`

Testing `process_requests` ...

```
accept called with: call()
test attribute passed: call(('localhost', 8080)) is equal to
call(('localhost', 8080))
test attribute passed: 5 is equal to 5
test attribute listen passed: True is True
test attribute accept passed: True is True
test attribute close passed: True is True
```