PennyJson/penny.py (Page 1 of 1)

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

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

import os
import time
import json
import flask
import dotenv
import database

dotenv.load_dotenv()
_IO_DELAY = int(os.getenv('IO_DELAY', '0'))

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

app = flask.Flask(__name__)

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

@app.route('/', methods=['GET'])
@app.route('/index', methods=['GET'])
def index():

    return flask.send_file('index.html')

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

@app.route('/searchresults', methods=['GET'])
def search_results():

    author = flask.request.args.get('author')
    if author is None:
        author = ''
    author = author.strip()

    # Simulate a slow database.
    time.sleep(_IO_DELAY)

    if author == '':
        books = []
    else:
        books = database.get_books(author) # Exception handling omitted

    json_doc = json.dumps(books)
    response = flask.make_response(json_doc)
    response.headers['Content-Type'] = 'application/json'
    return response
```

blank (Page 1 of 1)

```
This page is intentionally blank.
```

PennyPyqt/pennydesktop1base.py (Page 1 of 2)

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

#-----------------------------------------------------------------------
# pennydesktop1base.py
# Author: Bob Dondero
#-----------------------------------------------------------------------

import sys
import urllib.parse
import urllib.request
import json
import PyQt5.QtWidgets

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

def create_widgets():

    author_label = PyQt5.QtWidgets.QLabel('Author: ')
    author_lineedit = PyQt5.QtWidgets.QLineEdit()
    submit_button = PyQt5.QtWidgets.QPushButton('Submit')
    books_textedit = PyQt5.QtWidgets.QTextEdit()
    books_textedit.setReadOnly(True)

    layout = PyQt5.QtWidgets.QGridLayout()
    layout.addWidget(author_label, 0, 0)
    layout.addWidget(author_lineedit, 0, 1)
    layout.addWidget(submit_button, 0, 2)
    layout.addWidget(books_textedit, 1, 0, 1, 3)
    layout.setRowStretch(0, 0)
    layout.setRowStretch(1, 1)
    layout.setColumnStretch(0, 0)
    layout.setColumnStretch(1, 1)
    layout.setColumnStretch(2, 0)

    frame = PyQt5.QtWidgets.QFrame()
    frame.setLayout(layout)

    window = PyQt5.QtWidgets.QMainWindow()
    window.setWindowTitle('Penny: Author Search')
    window.setCentralWidget(frame)
    screen_size = PyQt5.QtWidgets.QDesktopWidget().screenGeometry()
    window.resize(screen_size.width()//2, screen_size.height()//2)

    return (window, author_lineedit, submit_button, books_textedit)

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

def author_slot_helper(server_url, author_lineedit, books_textedit):

    author = author_lineedit.text()
    encoded_author = urllib.parse.quote_plus(author)
    url = server_url + '/searchresults?author=' + encoded_author

    books_textedit.clear()

    try:
        with urllib.request.urlopen(url) as in_flo:
            response = in_flo.read()
            json_doc = response.decode('utf-8')
            books = json.loads(json_doc)

            if len(books) == 0:
                books_textedit.insertPlainText('(None)')
            else:
                pattern = '%s: <strong>%s</strong>: %s<br>'
                for book in books:
                    books_textedit.insertHtml(pattern %
                        (book['isbn'], book['author'], book['title']))

    except Exception as ex:
        books_textedit.insertPlainText(str(ex))

    books_textedit.repaint()  # Required on Mac.

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

def main():

    if len(sys.argv) != 2:
        print('Usage: penny serverURL', file=sys.stderr)
        sys.exit(1)

    server_url = sys.argv[1]

    # Create and lay out the widgets.

    app = PyQt5.QtWidgets.QApplication(sys.argv)
    window, author_lineedit, submit_button, books_textedit = (
        create_widgets())

    # Handle signals.

    def author_slot():
        author_slot_helper(server_url, author_lineedit, books_textedit)
    submit_button.clicked.connect(author_slot)

    # Start up.

    window.show()
    author_slot()  # Populate books_textedit initially.
    sys.exit(app.exec_())

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

PennyPyqt/pennydesktop1base.py (Page 2 of 2)


```
                for book in books:
                    books_textedit.insertHtml(pattern %
                        (book['isbn'], book['author'], book['title']))

    except Exception as ex:
        books_textedit.insertPlainText(str(ex))

    books_textedit.repaint()  # Required on Mac.

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

def main():

    if len(sys.argv) != 2:
        print('Usage: penny serverURL', file=sys.stderr)
        sys.exit(1)

    server_url = sys.argv[1]

    # Create and lay out the widgets.

    app = PyQt5.QtWidgets.QApplication(sys.argv)
    window, author_lineedit, submit_button, books_textedit = (
        create_widgets())

    # Handle signals.

    def author_slot():
        author_slot_helper(server_url, author_lineedit, books_textedit)
    submit_button.clicked.connect(author_slot)

    # Start up.

    window.show()
    author_slot()  # Populate books_textedit initially.
    sys.exit(app.exec_())

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

PennyPyqt/pennydesktop2seq.py (Page 1 of 2)

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

#-----------------------------------------------------------------------
# pennydesktop1seq.py
# Author: Bob Dondero
#-----------------------------------------------------------------------

import sys
import urllib.parse
import urllib.request
import json
import PyQt5.QtWidgets

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

def create_widgets():

    author_label = PyQt5.QtWidgets.QLabel('Author: ')
    author_lineedit = PyQt5.QtWidgets.QLineEdit()
    books_textedit = PyQt5.QtWidgets.QTextEdit()
    books_textedit.setReadOnly(True)

    layout = PyQt5.QtWidgets.QGridLayout()
    layout.addWidget(author_label, 0, 0)
    layout.addWidget(author_lineedit, 0, 1)
    layout.addWidget(books_textedit, 1, 0, 1, 2)
    layout.setRowStretch(0, 0)
    layout.setRowStretch(1, 1)
    layout.setColumnStretch(0, 0)
    layout.setColumnStretch(1, 1)
    layout.setColumnStretch(2, 0)

    frame = PyQt5.QtWidgets.QFrame()
    frame.setLayout(layout)

    window = PyQt5.QtWidgets.QMainWindow()
    window.setWindowTitle('Penny: Author Search')
    window.setCentralWidget(frame)
    screen_size = PyQt5.QtWidgets.QDesktopWidget().screenGeometry()
    window.resize(screen_size.width()//2, screen_size.height()//2)

    return (window, author_lineedit, books_textedit)

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

def author_slot_helper(server_url, author_lineedit, books_textedit):

    author = author_lineedit.text()
    encoded_author = urllib.parse.quote_plus(author)
    url = server_url + '/searchresults?author=' + encoded_author

    books_textedit.clear()

    try:
        with urllib.request.urlopen(url) as in_flo:
            response = in_flo.read()
            json_doc = response.decode('utf-8')
            books = json.loads(json_doc)

            if len(books) == 0:
                books_textedit.insertPlainText('(None)')
            else:
                pattern = '%s: <strong>%s</strong>: %s<br>'
                for book in books:
                    books_textedit.insertHtml(pattern %
```

PennyPyqt/pennydesktop2seq.py (Page 2 of 2)

```
                              (book['isbn'], book['author'], book['title']))

    except Exception as ex:
        books_textedit.insertPlainText(str(ex))

    books_textedit.repaint()  # Required on Mac.

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

def main():

    if len(sys.argv) != 2:
        print('Usage: penny serverURL', file=sys.stderr)
        sys.exit(1)

    server_url = sys.argv[1]

    # Create and lay out the widgets.

    app = PyQt5.QtWidgets.QApplication(sys.argv)
    window, author_lineedit, books_textedit = create_widgets()

    # Handle signals.

    def author_slot():
        author_slot_helper(server_url, author_lineedit, books_textedit)
    author_lineedit.textChanged.connect(author_slot)

    # Start up.

    window.show()
    author_slot()  # Populate books_textedit initially.
    sys.exit(app.exec_())

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

PennyPyqt/pennydesktop3bad.py (Page 1 of 2)

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

#-----------------------------------------------------------------------
# pennydesktop3bad.py
# Author: Bob Dondero
#-----------------------------------------------------------------------

import sys
import threading
import urllib.parse
import urllib.request
import json
import PyQt5.QtWidgets

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

def create_widgets():

    author_label = PyQt5.QtWidgets.QLabel('Author: ')
    author_lineedit = PyQt5.QtWidgets.QLineEdit()
    books_textedit = PyQt5.QtWidgets.QTextEdit()
    books_textedit.setReadOnly(True)

    layout = PyQt5.QtWidgets.QGridLayout()
    layout.addWidget(author_label, 0, 0)
    layout.addWidget(author_lineedit, 0, 1)
    layout.addWidget(books_textedit, 1, 0, 1, 2)
    layout.setRowStretch(0, 0)
    layout.setRowStretch(1, 1)
    layout.setColumnStretch(0, 0)
    layout.setColumnStretch(1, 1)
    layout.setColumnStretch(2, 0)

    frame = PyQt5.QtWidgets.QFrame()
    frame.setLayout(layout)

    window = PyQt5.QtWidgets.QMainWindow()
    window.setWindowTitle('Penny: Author Search')
    window.setCentralWidget(frame)
    screen_size = PyQt5.QtWidgets.QDesktopWidget().screenGeometry()
    window.resize(screen_size.width()//2, screen_size.height()//2)

    return (window, author_lineedit, books_textedit)

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

class WorkerThread (threading.Thread):

    def __init__(self, server_url, author, books_textedit):
        threading.Thread.__init__(self)
        self._server_url = server_url
        self._author = author
        self._books_textedit = books_textedit

    def run(self):
        encoded_author = urllib.parse.quote_plus(self._author)
        url = self._server_url
        url += '/searchresults?author=' + encoded_author

        self._books_textedit.clear()

        try:
            with urllib.request.urlopen(url) as in_flo:
                response = in_flo.read()
                json_doc = response.decode('utf-8')
                books = json.loads(json_doc)

                if len(books) == 0:
                    self._books_textedit.insertPlainText('(None)')
                else:
                    pattern = '%s: <strong>%s</strong>: %s<br>'
                    for book in books:
                        self._books_textedit.insertHtml(pattern %
                            (book['isbn'], book['author'],
                                book['title']))

        except Exception as ex:
            self._books_textedit.insertPlainText(str(ex))

        self._books_textedit.repaint()  # Required on Mac.

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

def main():

    if len(sys.argv) != 2:
        print('Usage: penny serverURL', file=sys.stderr)
        sys.exit(1)

    server_url = sys.argv[1]

    # Create and lay out the widgets.

    app = PyQt5.QtWidgets.QApplication(sys.argv)
    window, author_lineedit, books_textedit = create_widgets()

    # Handle signals.

    def author_slot():
        author = author_lineedit.text()
        worker_thread = WorkerThread(server_url, author, books_textedit)
        worker_thread.start()
    author_lineedit.textChanged.connect(author_slot)

    # Start up.

    window.show()
    author_slot()  # Populate books_textedit initially.
    sys.exit(app.exec_())

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

PennyPyqt/pennydesktop3bad.py (Page 2 of 2)


```
                books = json.loads(json_doc)

                if len(books) == 0:
                    self._books_textedit.insertPlainText('(None)')
                else:
                    pattern = '%s: <strong>%s</strong>: %s<br>'
                    for book in books:
                        self._books_textedit.insertHtml(pattern %
                            (book['isbn'], book['author'],
                                book['title']))

        except Exception as ex:
            self._books_textedit.insertPlainText(str(ex))

        self._books_textedit.repaint()  # Required on Mac.

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

def main():

    if len(sys.argv) != 2:
        print('Usage: penny serverURL', file=sys.stderr)
        sys.exit(1)

    server_url = sys.argv[1]

    # Create and lay out the widgets.

    app = PyQt5.QtWidgets.QApplication(sys.argv)
    window, author_lineedit, books_textedit = create_widgets()

    # Handle signals.

    def author_slot():
        author = author_lineedit.text()
        worker_thread = WorkerThread(server_url, author, books_textedit)
        worker_thread.start()
    author_lineedit.textChanged.connect(author_slot)

    # Start up.

    window.show()
    author_slot()  # Populate books_textedit initially.
    sys.exit(app.exec_())

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

PennyPyqt/pennydesktop4threads.py (Page 1 of 2)

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

#-------------------------------------------------------------------------
# pennydesktop4threads.py
# Author: Bob Dondero
#-------------------------------------------------------------------------

import sys
import urllib.parse
import urllib.request
import json
import PyQt5.QtWidgets
import PyQt5.QtCore

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

def create_widgets():

    author_label = PyQt5.QtWidgets.QLabel('Author: ')
    author_lineedit = PyQt5.QtWidgets.QLineEdit()
    books_textedit = PyQt5.QtWidgets.QTextEdit()
    books_textedit.setReadOnly(True)

    layout = PyQt5.QtWidgets.QGridLayout()
    layout.addWidget(author_label, 0, 0)
    layout.addWidget(author_lineedit, 0, 1)
    layout.addWidget(books_textedit, 1, 0, 1, 2)
    layout.setRowStretch(0, 0)
    layout.setRowStretch(1, 1)
    layout.setColumnStretch(0, 0)
    layout.setColumnStretch(1, 1)
    layout.setColumnStretch(2, 0)

    frame = PyQt5.QtWidgets.QFrame()
    frame.setLayout(layout)

    window = PyQt5.QtWidgets.QMainWindow()
    window.setWindowTitle('Penny: Author Search')
    window.setCentralWidget(frame)
    screen_size = PyQt5.QtWidgets.QDesktopWidget().screenGeometry()
    window.resize(screen_size.width()//2, screen_size.height()//2)

    return (window, author_lineedit, books_textedit)

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

class WorkerThread (PyQt5.QtCore.QThread):

    _response_signal = PyQt5.QtCore.pyqtSignal(bool, object)

    def __init__(self, server_url, author, handle_response):
        super().__init__()
        self._server_url = server_url
        self._author = author
        self._response_signal.connect(handle_response)

    def run(self):
        encoded_author = urllib.parse.quote_plus(self._author)
        url = self._server_url
        url += '/searchresults?author=' + encoded_author
        try:
            with urllib.request.urlopen(url) as in_flo:
                response = in_flo.read()
                json_doc = response.decode('utf-8')
                books = json.loads(json_doc)
            self._response_signal.emit(True, books)
        except Exception as ex:
            self._response_signal.emit(False, str(ex))

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

def main():

    if len(sys.argv) != 2:
        print('Usage: penny serverURL', file=sys.stderr)
        sys.exit(1)

    server_url = sys.argv[1]

    # Create and lay out the widgets.

    app = PyQt5.QtWidgets.QApplication(sys.argv)
    window, author_lineedit, books_textedit = create_widgets()

    # Handle signals.

    def handle_response(successful, data):
        books_textedit.clear()
        if successful:
            books = data
            if len(books) == 0:
                books_textedit.insertPlainText('(None)')
            else:
                pattern = '%s: <strong>%s</strong>: %s<br>'
                for book in books:
                    books_textedit.insertHtml(pattern %
                        (book['isbn'], book['author'], book['title']))
        else:
            ex = data
            books_textedit.insertPlainText(ex)
        books_textedit.repaint()

    worker_thread = None
    def author_slot():
        nonlocal worker_thread
        author = author_lineedit.text()
        worker_thread = WorkerThread(server_url, author,
            handle_response)
        worker_thread.start()

    author_lineedit.textChanged.connect(author_slot)

    # Start up.

    window.show()
    author_slot()  # Populate books_textedit initially.
    sys.exit(app.exec_())

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

PennyPyqt/pennydesktop4threads.py (Page 2 of 2)


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

#-------------------------------------------------------------------------
# pennydesktop4threads.py
# Author: Bob Dondero
#-------------------------------------------------------------------------

import sys
import urllib.parse
import urllib.request
import json
import PyQt5.QtWidgets
import PyQt5.QtCore

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

def create_widgets():

    author_label = PyQt5.QtWidgets.QLabel('Author: ')
    author_lineedit = PyQt5.QtWidgets.QLineEdit()
    books_textedit = PyQt5.QtWidgets.QTextEdit()
    books_textedit.setReadOnly(True)

    layout = PyQt5.QtWidgets.QGridLayout()
    layout.addWidget(author_label, 0, 0)
    layout.addWidget(author_lineedit, 0, 1)
    layout.addWidget(books_textedit, 1, 0, 1, 2)
    layout.setRowStretch(0, 0)
    layout.setRowStretch(1, 1)
    layout.setColumnStretch(0, 0)
    layout.setColumnStretch(1, 1)
    layout.setColumnStretch(2, 0)

    frame = PyQt5.QtWidgets.QFrame()
    frame.setLayout(layout)

    window = PyQt5.QtWidgets.QMainWindow()
    window.setWindowTitle('Penny: Author Search')
    window.setCentralWidget(frame)
    screen_size = PyQt5.QtWidgets.QDesktopWidget().screenGeometry()
    window.resize(screen_size.width()//2, screen_size.height()//2)

    return (window, author_lineedit, books_textedit)

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

class WorkerThread (PyQt5.QtCore.QThread):

    _response_signal = PyQt5.QtCore.pyqtSignal(bool, object)

    def __init__(self, server_url, author, handle_response):
        super().__init__()
        self._server_url = server_url
        self._author = author
        self._response_signal.connect(handle_response)

    def run(self):
        encoded_author = urllib.parse.quote_plus(self._author)
        url = self._server_url
        url += '/searchresults?author=' + encoded_author
        try:
            with urllib.request.urlopen(url) as in_flo:
                response = in_flo.read()
                json_doc = response.decode('utf-8')
                books = json.loads(json_doc)
            self._response_signal.emit(True, books)
        except Exception as ex:
            self._response_signal.emit(False, str(ex))

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

def main():

    if len(sys.argv) != 2:
        print('Usage: penny serverURL', file=sys.stderr)
        sys.exit(1)

    server_url = sys.argv[1]

    # Create and lay out the widgets.

    app = PyQt5.QtWidgets.QApplication(sys.argv)
    window, author_lineedit, books_textedit = create_widgets()

    # Handle signals.

    def handle_response(successful, data):
        books_textedit.clear()
        if successful:
            books = data
            if len(books) == 0:
                books_textedit.insertPlainText('(None)')
            else:
                pattern = '%s: <strong>%s</strong>: %s<br>'
                for book in books:
                    books_textedit.insertHtml(pattern %
                        (book['isbn'], book['author'], book['title']))
        else:
            ex = data
            books_textedit.insertPlainText(ex)
        books_textedit.repaint()

    worker_thread = None
    def author_slot():
        nonlocal worker_thread
        author = author_lineedit.text()
        worker_thread = WorkerThread(server_url, author,
            handle_response)
        worker_thread.start()

    author_lineedit.textChanged.connect(author_slot)

    # Start up.

    window.show()
    author_slot()  # Populate books_textedit initially.
    sys.exit(app.exec_())

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

PennyPyqt/pennydesktop5stop.py (Page 1 of 2)

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

#-------------------------------------------------------------------------
# pennydesktop5stop.py
# Author: Bob Dondero
#-------------------------------------------------------------------------

import sys
import threading
import urllib.parse
import urllib.request
import json
import queue
import PyQt5.QtWidgets
import PyQt5.QtCore

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

def create_widgets():

    author_label = PyQt5.QtWidgets.QLabel('Author: ')
    author_lineedit = PyQt5.QtWidgets.QLineEdit()
    books_textedit = PyQt5.QtWidgets.QTextEdit()
    books_textedit.setReadOnly(True)

    layout = PyQt5.QtWidgets.QGridLayout()
    layout.addWidget(author_label, 0, 0)
    layout.addWidget(author_lineedit, 0, 1)
    layout.addWidget(books_textedit, 1, 0, 1, 2)
    layout.setRowStretch(0, 0)
    layout.setRowStretch(1, 1)
    layout.setColumnStretch(0, 0)
    layout.setColumnStretch(1, 1)
    layout.setColumnStretch(2, 0)

    frame = PyQt5.QtWidgets.QFrame()
    frame.setLayout(layout)

    window = PyQt5.QtWidgets.QMainWindow()
    window.setWindowTitle('Penny: Author Search')
    window.setCentralWidget(frame)
    screen_size = PyQt5.QtWidgets.QDesktopWidget().screenGeometry()
    window.resize(screen_size.width()//2, screen_size.height()//2)

    return (window, author_lineedit, books_textedit)

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

class WorkerThread (PyQt5.QtCore.QThread):

    _response_signal = PyQt5.QtCore.pyqtSignal(bool, object)

    def __init__(self, server_url, author, handle_response):
        super().__init__()
        self._server_url = server_url
        self._author = author
        self._response_signal.connect(handle_response)
        self._should_stop = False

    def stop(self):
        self._should_stop = True

    def run(self):
        encoded_author = urllib.parse.quote_plus(self._author)
        url = self._server_url
```

PennyPyqt/pennydesktop5stop.py (Page 2 of 2)

```
        url += '/searchresults?author=' + encoded_author
        try:
            with urllib.request.urlopen(url) as in_flo:
                response = in_flo.read()
                json_doc = response.decode('utf-8')
                books = json.loads(json_doc)
            if not self._should_stop:
                self._response_signal.emit(True, books)
        except Exception as ex:
            if not self._should_stop:
                self._response_signal.emit(False, str(ex))

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

def main():

    if len(sys.argv) != 2:
        print('Usage: penny serverURL', file=sys.stderr)
        sys.exit(1)

    server_url = sys.argv[1]

    # Create and lay out the widgets.

    app = PyQt5.QtWidgets.QApplication(sys.argv)
    window, author_lineedit, books_textedit = create_widgets()

    # Handle signals.

    def handle_response(successful, data):
        books_textedit.clear()
        if successful:
            books = data
            if len(books) == 0:
                books_textedit.insertPlainText('(None)')
            else:
                pattern = '%s: <strong>%s</strong>: %s<br>'
                for book in books:
                    books_textedit.insertHtml(pattern %
                        (book['isbn'], book['author'], book['title']))
        else:
            ex = data
            books_textedit.insertPlainText(ex)
        books_textedit.repaint()

    worker_thread = None
    def author_slot():
        nonlocal worker_thread
        author = author_lineedit.text()
        if worker_thread is not None:
            worker_thread.stop()
        worker_thread = WorkerThread(server_url, author,
            handle_response)
        worker_thread.start()

    author_lineedit.textChanged.connect(author_slot)

    # Start up.

    window.show()
    author_slot()  # Populate books_textedit initially.
    sys.exit(app.exec_())

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

PennyPyqt/pennydesktop6debounce.py (Page 1 of 3)

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

#-------------------------------------------------------------------------
# pennydesktop6debounce.py
# Author: Bob Dondero
#-------------------------------------------------------------------------

import sys
import urllib.parse
import urllib.request
import json
import PyQt5.QtWidgets
import PyQt5.QtCore

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

def create_widgets():

    author_label = PyQt5.QtWidgets.QLabel('Author: ')
    author_lineedit = PyQt5.QtWidgets.QLineEdit()
    books_textedit = PyQt5.QtWidgets.QTextEdit()
    books_textedit.setReadOnly(True)

    layout = PyQt5.QtWidgets.QGridLayout()
    layout.addWidget(author_label, 0, 0)
    layout.addWidget(author_lineedit, 0, 1)
    layout.addWidget(books_textedit, 1, 0, 1, 2)
    layout.setRowStretch(0, 0)
    layout.setRowStretch(1, 1)
    layout.setColumnStretch(0, 0)
    layout.setColumnStretch(1, 1)
    layout.setColumnStretch(2, 0)

    frame = PyQt5.QtWidgets.QFrame()
    frame.setLayout(layout)

    window = PyQt5.QtWidgets.QMainWindow()
    window.setWindowTitle('Penny: Author Search')
    window.setCentralWidget(frame)
    screen_size = PyQt5.QtWidgets.QDesktopWidget().screenGeometry()
    window.resize(screen_size.width()//2, screen_size.height()//2)

    return (window, author_lineedit, books_textedit)

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

class WorkerThread (PyQt5.QtCore.QThread):

    _response_signal = PyQt5.QtCore.pyqtSignal(bool, object)

    def __init__(self, server_url, author, handle_response):
        super().__init__()
        self._server_url = server_url
        self._author = author
        self._response_signal.connect(handle_response)
        self._should_stop = False

    def stop(self):
        self._should_stop = True

    def run(self):
        encoded_author = urllib.parse.quote_plus(self._author)
        url = self._server_url
        url += '/searchresults?author=' + encoded_author
        try:
```

PennyPyqt/pennydesktop6debounce.py (Page 2 of 3)

```
            with urllib.request.urlopen(url) as in_flo:
                response = in_flo.read()
                json_doc = response.decode('utf-8')
                books = json.loads(json_doc)
            if not self._should_stop:
                self._response_signal.emit(True, books)
        except Exception as ex:
            if not self._should_stop:
                self._response_signal.emit(False, str(ex))

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

def main():

    if len(sys.argv) != 2:
        print('Usage: penny serverURL', file=sys.stderr)
        sys.exit(1)

    server_url = sys.argv[1]

    # Create and lay out the widgets.

    app = PyQt5.QtWidgets.QApplication(sys.argv)
    window, author_lineedit, books_textedit = create_widgets()

    # Handle signals.

    def handle_response(successful, data):
        books_textedit.clear()
        if successful:
            books = data
            if len(books) == 0:
                books_textedit.insertPlainText('(None)')
            else:
                pattern = '%s: <strong>%s</strong>: %s<br>'
                for book in books:
                    books_textedit.insertHtml(pattern %
                        (book['isbn'], book['author'], book['title']))
        else:
            ex = data
            books_textedit.insertPlainText(ex)
        books_textedit.repaint()

    worker_thread = None
    def author_slot():
        nonlocal worker_thread
        author = author_lineedit.text()
        if worker_thread is not None:
            worker_thread.stop()
        worker_thread = WorkerThread(server_url, author,
            handle_response)
        worker_thread.start()

    debounce_timer = None
    def debounced_author_slot():
        nonlocal debounce_timer
        debounce_timer = PyQt5.QtCore.QTimer()
        debounce_timer.timeout.connect(author_slot)
        debounce_timer.setSingleShot(True)
        debounce_timer.setInterval(500) # milliseconds
        debounce_timer.start()

    author_lineedit.textChanged.connect(debounced_author_slot)

    # Start up.
```

PennyPyqt/pennydesktop6debounce.py (Page 3 of 3)

```
131:
132:     window.show()
133:     author_slot()  # Populate books_textedit initially.
134:     sys.exit(app.exec_())
135:
136: if __name__ == '__main__':
137:     main()
```