

PennyAjax3/penny.py (Page 1 of 1)

```
1: #!/usr/bin/env python
2:
3: #-----
4: # penny.py
5: # Author: Bob Dondero
6: #-----
7:
8: import json
9: import flask
10: import database
11:
12: #-----
13:
14: app = flask.Flask(__name__)
15:
16: #-----
17:
18: @app.route('/', methods=['GET'])
19: @app.route('/index', methods=['GET'])
20: def index():
21:
22:     return flask.send_file('index.html')
23:
24: #-----
25:
26: @app.route('/searchresults', methods=['GET'])
27: def search_results():
28:
29:     author = flask.request.args.get('author')
30:     if author is None:
31:         author = ''
32:     author = author.strip()
33:
34:     if author == '':
35:         books = []
36:     else:
37:         books = database.get_books(author) # Exception handling omitted
38:
39:     json_doc = json.dumps(books)
40:     response = flask.make_response(json_doc)
41:     response.headers['Content-Type'] = 'application/json'
42:     return response
```

blank (Page 1 of 1)

1: This page is intentionally blank.

PennyAjax3/index.html (Page 1 of 2)

```

1: <!DOCTYPE html>
2: <html>
3:   <head>
4:     <title>Penny.com</title>
5:   </head>
6:   <body>
7:     <hr>
8:     Good <span id="ampmSpan"></span> and welcome to
9:     <strong>Penny.com</strong>
10:    <hr>
11:    <h1>Author Search</h1>
12:    Please enter an author name:
13:    <input type="text" id="authorInput" autoFocus>
14:    <hr>
15:    <div id="resultsDiv"></div>
16:    <hr>
17:    Date and time: <span id="datetimeSpan"></span><br>
18:    Created by <a href="https://www.cs.princeton.edu/~rdonero">
19:    Bob Donero</a>
20:    <hr>
21:    <script>
22:      'use strict';
23:
24:      function getAmPm() {
25:        let dateTime = new Date();
26:        let hours = dateTime.getHours();
27:        let amPm = (hours < 12) ? 'morning' : 'afternoon';
28:        let ampMspan = document.getElementById('ampmSpan');
29:        ampMspan.innerHTML = amPm;
30:      }
31:
32:      function getDateTime() {
33:        let dateTime = new Date();
34:        let datetimeSpan =
35:        document.getElementById('datetimeSpan');
36:        datetimeSpan.innerHTML = dateTime.toLocaleString();
37:      }
38:
39:      function escape(s) {
40:        s = s.replace('&', '&amp;');
41:        s = s.replace('<', '&lt;');
42:        s = s.replace('>', '&gt;');
43:        s = s.replace('"', '&quot;');
44:        s = s.replace("'", '&apos;');
45:        return s;
46:      }
47:
48:      function convertToHtml(books) {
49:        let html = '';
50:        for (let book of books) {
51:          html += escape(book.isbn) + ': ';
52:          html += '<strong>';
53:          html += escape(book.author);
54:          html += '</strong>: ';
55:          html += escape(book.title) + '<br>';
56:        }
57:        return html;
58:      }
59:    </script>
60:  }
61: </body>
62: </html>

```

PennyAjax3/index.html (Page 2 of 2)

```

63:    function handleResponse() {
64:      if (this.status !== 200) {
65:        alert('Error: Failed to fetch data from server');
66:        return;
67:      }
68:      let books = JSON.parse(this.response);
69:      let html = convertToHtml(books);
70:      let resultsDiv = document.getElementById('resultsDiv');
71:      resultsDiv.innerHTML = html;
72:    }
73:
74:    function handleError() {
75:      alert('Error: Failed to fetch data from server');
76:    }
77:
78:    let request = null;
79:
80:    function getResults() {
81:      let authorInput = document.getElementById('authorInput');
82:      let author = authorInput.value;
83:      let encodedAuthor = encodeURIComponent(author);
84:      let url = '/searchresults?author=' + encodedAuthor;
85:      if (request !== null)
86:        request.abort();
87:      request = new XMLHttpRequest();
88:      request.onload = handleResponse;
89:      request.onerror = handleError;
90:      request.open('GET', url);
91:      request.send();
92:    }
93:
94:    let timer = null;
95:
96:    function debouncedGetResults() {
97:      clearTimeout(timer);
98:      timer = window.setTimeout(getResults, 500);
99:    }
100:
101:    function setupHeader() {
102:      getAmPm();
103:      let apInterval = window.setInterval(getAmPm, 1000);
104:      window.addEventListener('beforeunload',
105:        function(e) {window.clearInterval(apInterval);})
106:    }
107:
108:    function setupFooter() {
109:      getDateTime();
110:      let dtInterval = window.setInterval(getDateTime, 1000);
111:      window.addEventListener('beforeunload',
112:        function(e) {window.clearInterval(dtInterval);})
113:    }
114:
115:    function setup() {
116:      setupHeader();
117:      setupFooter();
118:      let authorInput = document.getElementById('authorInput');
119:      authorInput.addEventListener('input', debouncedGetResults);
120:    }
121:
122:    document.addEventListener('DOMContentLoaded', setup);
123:  }
124: </script>
125: </body>
126: </html>

```

PennyAjax4/index.html (Page 1 of 2)

```

1: <!DOCTYPE html>
2: <html>
3:   <head>
4:     <title>Penny.com</title>
5:   </head>
6:
7:   <body>
8:
9:     <hr>
10:    Good <span id="ampmSpan"></span> and welcome to
11:    <strong>Penny.com</strong>
12:    <hr>
13:
14:    <h1>Author Search</h1>
15:    Please enter an author name:
16:    <input type="text" id="authorInput" autoFocus>
17:    <hr>
18:    <div id="resultsDiv"></div>
19:
20:    <hr>
21:    Date and time: <span id="datetimeSpan"></span><br>
22:    Created by <a href="https://www.cs.princeton.edu/~rdonero">
23:    Bob Donero</a>
24:    <hr>
25:
26:    <script src=
27:    "https://cdn.jsdelivr.net/npm/mustache@4.2.0/mustache.min.js">
28:    </script>
29:
30:    <!-- <script src="/static/mustache.min.js"></script> -->
31:
32:    <script>
33:
34:      'use strict';
35:
36:      function getAmPm() {
37:        let dateTime = new Date();
38:        let hours = dateTime.getHours();
39:        let amPm = (hours < 12) ? 'morning' : 'afternoon';
40:        let ampMspan = document.getElementById('ampmSpan');
41:        ampMspan.innerHTML = amPm;
42:      }
43:
44:      function getDateTime() {
45:        let dateTime = new Date();
46:        let datetimeSpan =
47:        document.getElementById('datetimeSpan');
48:        datetimeSpan.innerHTML = dateTime.toLocaleString();
49:      }
50:
51:      function convertToHtml(books) {
52:        let template = `
53:          {{#books}}
54:            {{isbn}}:
55:            <strong>{{author}}</strong>:
56:            {{title}}
57:            <br>
58:          {{/books}}
59:        `;
60:        let map = {books: books};
61:        let html = Mustache.render(template, map);
62:        return html;
63:      }
64:
65:      function handleResponse() {

```

PennyAjax4/index.html (Page 2 of 2)

```

66:        if (this.status !== 200) {
67:          alert('Error: Failed to fetch data from server');
68:          return;
69:        }
70:        let books = JSON.parse(this.response);
71:        let html = convertToHtml(books);
72:        let resultsDiv = document.getElementById('resultsDiv');
73:        resultsDiv.innerHTML = html;
74:      }
75:
76:      function handleError() {
77:        alert('Error: Failed to fetch data from server');
78:      }
79:
80:      let request = null;
81:
82:      function getResults() {
83:        let authorInput = document.getElementById('authorInput');
84:        let author = authorInput.value;
85:        let encodedAuthor = encodeURIComponent(author);
86:        let url = '/searchresults?author=' + encodedAuthor;
87:        if (request !== null)
88:          request.abort();
89:        request = new XMLHttpRequest();
90:        request.onload = handleResponse;
91:        request.onerror = handleError;
92:        request.open('GET', url);
93:        request.send();
94:      }
95:
96:      let timer = null;
97:
98:      function debouncedGetResults() {
99:        clearTimeout(timer);
100:        timer = window.setTimeout(getResults, 500);
101:      }
102:
103:      function setupHeader() {
104:        getAmPm();
105:        let apInterval = window.setInterval(getAmPm, 1000);
106:        window.addEventListener('beforeunload',
107:          function(e) {window.clearInterval(apInterval);})
108:      }
109:
110:      function setupFooter() {
111:        getDateTime();
112:        let dtInterval = window.setInterval(getDateTime, 1000);
113:        window.addEventListener('beforeunload',
114:          function(e) {window.clearInterval(dtInterval);})
115:      }
116:
117:      function setup() {
118:        setupHeader();
119:        setupFooter();
120:        let authorInput = document.getElementById('authorInput');
121:        authorInput.addEventListener('input', debouncedGetResults);
122:      }
123:
124:      document.addEventListener('DOMContentLoaded', setup);
125:
126:    </script>
127:  </body>
128: </html>

```

PennyjQuery/index.html (Page 1 of 2)

```

1: <!DOCTYPE html>
2: <html>
3:   <head>
4:     <title>Penny.com</title>
5:   </head>
6:
7:   <body>
8:     <hr>
9:     Good <span id="ampmSpan"></span> and welcome to
10:    <strong>Penny.com</strong>
11:    <hr>
12:
13:    <h1>Author Search</h1>
14:    Please enter an author name:
15:    <input type="text" id="authorInput" autoFocus>
16:    <hr>
17:    <div id="resultsDiv"></div>
18:
19:    <hr>
20:    Date and time: <span id="datetimeSpan"></span><br>
21:    Created by <a href="https://www.cs.princeton.edu/~rdonero">
22:    Bob Donero</a>
23:    <hr>
24:
25:    <script src=
26:      "https://cdn.jsdelivr.net/npm/jquery@3.7.1/dist/jquery.min.js">
27:    </script>
28:
29:    <script src=
30:      "https://cdn.jsdelivr.net/npm/mustache@4.2.0/mustache.min.js">
31:    </script>
32:
33:    <script>
34:      'use strict';
35:
36:      function getAmPm() {
37:        let dateTime = new Date();
38:        let hours = dateTime.getHours();
39:        let amPm = 'morning';
40:        if (hours >= 12)
41:          amPm = 'afternoon';
42:        $('#ampmSpan').html(amPm);
43:      }
44:
45:      function getDateTime() {
46:        let dateTime = new Date();
47:        $('#datetimeSpan').html(dateTime.toLocaleString());
48:      }
49:
50:      function convertToHtml(books) {
51:        let template = `
52:          {{#books}}
53:            {{isbn}}:
54:            <strong>{{author}}</strong>:
55:            {{title}}
56:            <br>
57:          {{/books}}
58:        `;
59:        let map = {books: books};
60:        let html = Mustache.render(template, map);
61:        return html;
62:      }
63:
64:
65:      function handleResponse(books) {

```

PennyjQuery/index.html (Page 2 of 2)

```

66:        let html = convertToHtml(books);
67:        $('#resultsDiv').html(html);
68:      }
69:
70:      function handleError(request) {
71:        if (request.statusText !== 'abort')
72:          alert('Error: Failed to fetch data from server');
73:      }
74:
75:      let request = null;
76:
77:      function getResults() {
78:        let author = $('#authorInput').val();
79:        let encodedAuthor = encodeURIComponent(author);
80:        let url = '/searchresults?author=' + encodedAuthor;
81:        if (request !== null)
82:          request.abort();
83:        let requestData = {
84:          type: 'GET',
85:          url: url,
86:          success: handleResponse,
87:          error: handleError
88:        };
89:        request = $.ajax(requestData);
90:      }
91:
92:      let timer = null;
93:
94:      function debouncedGetResults() {
95:        clearTimeout(timer);
96:        timer = window.setTimeout(getResults, 500);
97:      }
98:
99:      function setupHeader() {
100:        getAmPm();
101:        let apInterval = window.setInterval(getAmPm, 1000);
102:        $(window).on('beforeunload',
103:          function(e) {window.clearInterval(apInterval);})
104:      }
105:
106:      function setupFooter() {
107:        getDateTime();
108:        let dtInterval = window.setInterval(getDateTime, 1000);
109:        $(window).on('beforeunload',
110:          function(e) {window.clearInterval(dtInterval);})
111:      }
112:
113:      function setup() {
114:        setupHeader();
115:        setupFooter();
116:        $('#authorInput').on('input', debouncedGetResults);
117:      }
118:
119:      $('document').ready(setup);
120:
121:    </script>
122:  </body>
123: </html>

```