

DayTimeWsgi/runserver.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # runserver.py
5: # Author: Bob Dondero
6: #-----
7:
8: import sys
9: import wsgiref.simple_server
10: import daytime
11:
12: def main():
13:
14:     if len(sys.argv) != 2:
15:         print(f'usage: {sys.argv[0]} port', file=sys.stderr)
16:         sys.exit(1)
17:
18:     try:
19:         port = int(sys.argv[1])
20:     except Exception:
21:         print(f'{sys.argv[0]}: Port must be an integer.',
22:               file=sys.stderr)
23:         sys.exit(1)
24:
25:     try:
26:         httpd = wsgiref.simple_server.make_server(
27:             '0.0.0.0', port, daytime.app)
28:         print('Listening on port ' + str(port))
29:         httpd.serve_forever()
30:     except Exception as ex:
31:         print(f'{sys.argv[0]}: {ex}', file=sys.stderr)
32:         sys.exit(1)
33:
34: if __name__ == '__main__':
35:     main()

```

DayTimeWsgi/daytime.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # daytime.py
5: # Author: Bob Dondero
6: #-----
7:
8: import time
9:
10: #-----
11:
12: def index(envIRON, start_response):
13:
14:     daytime = time.asctime(time.localtime())
15:
16:     html_code = f'''
17:         <!DOCTYPE html>
18:         <html>
19:             <head>
20:                 <title>DayTime</title>
21:             </head>
22:             <body>
23:                 <p>{daytime}</p>
24:             </body>
25:         </html>
26:     '''
27:
28:     content_header = ('content-type', 'text/html; charset=utf-8')
29:     headers = [content_header]
30:     start_response('200 OK', headers)
31:     return [html_code.encode('utf-8')]
32:
33: #-----
34:
35: def not_found(envIRON, start_response):
36:
37:     html_code = '''
38:         <!DOCTYPE html>
39:         <html>
40:             <head>
41:                 <title>404 Not Found</title>
42:             </head>
43:             <body>
44:                 <h1>Not Found</h1>
45:                 <p>The requested URL was not found on the server.
46:                 If you entered the URL manually please check your
47:                 spelling and try again.</p>
48:             </body>
49:         </html>
50:     '''
51:
52:     content_header = ('content-type', 'text/html; charset=utf-8')
53:     headers = [content_header]
54:     start_response('404 Not Found', headers)
55:     return [html_code.encode('utf-8')]
56:
57: #-----
58:
59: def app(envIRON, start_response):
60:
61:     path = envIRON.get('PATH_INFO', '').strip('/')
62:
63:     if path in ('', 'index'):
64:         return index(envIRON, start_response)
65:     return not_found(envIRON, start_response)

```

EchoWsgiGet/runserver.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # runserver.py
5: # Author: Bob Dondero
6: #-----
7:
8: import sys
9: import wsgiref.simple_server
10: import echo
11:
12: def main():
13:
14:     if len(sys.argv) != 2:
15:         print('Usage: ' + sys.argv[0] + ' port', file=sys.stderr)
16:         sys.exit(1)
17:
18:     try:
19:         port = int(sys.argv[1])
20:     except Exception:
21:         print('Port must be an integer.', file=sys.stderr)
22:         sys.exit(1)
23:
24:     try:
25:         httpd = wsgiref.simple_server.make_server(
26:             '0.0.0.0', port, echo.app)
27:         print('Listening on port ' + str(port))
28:         httpd.serve_forever()
29:     except Exception as ex:
30:         print(ex, file=sys.stderr)
31:         sys.exit(1)
32:
33: if __name__ == '__main__':
34:     main()

```

blank (Page 1 of 1)

1: This page is intentionally blank.

EchoWsgiGet/echo.py (Page 1 of 2)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # echo.py
5: # Author: Bob Dondero
6: #-----
7:
8: import html
9: import urllib.parse
10:
11: #-----
12:
13: def index(envIRON, start_response):
14:
15:     html_code = '''
16:         <!DOCTYPE html>
17:         <html>
18:             <body>
19:                 <form action="/echoresult" method="get">
20:                     Enter a line to be echoed:<br>
21:                     <input type="text" name="line"
22:                         value="Enter the line here">
23:                     <br>
24:                     <input type="submit">
25:                 </form>
26:             </body>
27:         </html>
28:     '''
29:
30:     content_header = ('content-type', 'text/html; charset=utf-8')
31:     headers = [content_header]
32:     start_response('200 OK', headers)
33:     return [html_code.encode('utf-8')]
34:
35: #-----
36:
37: def parse_args_get(envIRON, key):
38:
39:     args_str = envIRON.get('QUERY_STRING', None)
40:     if args_str is None:
41:         return None
42:     args = urllib.parse.parse_qs(args_str)
43:     values = args.get(key, None)
44:     if values is None or len(values) == 0:
45:         return None
46:     return values[0]
47:
48: #-----
49:
50: def echo_result(envIRON, start_response):
51:
52:     line_to_be_echoed = parse_args_get(envIRON, 'line')
53:     if line_to_be_echoed is None:
54:         line_to_be_echoed = '(You did not enter a line.)'
55:
56:     html_code = f'''
57:         <!DOCTYPE html>
58:         <html>
59:             <body>
60:                 <p>This is the line that you entered:</p>
61:                 <p>{html.escape(line_to_be_echoed)}</p>
62:             </body>
63:         </html>
64:     '''
65:

```

EchoWsgiGet/echo.py (Page 2 of 2)

```

66:     content_header = ('content-type', 'text/html; charset=utf-8')
67:     headers = [content_header]
68:     start_response('200 OK', headers)
69:     return [html_code.encode('utf-8')]
70:
71: #-----
72:
73: def not_found(envIRON, start_response):
74:
75:     html_code = '''
76:         <!DOCTYPE html>
77:         <html>
78:             <head>
79:                 <title>404 Not Found</title>
80:             </head>
81:             <body>
82:                 <h1>Not Found</h1>
83:                 <p>The requested URL was not found on the server.
84:                 If you entered the URL manually please check your
85:                 spelling and try again.</p>
86:             </body>
87:         </html>
88:     '''
89:
90:     content_header = ('content-type', 'text/html; charset=utf-8')
91:     headers = [content_header]
92:     start_response('404 Not Found', headers)
93:     return [html_code.encode('utf-8')]
94:
95: #-----
96:
97: def app(envIRON, start_response):
98:
99:     path = envIRON.get('PATH_INFO', '').strip('/')
100:
101:     if path in ('', 'index'):
102:         return index(envIRON, start_response)
103:     if path == 'echoresult':
104:         return echo_result(envIRON, start_response)
105:
106:     return not_found(envIRON, start_response)

```