

pennyheroku/databasesqlite.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # databasesqlite.py
5: # Author: Bob Dondero
6: #-----
7:
8: import sqlite3
9: import contextlib
10:
11: #-----
12:
13: _DATABASE_URL = 'file:penny.sqlite'
14:
15: #-----
16:
17: def get_books(author):
18:
19:     books = []
20:
21:     with contextlib.closing(
22:         sqlite3.connect(_DATABASE_URL + '?mode=ro',
23:             isolation_level=None, uri=True)) as connection:
24:
25:         with contextlib.closing(connection.cursor()) as cursor:
26:
27:             query_str = '''
28:                 SELECT isbn, author, title FROM books
29:                 WHERE author LIKE ?
30:             '''
31:             cursor.execute(query_str, [author+'%'])
32:
33:             table = cursor.fetchall()
34:             for row in table:
35:                 book = {'isbn': row[0], 'author': row[1],
36:                     'title': row[2]}
37:                 books.append(book)
38:
39:     return books
40:
41: #-----
42:
43: # For testing:
44:
45: def _test():
46:     books = get_books('ker')
47:     for book in books:
48:         print(book['isbn'])
49:         print(book['author'])
50:         print(book['title'])
51:         print()
52:
53: if __name__ == '__main__':
54:     _test()

```

pennyheroku/databasepostgres1.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # databasepostgres1.py
5: # Author: Bob Dondero
6: #-----
7:
8: import os
9: import contextlib
10: import psycopg
11: import dotenv
12:
13: #-----
14:
15: dotenv.load_dotenv()
16: _DATABASE_URL = os.environ['DATABASE_URL']
17:
18: #-----
19:
20: def get_books(author):
21:
22:     books = []
23:
24:     with psycopg.connect(_DATABASE_URL) as connection:
25:
26:         with connection.cursor() as cursor:
27:             query_str = '''
28:                 SELECT isbn, author, title FROM books
29:                 WHERE author ILIKE %s
30:             '''
31:             cursor.execute(query_str, [author+'%'])
32:
33:             table = cursor.fetchall()
34:             for row in table:
35:                 book = {'isbn': row[0], 'author': row[1],
36:                     'title': row[2]}
37:                 books.append(book)
38:
39:     return books
40:
41: #-----
42:
43: # For testing:
44:
45: def _test():
46:     books = get_books('ker')
47:     for book in books:
48:         print(book['isbn'])
49:         print(book['author'])
50:         print(book['title'])
51:         print()
52:
53: if __name__ == '__main__':
54:     _test()

```

pennyheroku/databasepostgres2bad.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # databasepostgres2bad.py
5: # Author: Bob Dondero
6: #-----
7:
8: import os
9: import contextlib
10: import psycopg
11: import dotenv
12:
13: #-----
14:
15: dotenv.load_dotenv()
16: _DATABASE_URL = os.environ['DATABASE_URL']
17:
18: _connection = psycopg.connect(_DATABASE_URL)
19:
20: #-----
21:
22: def get_books(author):
23:
24:     books = []
25:
26:     with _connection.cursor() as cursor:
27:         query_str = '''
28:             SELECT isbn, author, title FROM books
29:             WHERE author ILIKE %s
30:             '''
31:         cursor.execute(query_str, [author+'%'])
32:
33:         table = cursor.fetchall()
34:         for row in table:
35:             book = {'isbn': row[0], 'author': row[1],
36:                   'title': row[2]}
37:             books.append(book)
38:
39:     return books
40:
41: #-----
42:
43: # For testing:
44:
45: def _test():
46:     books = get_books('ker')
47:     for book in books:
48:         print(book['isbn'])
49:         print(book['author'])
50:         print(book['title'])
51:         print()
52:
53: if __name__ == '__main__':
54:     _test()

```

pennyheroku/databasepostgres3.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # databasepostgres3.py
5: # Author: Bob Dondero
6: #-----
7:
8: import os
9: import psycopg
10: import psycopg_pool
11: import dotenv
12:
13: #-----
14:
15: dotenv.load_dotenv()
16: _DATABASE_URL = os.environ['DATABASE_URL']
17:
18: _connection_pool = psycopg_pool.ConnectionPool(_DATABASE_URL)
19:
20: #-----
21:
22: def get_books(author):
23:
24:     books = []
25:
26:     with _connection_pool.connection() as connection:
27:         with connection.cursor() as cursor:
28:             query_str = '''
29:                 SELECT isbn, author, title FROM books
30:                 WHERE author ILIKE %s
31:                 '''
32:             cursor.execute(query_str, [author+'%'])
33:             table = cursor.fetchall()
34:             for row in table:
35:                 book = {'isbn': row[0], 'author': row[1],
36:                       'title': row[2]}
37:                 books.append(book)
38:
39:     return books
40:
41: #-----
42:
43: # For testing:
44:
45: def _test():
46:     books = get_books('ker')
47:     for book in books:
48:         print(book['isbn'])
49:         print(book['author'])
50:         print(book['title'])
51:         print()
52:
53: if __name__ == '__main__':
54:     _test()

```

pennyheroku/database.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # database.py
5: # Author: Bob Dondero
6: #-----
7:
8: import os
9: import sqlalchemy
10: import sqlalchemy.orm
11: import dotenv
12:
13: #-----
14:
15: dotenv.load_dotenv()
16: _DATABASE_URL = os.getenv('DATABASE_URL', 'sqlite:///penny.sqlite')
17: _DATABASE_URL = _DATABASE_URL.replace('postgres://', 'postgresql://')
18: _DATABASE_URL = _DATABASE_URL.replace(
19:     'postgresql://', 'postgresql+psycopg://')
20:
21: #-----
22:
23: class Base(sqlalchemy.orm.DeclarativeBase):
24:     pass
25:
26: class Book(Base):
27:     __tablename__ = 'books'
28:     isbn = sqlalchemy.Column(sqlalchemy.String, primary_key=True)
29:     author = sqlalchemy.Column(sqlalchemy.String)
30:     title = sqlalchemy.Column(sqlalchemy.String)
31:
32: _engine = sqlalchemy.create_engine(_DATABASE_URL)
33:
34: #-----
35:
36: def get_books(author):
37:
38:     books = []
39:
40:     with sqlalchemy.orm.Session(_engine) as session:
41:
42:         query = session.query(Book).filter(
43:             Book.author.ilike(author+'%'))
44:         table = query.all()
45:         for row in table:
46:             book = {'isbn': row.isbn, 'author': row.author,
47:                   'title': row.title}
48:             books.append(book)
49:
50:     return books
51:
52: #-----
53:
54: # For testing:
55:
56: def _test():
57:     books = get_books('ker')
58:     for book in books:
59:         print(book['isbn'])
60:         print(book['author'])
61:         print(book['title'])
62:         print()
63:
64: if __name__ == '__main__':
65:     _test()

```