PennyWsgi/runserver.py (Page 1 of 1)

```
#!/usr/bin/env python

#-------------------------------------------------------------------------
# runserver.py
# Author: Bob Dondero
#-------------------------------------------------------------------------

import sys
import wsgiref.simple_server
import penny

def main():

    if len(sys.argv) != 2:
        print('Usage: ' + sys.argv[0] + ' port', file=sys.stderr)
        sys.exit(1)

    try:
        port = int(sys.argv[1])
    except Exception:
        print('Port must be an integer.', file=sys.stderr)
        sys.exit(1)

    try:
        httpd = wsgiref.simple_server.make_server(
            '0.0.0.0', port, penny.app)
        print('Listening on port ' + str(port))
        httpd.serve_forever()
    except Exception as ex:
        print(ex, file=sys.stderr)
        sys.exit(1)

if __name__ == '__main__':
    main()
```

PennyWsgi/common.py (Page 1 of 1)

```
#!/usr/bin/env python

#-------------------------------------------------------------------------
# common.py
# Author: Bob Dondero
#-------------------------------------------------------------------------

import time

#-------------------------------------------------------------------------

def get_header():
    html_str = ''
    if time.strftime('%p') == "AM":
        greeting = 'morning'
    else:
        greeting = 'afternoon'
    html_str += '<hr>'
    html_str += 'Good ' + greeting + ' and welcome to '
    html_str += '<strong>Penny.com</strong>'
    html_str += '<hr>'
    return html_str

#-------------------------------------------------------------------------

def get_footer():
    html_str = ''
    html_str += '<hr>'
    html_str += 'Today is ' + time.asctime(time.localtime()) + '.<br>'
    html_str += 'Created by '
    html_str += '<a href="https://www.cs.princeton.edu/~rdondero">'
    html_str += 'Bob Dondero</a>'
    html_str += '<hr>'
    return html_str

#-------------------------------------------------------------------------

# For testing:

def _test():
    print(get_header())
    print()
    print()
    print(get_footer())

if __name__ == '__main__':
    _test()
```

PennyWsgi/penny.py (Page 1 of 3)

```
#!/usr/bin/env python

#------------------------------------------------------------------------
# penny.py
# Author: Bob Dondero
#------------------------------------------------------------------------

import http.cookies
import html
import common
import parseargs
import database

def index(environ, start_response):

    html_code = ''
    html_code += '<!DOCTYPE html>'
    html_code += '<html>'
    html_code += '<head>'
    html_code += '<title>Penny.com</title>'
    html_code += '</head>'
    html_code += '<body>'
    html_code += common.get_header()
    html_code += '<br>'
    html_code += 'Click here to <a href="/searchform">begin</a>.<br>'
    html_code += '<br>'
    html_code += common.get_footer()
    html_code += '</body>'
    html_code += '</html>'

    content_header = ('content-type', 'text/html; charset=utf-8')
    headers = [content_header]
    start_response('200 OK', headers)
    return [html_code.encode('utf-8')]

def search_form(environ, start_response):

    prev_author = '(None)'
    if 'HTTP_COOKIE' in environ:
        cookie = http.cookies.SimpleCookie(environ['HTTP_COOKIE'])
        if 'prev_author' in cookie:
            prev_author_morsel = cookie['prev_author']
            if prev_author_morsel:
                prev_author = prev_author_morsel.value

    html_code = ''
    html_code += '<!DOCTYPE html>'
    html_code += '<html>'
    html_code += '<head>'
    html_code += '<title>Penny.com</title>'
    html_code += '</head>'
    html_code += '<body>'
    html_code += common.get_header()
    html_code += '<h1>Author Search</h1>'
    html_code += '<form action="/searchresults" method="get">'
    html_code += 'Please enter an author name:'
    html_code += '<input type="text" name="author" autofocus>'
    html_code += '<input type="submit" value="Go">'
    html_code += '</form>'
    html_code += '<br>'
    html_code += '<br>'
    html_code += '<strong>Previous author search:</strong> '
    html_code += html.escape(prev_author)
    html_code += '<br>'
    html_code += '<br>'
```

PennyWsgi/penny.py (Page 2 of 3)

```
    html_code += common.get_footer()
    html_code += '</body>'
    html_code += '</html>'

    content_header = ('content-type', 'text/html; charset=utf-8')
    headers = [content_header]
    start_response('200 OK', headers)
    return [html_code.encode('utf-8')]

def search_results(environ, start_response):

    args_str = environ.get('QUERY_STRING', '')
    args = parseargs.parse(args_str)
    author = args.get('author', '')

    author = author.strip()

    if author == '':
        prev_author = '(None)'
        books = []
    else:
        prev_author = author
        books = database.get_books(author) # Exception handling omitted

    html_code = ''
    html_code += '<!DOCTYPE html>'
    html_code += '<html>'
    html_code += '<head>'
    html_code += '<title>Penny.com</title>'
    html_code += '</head>'
    html_code += '<body>'
    html_code += common.get_header()
    html_code += '<h1>Author Search Results</h1>'
    html_code += '<h2>Books by ' + html.escape(prev_author) + ':</h2>'

    if len(books) == 0:
        html_code += '(None)<br>'
    else:
        pattern = '%s: <strong>%s</strong>: %s<br>'
        for book in books:
            html_code += pattern % (book['isbn'], book['author'],
                book['title'])
    html_code += '<br>'
    html_code += '<br>'
    html_code += 'Click here to do another '
    html_code += '<a href="/searchform">author search</a>.'
    html_code += '<br>'
    html_code += '<br>'
    html_code += '<br>'
    html_code += '<br>'
    html_code += common.get_footer()
    html_code += '</body>'
    html_code += '</html>'

    content_header = ('content-type', 'text/html; charset=utf-8')
    cookie = http.cookies.SimpleCookie()
    cookie['prev_author'] = prev_author
    cookie_header = ('Set-Cookie', cookie['prev_author'].OutputString())
    headers = [content_header, cookie_header]
    start_response('200 OK', headers)
    return [html_code.encode('utf-8')]

def not_found(environ, start_response):

    # This is the response generated by Flask applications.
```

PennyWsgi/penny.py (Page 3 of 3)

```
131:     html_code = ''
132:     html_code += '<!DOCTYPE html>'
133:     html_code += '<title>404 Not Found</title>'
134:     html_code += '<h1>Not Found</h1>'
135:     html_code += '<p>The requested URL was not found on the server. '
136:     html_code += 'If you entered the URL manually please check your '
137:     html_code += 'spelling and try again.</p>'
138:
139:     content_header = ('content-type', 'text/html; charset=utf-8')
140:     headers = [content_header]
141:     start_response('200 OK', headers)
142:     return [html_code.encode('utf-8')]
143:
144: def app(environ, start_response):
145:
146:     path = environ.get('PATH_INFO', '').strip('/')
147:
148:     if path in ('', 'index'):
149:         return index(environ, start_response)
150:     if path == 'searchform':
151:         return search_form(environ, start_response)
152:     if path == 'searchresults':
153:         return search_results(environ, start_response)
154:
155:     return not_found(environ, start_response)
```