

```
File Edit Selection View Go Run Terminal Help
django

EXPLORER
djangoenv
  djangoenv
  djangoenv
  apps
    __pycache__
    migrations
    _init_.py
    admin.py
    apps.py
    models.py
    tests.py
    urls.py
    views.py
  templates
    bookmodule
    html5
    includes
    layouts
    aboutus.html
    book.html
    booklist.html
    books.html
    index.html
    list_books.html
    one_book.html
    search.html
    show.html
    usermodule

OUTLINE
TIMELINE

search.html A
book.html A
views.py R X
booklist.html U
db.sqlite3 M
MOCK_DATA.sql
settings.py U
mod >
...

djangoenv > djangoenv > apps > bookmodule > views.py > ...
80 return render(request, "bookmodule/booklist.html", {'books': new_books})"""
81
82 def filterbooks(request):
83     if request.method == "POST":
84         string = request.POST.get('keyword', '').lower()
85         is_title = request.POST.get('option1') is not None
86         is_author = request.POST.get('option2') is not None
87         # now filter
88         books = _getBooksList()
89         new_books = []
90         for item in books:
91             contained = False
92             if is_title and string in item['title'].lower():
93                 contained = True
94             if not contained and is_author and string in item['author'].lower():
95                 contained = True
96             if contained:
97                 new_books.append(item)
98         return render(request, "bookmodule/booklist.html", {'books': new_books})
99     else:
100         books = Book.objects.filter(title__contains='and')
101         if len(books) >= 1:
102             return render(request, "bookmodule/search.html", {'booksObj': books})
103         else:
104             return render(request, "bookmodule/list_books.html")
105
106 def simple_query(request):
107     mybooks = Book.objects.filter(title__icontains='and') # <- multiple objects
108     return render(request, "bookmodule/booklist.html", {'books': mybooks})
109
110 def complex_query(request):
111     mybooks = Book.objects.filter(title__icontains='and') # <- multiple objects
112     return render(request, "bookmodule/booklist.html", {'books': mybooks})
113
114 def complex_query(request):
115     mybooks = Book.objects.filter(
116         author__isnull=False,
117         title__icontains='and',
118         edition__gte=2,
119         price__gt=100)[:10]
120     if len(mybooks) >= 1:
121         return render(request, "bookmodule/booklist.html", {'books': mybooks})
122     else:
123         return render(request, "bookmodule/index.html")
124
125 def book(request):
126     if request.method == "POST":
127         title = request.POST.get('title')
128         author = request.POST.get('author')
129         edition = request.POST.get('edition')
130         price = request.POST.get('price')
131         mybook = Book.objects.create(title=title, author=author, price=price, edition=edition)
132         mybook.save()
133         #context = {'book': mybook}
134         return render(request, "bookmodule/book.html", context)
135
136 def list_books(request):
137     mybook = Book.objects.create(title='Continuous and Delivery', author='J. Humble and D. Farley', edition=1)
138     mybook.save()
139     context = {'book': mybook}
140     return render(request, "bookmodule/list_books.html", context)
141
142 def _getBooksList():
143     book1 = {'id': 12344321, 'title': 'Continuous Delivery', 'author': 'J. Humble and D. Farley'}
144     book2 = {'id': 56788765, 'title': 'Reversing: Secrets of Reverse Engineering', 'author': 'E. Eilam'}
145     book3 = {'id': 43211234, 'title': 'The Hundred-Page Machine Learning Book', 'author': 'Andriy Burkov'}
146     return [book1, book2, book3]
147
148 if request.method == "POST":
149     string = request.POST.get('keyword', '').lower()
150     is_title = request.POST.get('option1') is not None
```

```
File Edit Selection View Go Run Terminal Help
django

EXPLORER
djangoenv
  djangoenv
  djangoenv
  apps
    __pycache__
    migrations
    _init_.py
    admin.py
    apps.py
    models.py
    tests.py
    urls.py
    views.py
  templates
    bookmodule
    html5
    includes
    layouts
    aboutus.html
    book.html
    booklist.html
    books.html
    index.html
    list_books.html
    one_book.html
    search.html
    show.html
    usermodule

OUTLINE
TIMELINE

search.html A
book.html A
views.py R X
booklist.html U
db.sqlite3 M
MOCK_DATA.sql
settings.py U
mod >
...

djangoenv > djangoenv > apps > bookmodule > views.py > ...
107 def simple_query(request):
108     mybooks = Book.objects.filter(title__icontains='and') # <- multiple objects
109     return render(request, "bookmodule/booklist.html", {'books': mybooks})
110
111 def complex_query(request):
112     mybooks = Book.objects.filter(
113         author__isnull=False,
114         title__icontains='and',
115         edition__gte=2,
116         price__gt=100)[:10]
117     if len(mybooks) >= 1:
118         return render(request, "bookmodule/booklist.html", {'books': mybooks})
119     else:
120         return render(request, "bookmodule/index.html")
121
122 def book(request):
123     if request.method == "POST":
124         title = request.POST.get('title')
125         author = request.POST.get('author')
126         edition = request.POST.get('edition')
127         price = request.POST.get('price')
128         mybook = Book.objects.create(title=title, author=author, price=price, edition=edition)
129         mybook.save()
130         #context = {'book': mybook}
131         return render(request, "bookmodule/book.html", context)
132
133 def list_books(request):
134     mybook = Book.objects.create(title='Continuous and Delivery', author='J. Humble and D. Farley', edition=1)
135     mybook.save()
136     context = {'book': mybook}
137     return render(request, "bookmodule/list_books.html", context)
138
139 def _getBooksList():
140     book1 = {'id': 12344321, 'title': 'Continuous Delivery', 'author': 'J. Humble and D. Farley'}
141     book2 = {'id': 56788765, 'title': 'Reversing: Secrets of Reverse Engineering', 'author': 'E. Eilam'}
142     book3 = {'id': 43211234, 'title': 'The Hundred-Page Machine Learning Book', 'author': 'Andriy Burkov'}
143     return [book1, book2, book3]
144
145 if request.method == "POST":
146     string = request.POST.get('keyword', '').lower()
147     is_title = request.POST.get('option1') is not None
```

```
File Edit Selection View Go Run Terminal Help
django

EXPLORER
djangoenv
  djangoenv
  djangoenv
  apps
    __pycache__
    migrations
    _init_.py
    admin.py
    apps.py
    models.py
    tests.py
    urls.py
    views.py
  templates
    bookmodule
    html5
    includes
    layouts
    aboutus.html
    book.html
    booklist.html
    books.html
    index.html
    list_books.html
    one_book.html
    search.html
    show.html
    usermodule

OUTLINE
TIMELINE

search.html A
book.html A
views.py R X
booklist.html U
db.sqlite3 M
MOCK_DATA.sql
settings.py U
models.py M
cleanses >
...

djangoenv > djangoenv > apps > bookmodule > views.py > list_books
34 return render(request, "bookmodule/buybook.html")
35
36 def viewbook(request, bid):
37     # assume that we have the following books somewhere (e.g. database)
38     book1 = {'id': 123, 'title': 'Continuous Delivery', 'author': 'J. Humble and D. Farley'}
39     book2 = {'id': 567, 'title': 'Secrets of Reverse Engineering', 'author': 'E. Eilam'}
40     book3 = {'id': 456, 'title': 'Last Live', 'author': 'B. Xianzhao'}
41
42     targetBook = None
43     if book1['id'] == bid:
44         targetBook = book1
45     if book2['id'] == bid:
46         targetBook = book2
47     if book3['id'] == bid:
48         targetBook = book3
49     context = {'book': targetBook} # book is the variable name accessible by the template
50     return render(request, "bookmodule/book.html", context)
51
52 def list_books(request):
53     mybook = Book.objects.create(title='Continuous and Delivery', author='J. Humble and D. Farley', edition=1)
54     mybook.save()
55     context = {'book': mybook}
56     return render(request, "bookmodule/list_books.html", context)
57
58 def _getBooksList():
59     book1 = {'id': 12344321, 'title': 'Continuous Delivery', 'author': 'J. Humble and D. Farley'}
60     book2 = {'id': 56788765, 'title': 'Reversing: Secrets of Reverse Engineering', 'author': 'E. Eilam'}
61     book3 = {'id': 43211234, 'title': 'The Hundred-Page Machine Learning Book', 'author': 'Andriy Burkov'}
62     return [book1, book2, book3]
63
64 if request.method == "POST":
65     string = request.POST.get('keyword', '').lower()
66     is_title = request.POST.get('option1') is not None
```



