

post_200_monster

The screenshot shows a REST client interface with a POST request to `http://127.0.0.1:5000/server/character`. The request body is a JSON object representing a monster:

```
1 {
2   "type": "monster",
3   "monster_ai_difficulty": "hard",
4   "monster_type": "dragon"
5 }
```

The response status is 200 OK, and the body is displayed as `1 1`.

```
18 try:
19     if content["type"] == "player":
20         character = Player(content['player_level'], content['
21         server.add_character(character)
22     else:
23         character = Monster(content['monster_type'],
24                             content['monster_ai_difficulty'])
25         server.add_character(character)
26
27     response = app.response_class(
28         status=200,
29         response=json.dumps(
30             server.get_assigned_id()
31         ), mimetype='application/json'
32     )
33 except ValueError as e:
34     response = app.response_class(
35         response=str(e),
36         status=400
```

The terminal output shows the server running on `http://127.0.0.1:5000/` and receiving a POST request to `/server/character` with a 200 status.

post_200_player

The screenshot shows a REST client interface with a POST request to `http://127.0.0.1:5000/server/character`. The request body is a JSON object representing a player:

```
1 {
2   "type": "player",
3   "player_level": 4,
4   "job": "assassin"
5 }
```

The response status is 200 OK, and the body is displayed as `1 2`.

```
18 try:
19     if content["type"] == "player":
20         character = Player(content['player_level'], content['
21         server.add_character(character)
22     else:
23         character = Monster(content['monster_type'],
24                             content['monster_ai_difficulty'])
25         server.add_character(character)
26
27     response = app.response_class(
28         status=200,
29         response=json.dumps(
30             server.get_assigned_id()
31         ), mimetype='application/json'
32     )
33 except ValueError as e:
34     response = app.response_class(
35         response=str(e),
36         status=400
```

The terminal output shows the server running on `http://127.0.0.1:5000/` and receiving a POST request to `/server/character` with a 200 status.

post_400

POST http://127.0.0.1:5000/server/character

Body

```
1 {
2   "type": "player",
3   "player_level": 4,
4   "job": "assassin"
5 }
```

Status: 400 BAD REQUEST Time: 30ms Size: 34

Body

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
2 <title>400 Bad Request</title>
3 <h1>Bad Request</h1>
4 <n>The browser (or proxy) sent a request that this server could not understand.</n>
```

```
13
14 app.route('/server/character', methods=['POST'])
15 def add_character():
16     """ Adds a character to the Server """
17     content = request.json
18     try:
19         if content["type"] == "player":
20             character = Player(content['player_level'], content[''])
21             server.add_character(character)
22         else:
23             character = Monster(content['monster_type'],
24                                 content['monster_ai_difficulty'])
25             server.add_character(character)
26
27         response = app.response_class(
28             status=200
29         )
30
31     except ValueError as e:
```

put_200

PUT http://127.0.0.1:5000/server/character/2

Body

```
1 {
2   "player_level": 3,
3   "job": "warrior"
4 }
```

Status: 200 OK Time: 20ms Size: 153

Body

```
1
```

```
13
14 app.route('/server/character', methods=['POST'])
15 def add_character():
16     """ Adds a character to the Server """
17     content = request.json
18     try:
19         if content["type"] == "player":
20             character = Player(content['player_level'], content[''])
21             server.add_character(character)
22         else:
23             character = Monster(content['monster_type'],
24                                 content['monster_ai_difficulty'])
25             server.add_character(character)
26
27         response = app.response_class(
28             status=200
29         )
30
31     except ValueError as e:
```

put_404

The screenshot shows a REST client interface with a PUT request to `http://127.0.0.1:5000/server/character/5`. The request body is a JSON object: `{ "player_level": 3, "job": "warrior" }`. The status is `404 NOT FOUND`. The response body is `1 Character does not exist`.

The Python code snippet shows the `add_character` function, which adds a character to the server based on the request type (player or monster).

```
app.route('/server/character', methods=['POST'])
def add_character():
    """ Adds a character to the Server """
    content = request.json
    try:
        if content["type"] == "player":
            character = Player(content['player_level'], content['job'])
            server.add_character(character)
        else:
            character = Monster(content['monster_type'], content['monster_ai_difficulty'])
            server.add_character(character)
        response = app.response_class(
            status=200
        )
    except ValueError as e:
```

delete_200

The screenshot shows a REST client interface with a DELETE request to `http://127.0.0.1:5000/server/characters/1`. The status is `200 OK`. The response body is empty.

The Python code snippet shows the `delete_character` function, which removes a character from the server based on the request type (player or monster).

```
try:
    if content["type"] == "player":
        character = Player(content['player_level'], content['job'])
        server.add_character(character)
    else:
        character = Monster(content['monster_type'], content['monster_ai_difficulty'])
        server.add_character(character)
    response = app.response_class(
        status=200,
        response=isjson.dumps(
            ser Response sined_id()
        ), mimetype='application/json'
    )
except ValueError as e:
    response = app.response_class(
        response=str(e),
        status=400
    )
```

delete_404

POST http://127.0.0.1:5000... POST http://127.0.0.1:5000... DEL http://127.0.0.1:5000... + ... No Environment

▶ http://127.0.0.1:5000/server/characters/1

DELETE http://127.0.0.1:5000/server/characters/5

Params Authorization Headers (8) Body Pre-request Script Tests Settings

Query Params

Body Cookies Headers (4) Test Results Status: 404 NOT FOUND Time: 23ms Size: 18

1 Character does not exist.

```
18 try:
19     if content["type"] == "player":
20         character = Player(content['player_level'], content['
21         server.add_character(character)
22     else:
23         character = Monster(content['monster_type'],
24                             content['monster_ai_difficulty'])
25         server.add_character(character)
26
27     response = app.response_class(
28         status=200,
29         response=json.dumps(
30             server.get_assigned_id()
31         ), mimetype='application/json'
32     )
33 except ValueError as e:
34     response = app.response_class(
35         response=str(e),
36         status=400
```

get_200

POST http://127.0.0.1:5000... POST http://127.0.0.1:5000... GET http://127.0.0.1:5000... + ... No Environment

▶ http://127.0.0.1:5000/server/1

GET http://127.0.0.1:5000/server/characters/2

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Query Params

Body Cookies Headers (4) Test Results Status: 200 OK Time: 13ms Size: 275

1 {

2 "id": 2,

3 "health": 92,

4 "damage": 39,

5 "position": [

6 0,

7 0

8],

9 "alive": true,

10 "player_level": 4,

11 "job": "assassin",

12 "type": "player"

13 }

```
18 try:
19     if content["type"] == "player":
20         character = Player(content['player_level'], content['
21         server.add_character(character)
22     else:
23         character = Monster(content['monster_type'],
24                             content['monster_ai_difficulty'])
25         server.add_character(character)
26
27     response = app.response_class(
28         status=200,
29         response=json.dumps(
30             server.get_assigned_id()
31         ), mimetype='application/json'
32     )
33 except ValueError as e:
34     response = app.response_class(
35         response=str(e),
36         status=400
```

get_404

POST http://127.0.0.1:5000... POST http://127.0.0.1:5000... GET http://127.0.0.1:5000... + ... No Environment

▶ http://127.0.0.1:5000/server/1

GET http://127.0.0.1:5000/server/characters/5

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (4) Test Results Status: 404 NOT FOUND Time: 25ms Size: 185

1 Character does not exist

```
18 try:
19     if content["type"] == "player":
20         character = Player(content['player_level'], content['
21             server.add_character(character)
22     else:
23         character = Monster(content['monster_type'],
24             content['monster_ai_difficulty'])
25         server.add_character(character)
26
27     response = app.response_class(
28         status=200,
29         response=json.dumps(
30             server.get_assigned_id()
31         ), mimetype='application/json'
32     )
33 except ValueError as e:
34     response = app.response_class(
35         response=str(e),
36         status=400
```

QB@QB-MacBook Assignment2 % python3 server_api.py
* Serving Flask app "server_api" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - [17/Nov/2019 13:21:28] "POST /server/character HTTP/1.1" 200 -
127.0.0.1 - [17/Nov/2019 13:21:39] "POST /server/character HTTP/1.1" 200 -
127.0.0.1 - [17/Nov/2019 13:21:56] "POST /server/character HTTP/1.1" 400 -
127.0.0.1 - [17/Nov/2019 13:24:16] "DELETE /server/characters/1 HTTP/1.1" 200 -
127.0.0.1 - [17/Nov/2019 13:24:40] "DELETE /server/characters/5 HTTP/1.1" 404 -
127.0.0.1 - [17/Nov/2019 13:26:32] "GET /server/characters/2 HTTP/1.1" 200 -
127.0.0.1 - [17/Nov/2019 13:26:46] "GET /server/characters/5 HTTP/1.1" 404 -

get_all_200

POST http://127... POST http://127... GET http://127... GET http://127.0... X + ... No Environment

▶ http://127.0.0.1:5000/server/characters/all

GET http://127.0.0.1:5000/server/characters/all

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (4) Test Results Status: 200 OK Time: 17ms Size: 423

1 [
2 "The player (id: 2) is level 4 assassin with 92 health and 39 damage, Position: X = 0 Y = 0
3 "The monster (id: 3) is hard dragon with 150 health and 10 damage, Position: X = 0 Y = 0
4 "The monster (id: 4) is hard dragon with 150 health and 10 damage, Position: X = 0 Y = 0
5]

```
18 try:
19     if content["type"] == "player":
20         character = Player(content['player_level'], content['
21             server.add_character(character)
22     else:
23         character = Monster(content['monster_type'],
24             content['monster_ai_difficulty'])
25         server.add_character(character)
26
27     response = app.response_class(
28         status=200,
29         response=json.dumps(
30             server.get_assigned_id()
31         ), mimetype='application/json'
32     )
33 except ValueError as e:
34     response = app.response_class(
35         response=str(e),
36         status=400
```

QB@QB-MacBook Assignment2 % python3 server_api.py
* Serving Flask app "server_api" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - [17/Nov/2019 13:21:28] "POST /server/character HTTP/1.1" 200 -
127.0.0.1 - [17/Nov/2019 13:21:39] "POST /server/character HTTP/1.1" 200 -
127.0.0.1 - [17/Nov/2019 13:21:56] "POST /server/character HTTP/1.1" 400 -
127.0.0.1 - [17/Nov/2019 13:24:16] "DELETE /server/characters/1 HTTP/1.1" 200 -
127.0.0.1 - [17/Nov/2019 13:24:40] "DELETE /server/characters/5 HTTP/1.1" 404 -
127.0.0.1 - [17/Nov/2019 13:26:32] "GET /server/characters/2 HTTP/1.1" 200 -
127.0.0.1 - [17/Nov/2019 13:26:46] "GET /server/characters/5 HTTP/1.1" 404 -
127.0.0.1 - [17/Nov/2019 13:28:22] "GET /server/characters/all HTTP/1.1" 200 -
127.0.0.1 - [17/Nov/2019 13:28:34] "POST /server/character HTTP/1.1" 200 -
127.0.0.1 - [17/Nov/2019 13:28:36] "POST /server/character HTTP/1.1" 200 -
127.0.0.1 - [17/Nov/2019 13:28:38] "GET /server/characters/all HTTP/1.1" 200 -

get_all_200_empty

POST http://... POST http://... GET http://... GET http://... X DEL http://... + ... No Environment

http://127.0.0.1:5000/server/characters/all

GET http://127.0.0.1:5000/server/characters/all

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (4) Test Results Status: 200 OK Time: 15ms Size: 0

Pretty Raw Preview Visualize BETA JSON

```
1 []
```

```
18 try:
19     if content["type"] == "player":
20         character = Player(content['player_level'], content['
21         server.add_character(character)
22     else:
23         character = Monster(content['monster_type'],
24         content['monster_ai_difficulty'])
25         server.add_character(character)
26
27 response = app.response_class(
28     status=200,
29     response=json.dumps(CharacterManager.get_assigned_id() -> int
30     server.get_assigned_id()
31     ), mimetype='application/json'
32 )
33 except ValueError as e:
34     response = app.response_class(
35         response=str(e),
36         status=400
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: Python

QB008-MacBook Assignment2 % python3 server_api.py

- * Serving Flask app "server_api" (lazy loading)
- * Environment: production
- WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
- * Debug mode: off
- * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

```
127.0.0.1 - - [17/Nov/2019 13:21:28] "POST /server/character HTTP/1.1" 200 -
127.0.0.1 - - [17/Nov/2019 13:21:39] "POST /server/character HTTP/1.1" 200 -
127.0.0.1 - - [17/Nov/2019 13:21:56] "POST /server/character HTTP/1.1" 400 -
127.0.0.1 - - [17/Nov/2019 13:24:16] "DELETE /server/characters/1 HTTP/1.1" 200 -
127.0.0.1 - - [17/Nov/2019 13:24:40] "DELETE /server/characters/5 HTTP/1.1" 404 -
127.0.0.1 - - [17/Nov/2019 13:26:32] "GET /server/characters/2 HTTP/1.1" 200 -
127.0.0.1 - - [17/Nov/2019 13:26:46] "GET /server/characters/5 HTTP/1.1" 404 -
127.0.0.1 - - [17/Nov/2019 13:28:22] "GET /server/characters/all HTTP/1.1" 200 -
127.0.0.1 - - [17/Nov/2019 13:28:34] "POST /server/character HTTP/1.1" 200 -
127.0.0.1 - - [17/Nov/2019 13:28:36] "POST /server/character HTTP/1.1" 200 -
127.0.0.1 - - [17/Nov/2019 13:28:38] "GET /server/characters/all HTTP/1.1" 200 -
127.0.0.1 - - [17/Nov/2019 13:30:11] "DELETE /server/characters/2 HTTP/1.1" 200 -
127.0.0.1 - - [17/Nov/2019 13:30:14] "DELETE /server/characters/3 HTTP/1.1" 200 -
127.0.0.1 - - [17/Nov/2019 13:30:16] "DELETE /server/characters/4 HTTP/1.1" 200 -
127.0.0.1 - - [17/Nov/2019 13:30:22] "DELETE /server/characters/all HTTP/1.1" 200 -
```

get_all_type_200

POST http://127.0.0.1:50... POST http://127.0.0.1:50... GET http://127.0.0.1:500... X + ... No Environment

http://127.0.0.1:5000/server/characters/all/player

GET http://127.0.0.1:5000/server/characters/all/player

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (4) Test Results Status: 200 OK Time: 15ms Size: 6

Pretty Raw Preview Visualize BETA JSON

```
1 [
2     "The player (id: 1) is level 1 knight with 100 health and 20 damage, Position: X = 0 Y = 0",
3     "The player (id: 2) is level 1 knight with 100 health and 20 damage, Position: X = 0 Y = 0",
4     "The player (id: 3) is level 4 assassin with 92 health and 39 damage, Position: X = 0 Y = 0",
5     "The player (id: 5) is level 4 assassin with 92 health and 39 damage, Position: X = 0 Y = 0"
6 ]
```

```
76 def get_all_by_type(character_type):
77     """ Gets all existing characters from the Server by type """
78     try:
79         response = app.response_class(
80             status=200,
81             response=json.dumps(
82                 server.get_character_details_by_type(character_t
83             ),
84             mimetype='application/json'
85         )
86         return response
87     except ValueError as e:
88         response = app.response_class(response=str(e), status=40
89         return response
90
91 @app.route('/server/characters/all', methods=['GET'])
92 def get_all():
93     """ Gets all existing characters from the Server """
94     try:
95         response = app.response_class(
96             status=200,
97             response=json.dumps(
98                 server.get_all_character_details()
99             ),
100             mimetype='application/json'
101         )
102         return response
```

PROBLEMS COMMENTS TERMINAL ... 2: Python Debug Console

QB008-MacBook Assignment2 % python3 server_api.py

- * Serving Flask app "server_api" (lazy loading)
- * Environment: production
- WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
- * Debug mode: off
- * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

```
127.0.0.1 - - [17/Nov/2019 17:41:29] "POST /server/character HTTP/1.1" 200 -
127.0.0.1 - - [17/Nov/2019 17:41:55] "POST /server/character HTTP/1.1" 200 -
127.0.0.1 - - [17/Nov/2019 17:45:26] "GET /server/characters/all/player HTTP/1.1" 200 -
```

get_all_type_400

POST http://127.0.0.1:5000/... POST http://127.0.0.1:5000/... GET http://127.0.0.1:5000/... + ... No Environment

▶ http://127.0.0.1:5000/server/characters/all/player

GET http://127.0.0.1:5000/server/characters/all/human

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (4) Test Results

Status: 400 BAD REQUEST Time: 14ms Size: 185

Pretty Raw Preview Visualize BETA HTML

1 Invalid character type

```
77 """ Gets all existing characters from the Server by type """
78 try:
79     response = app.response_class(
80         status=200,
81         response=json.dumps(
82             server.get_character_details_by_type(character_type)
83         ),
84         mimetype='application/json'
85     )
86     return response
87 except ValueError as e:
88     response = app.response_class(response=str(e), status=400)
89     return response
90
91 @app.route('/server/characters/all', methods=['GET'])
92 def get_all():
93     """ Gets all existing characters from the Server """
94     try:
95         response = app.response_class(
96             status=200,
97             response=json.dumps(
98                 server.get_all_character_details()
99             ),
100             mimetype='application/json'
101         )
102         return response
103 except ValueError as e:
104     response = app.response_class(response=str(e), status=400)
105     return response
```

QB008-MacBook Assignment2 % python3 server_api.py
* Serving Flask app "server_api" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [17/Nov/2019 13:34:48] "POST /server/character HTTP/1.1" 200 -
127.0.0.1 - - [17/Nov/2019 13:34:55] "GET /server/characters/all/monster HTTP/1.1" 200 -
127.0.0.1 - - [17/Nov/2019 13:35:00] "GET /server/characters/all/human HTTP/1.1" 400 -

server_stats_200

POST http://127.0.0.1:5000/... POST http://127.0.0.1:5000/... GET http://127.0.0.1:5000/... X + ... No Environment

▶ http://127.0.0.1:5000/server/serverstats

GET http://127.0.0.1:5000/server/serverstats

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (4) Test Results

Status: 200 OK Time: 11ms Size: 200

Pretty Raw Preview Visualize BETA JSON

1 {
2 "total_num_characters": 5,
3 "num_monsters": 1,
4 "num_players": 4,
5 "avg_player_level": 2,
6 "avg_monster_ai_difficulty": "hard"
7 }

```
76 def get_all_by_type(character_type):
77     """ Gets all existing characters from the Server by type """
78     try:
79         response = app.response_class(
80             status=200,
81             response=json.dumps(
82                 server.get_character_details_by_type(character_type)
83             ),
84             mimetype='application/json'
85         )
86         return response
87 except ValueError as e:
88     response = app.response_class(response=str(e), status=400)
89     return response
90
91 @route_decorator()
92 @app.route('/server/characters/all', methods=['GET'])
93 def get_all():
94     """ Gets all existing characters from the Server """
95     try:
96         response = app.response_class(
97             status=200,
98             response=json.dumps(
99                 server.get_all_character_details()
100             ),
101             mimetype='application/json'
102         )
103         return response
104 except ValueError as e:
105     response = app.response_class(response=str(e), status=400)
106     return response
```

QB008-MacBook Assignment2 % python3 server_api.py
* Serving Flask app "server_api" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [17/Nov/2019 17:41:29] "POST /server/character HTTP/1.1" 200 -
127.0.0.1 - - [17/Nov/2019 17:41:55] "POST /server/character HTTP/1.1" 200 -
127.0.0.1 - - [17/Nov/2019 17:45:26] "POST /server/characters/all/player HTTP/1.1" 200 -
127.0.0.1 - - [17/Nov/2019 17:47:11] "GET /server/serverstats HTTP/1.1" 200 -

server_no_stats_200

POST http://127.0.0.1:5000/... POST http://127.0.0.1:5000/... GET http://127.0.0.1:5000/... X + ... No Environment

▶ http://127.0.0.1:5000/server/serverstats

GET http://127.0.0.1:5000/server/serverstats

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (4) Test Results Status: 200 OK Time: 22ms Size: 288

Pretty Raw Preview Visualize BETA JSON

```
1 {
2   "total_num_characters": 0,
3   "num_monsters": 0,
4   "num_players": 0,
5   "avg_player_level": 0,
6   "avg_monster_ai_difficulty": "not available"
7 }
```

```
77 # Gets all existing characters from the Server by type
78 try:
79     response = app.response_class(
80         status=200,
81         response=json.dumps(
82             server.get_character_details_by_type(character_type)
83         ),
84         mimetype='application/json'
85     )
86     return response
87 except ValueError as e:
88     response = app.response_class([response=str(e), status=400])
89     return response
90
91
92 app.route('/server/characters/all', methods=['GET'])
93 if get_all():
94     """ Gets all existing characters from the Server """
95     try:
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: Python

```
QB@QB-MacBook Assignment2 % python3 server_api.py
* Serving Flask app "server_api" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [17/Nov/2019 13:34:48] "POST /server/character HTTP/1.1" 200 -
127.0.0.1 - - [17/Nov/2019 13:34:55] "GET /server/characters/all/monster HTTP/1.1" 200 -
127.0.0.1 - - [17/Nov/2019 13:35:00] "GET /server/characters/all/human HTTP/1.1" 400 -
127.0.0.1 - - [17/Nov/2019 13:35:48] "POST /server/character HTTP/1.1" 200 -
127.0.0.1 - - [17/Nov/2019 13:35:48] "POST /server/character HTTP/1.1" 200 -
127.0.0.1 - - [17/Nov/2019 13:36:48] "GET /server/serverstats HTTP/1.1" 200 -
^C
QB@QB-MacBook Assignment2 % python3 server_api.py
* Serving Flask app "server_api" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [17/Nov/2019 13:37:27] "GET /server/serverstats HTTP/1.1" 200 -
```