

```
python -m venv Django
```

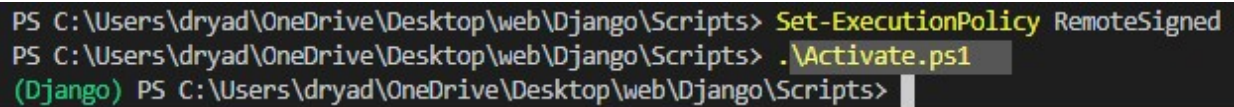
```
#source Django/bin/activate
```

```
Django\Scripts\activate.bat
```

In vscode

```
Set-ExecutionPolicy RemoteSigned #get the permission
```

```
Django\Scripts\Activate.ps1
```



```
PS C:\Users\dryad\OneDrive\Desktop\web\Django\Scripts> Set-ExecutionPolicy RemoteSigned
PS C:\Users\dryad\OneDrive\Desktop\web\Django\Scripts> .\Activate.ps1
(Django) PS C:\Users\dryad\OneDrive\Desktop\web\Django\Scripts> |
```

```
pip install Django
```

```
pip install DjangoRESTframework
```

```
django-admin.py startproject first .
```

```
python manage.py runserver
```

```
#server is on!
```

```
django-admin.py startapp demo
```

```
python .\manage.py migrate #set up database
```

```
set the models you want in demo/models.py
```

```
'''
```

```
class book(models.Model):
```

```
    title = models.CharField(max_length=36)
```

```
'''
```

```
regist the app in first/setting.py
```

```
'''
```

```
INSTALLED_APPS = [
```

```
'demo'
```

```
]
```

```
'''
```

```

first > settings.py > ...
27
28 ALLOWED_HOSTS = []
29
30
31 # Application definition
32 |
33 INSTALLED_APPS = [
34     'rest_framework',
35     'rest_framework.authtoken',
36     'demo',
37     'django.contrib.admin',
38     'django.contrib.auth',
39     'django.contrib.contenttypes',
40     'django.contrib.sessions',
41     'django.contrib.messages',
42     'django.contrib.staticfiles',
43
44 ]
45

```

python .\manage.py makemigrations #update database (migrations for 'demo')

in the demo/migrations folder you will the the new file 0001\_initial.py for the model setting

python .\manage.py migrate # migrate for 0001\_initial.py (apply the model to database)

models.py -> create model -> makemigrations -> create database XXX\_initial.py file -> migrate to apply the model to database

python .\manage.py createsuperuser # create the superuser for database

# assign the model to admin in admin.py

from .models import book

admin.site.register(book)

python manage.py runserver

http://127.0.0.1:8000/admin/

#the books is already in you site

**URL**

Add demo url to first

```
urls.py demo  urls.py first X
first > urls.py > ...
13     1. Import the include() function: from django.urls import include, path
14     2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
15     """
16     from django.contrib import admin
17     from django.urls import path, include
18
19     urlpatterns = [
20         path('demo/', include('demo.urls')),
21         path('admin/', admin.site.urls),
22     ]
23
```

creat urls.py in demo

add URL

```
urls.py demo X  views.py  manage.py
demo > urls.py > ...
1  from django.urls import path
2  from . import views
3  from .views import another
4
5  urlpatterns = [
6      path('',views.first),
7      path('another',another.as_view()),
8  ]
9
```

Add view

```
urls.py demo X views.py X manage.py models
demo > views.py > ...
1 from django.shortcuts import render
2 from django.http import HttpResponse
3 from django.views import View
4 from .models import book
5
6 # Create your views here.
7 class another(View):
8
9     books = book.objects.all()
10    output = f"we have {len(books)} in DB"
11    def get(self, request):
12        return HttpResponse(self.output)
13
14    def first(request):
15        return HttpResponse('first view')
```

//=====BUG===

if the book doesn't have objects

Press **ctr+sft+P** to open the the **Command Palette**. Now in command palette type **Preferences: Configure Language Specific Settings**. Now select **Python**. Here in right side paste this code

```
"python.linting.pylintArgs": [
    "--load-plugins=pylint_django",
]
```

```
urls.py demo views.py models.py {} settings.json X urls.py first
C: > Users > dryad > AppData > Roaming > Code > User > {} settings.json > [ ] python.linting.pylintArgs
1 {
2     "[python]": {
3
4     },
5     "python.linting.pylintArgs": [
6         "--load-plugins=pylint_django",
7     ]
8 }
```

//=====

Test result

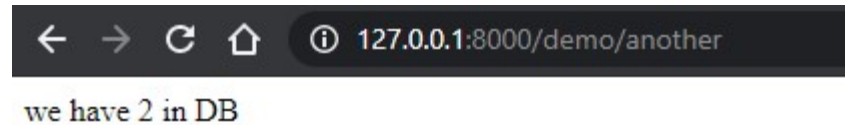


Table schema

```
class Book(models.Model):
    title = models.CharField(max_length=36, blank=False, unique=True)
    description = models.TextField(max_length=256, blank=True)

    price = models.DecimalField(default=0, max_digits=3, decimal_places=2)

    published = models.DateField(blank=True, null=True, default=None)
    is_published = models.BooleanField(default=False)

    cover = models.ImageField(upload_to='covers/', blank=True)
```

Use filter

```
books = Book.objects.filter(is_published=True)
output = ''

for book in books:
    output += f"We have {book.title} book with ID {book.id}<br>"

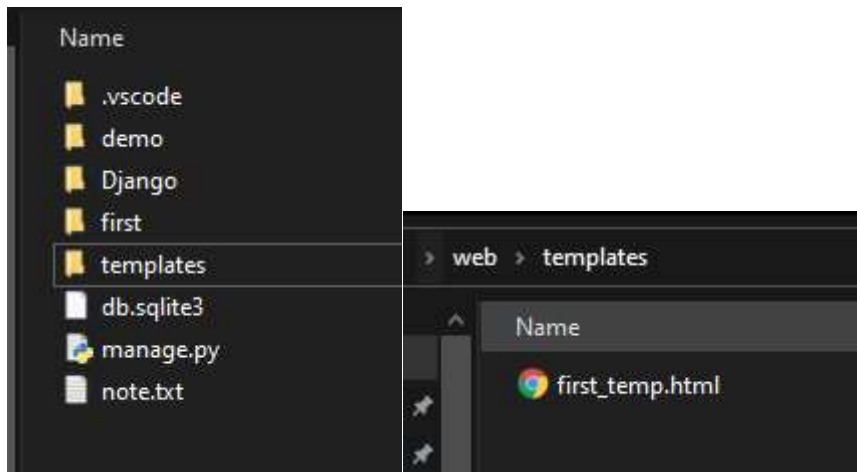
def get(self, request):
    return HttpResponse(self.output)
```

Get will one retrieve one record.

```
books = Book.objects.get(id=2)
```

Add template

Create template folder and html



Add to setting

```
terminal  Help  settings.py  web  visual studio code
urls.py demo  views.py  settings.py X  manage.py
first > settings.py > ...
52
53 ROOT_URLCONF = 'first.urls'
54
55 TEMPLATES = [
56     {
57         'BACKEND': 'django.template.backends.django.DjangoTemplates',
58         'DIRS': ['templates'],
59         'APP_DIRS': True,
60         'OPTIONS': {
61             'context_processors': [
62                 'django.template.context_processors.debug',
```

```
from django.shortcuts import render

# Create your views here.
class another(View):
    books = book.objects.all()
    output = f"we have {len(books)} in DB"
    def get(self, request):
        return HttpResponse(self.output)

def first(request):
    #return HttpResponse('first view')
    return render(request, 'first_temp.html')
```

```
demo > 📄 urls.py > ...
1  from django.urls import path
2  from . import views
3  from .views import another
4
5  urlpatterns = [
6      path('', views.first),
7      path('first', views.first),
8      path('another', another.as_view()),
9  ]
10
```

← → ↺ 🏠 ⓘ 127.0.0.1:8000/demo/first

# This is a Heading

This is a paragraph.

## Dynamic template

```
demo > 📄 views.py > 📄 first
5  from django.shortcuts import render
6
7  # Create your views here.
8  class another(View):
9
10     books = book.objects.all()
11     output = f"we have {len(books)} in DB"
12     def get(self, request):
13         return HttpResponse(self.output)
14
15     def first(request):
16         #return HttpResponse('first view')
17         return render(request, 'first_temp.html', {'data': 'this is a test data'})
18
```



templates > ≡ first\_temp.html

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>Page Title</title>
5  </head>
6  <body>
7  |
8  |   <h1>This is a Heading</h1>
9  |   <p>{{data}}</p>
10 |
11 </body>
12 </html>
13
```

← → ↻ 🏠 ⓘ 127.0.0.1:8000/demo/first

# This is a Heading

this is a test data

Deliver the obj

```
def first(request):
    books = book.objects.all()
    #return HttpResponse('first view')
    return render(request, 'first_temp.html', {'books': books})
```



```
<h1>This is a Heading</h1>
{% for book in books%}
    <p>{{book.title}}</p>
{% endfor %}
```

← → ↻ ⬆ ⓘ 127.0.0.1:8000/demo/first

# This is a Heading

OS

Network

Add If

```
{% for book in books %}
    {% if book.is_published %}
        <h2>{{ book.title }}</h2>
    {% endif %}
{% endfor %}
```

Admin customization

# Django administration

Home > Demo > Books

Select book to change

Action:  Go 0 of 2 selected

☐ BOOK

☐ book object (2)

☐ book object (1)

2 books

## Unreadable

```
class Book(models.Model):
    title = models.CharField(max_length=36, blank=False, unique=True)
    description = models.TextField(max_length=256, blank=True)

    price = models.DecimalField(default=0, max_digits=3, decimal_places=2)

    published = models.DateField(blank=True, null=True, default=None)
    is_published = models.BooleanField(default=False)

    cover = models.ImageField(upload_to='covers/', blank=True)

    def __str__(self):
        return self.title
```

# Django administration

Home › Demo › Books

Select book to change

Action:

☐ BOOK

☐ Network

☐ OS

2 books

Choose which to display

```
admin.py x admin.py x  
from django.contrib import admin  
from .models import Book  
  
@admin.register(Book)  
class BookAdmin(admin.ModelAdmin):  
    fields = ['title', 'description']
```

Change book

Title:

Hobbit

Description:



Delete

Save as

Choose which to display in list

```
@admin.register(Book)
class BookAdmin(admin.ModelAdmin):
    list_display = ['title', 'description']
```

Select book to change

Action:   0 of 2 selected

<input type="checkbox"/>	TITLE	DESCRIPTION
<input type="checkbox"/>	Lord of the ring	some description goes here
<input type="checkbox"/>	Hobbit	

2 books

filter

```
@admin.register(Book)
class BookAdmin(admin.ModelAdmin):
    list_display = ['title', 'price']
    list_filter = ['published']
```

Select book to change

Action:   0 of 2 selected

<input type="checkbox"/>	TITLE	PRICE
<input type="checkbox"/>	Lord of the ring	1.00
<input type="checkbox"/>	Hobbit	0.00

2 books

ADD BOOK +

**FILTER**

By published

- Any date
- Today
- Past 7 days
- This month
- This year
- No date
- Has date

Search

```
@admin.register(Book)
class BookAdmin(admin.ModelAdmin):
    list_display = ['title', 'price']
    list_filter = ['published']
    search_fields = ['title', 'description']
```

Select book to change

✓  Search

Action:  Go 0 of 2 selected

<input type="checkbox"/> TITLE	PRICE
<input type="checkbox"/> Lord of the ring	1.00
<input type="checkbox"/> Hobbit	0.00

2 books

ADD BOOK +

FILTER

By published

Any date

Today

Past 7 days

This month

This year

No date

Has date

## Use rest framework

Pip3 install djangoRESTframework

Create the serializers.py

```
demo > serializers.py > ...
1 from rest_framework import serializers
2 from .models import book
3
4 class BookSerializer(serializers.ModelSerializer):
5     class Meta:
6         model = book
7         fields = ['title']
```

```
demo > views.py > ...
1  # from django.shortcuts import render
2  # from django.http import HttpResponse
3  # from django.views import View
4  from .models import book
5  # from django.shortcuts import render
6  from rest_framework import viewsets
7  from .serializers import BookSerializer
8
9
10 # Create your views here.
11 class BookViewSet(viewsets.ModelViewSet):
12     serializer_class = BookSerializer
13     queryset = book.objects.all()
```

```
demo > urls.py > ...
1  from django.urls import path, include
2  #from . import views
3  #from .views import another
4  from rest_framework import routers
5  from .views import BookViewSet
6
7  routers = routers.DefaultRouter()
8  routers.register('books', BookViewSet)
9
10 urlpatterns = [
11     path('', include(routers.urls)),
12     #path('first', views.first),
13     #path('another', another.as_view()),
14 ]
```

## Django REST framework

Api Root

# Api Root

The default basic root view for DefaultRouter

GET /demo/

HTTP 200 OK

Allow: GET, HEAD, OPTIONS

Content-Type: application/json

Vary: Accept

```
{
  "books": "http://127.0.0.1:8000/demo/books/"
}
```



127.0.0.1:8000/demo/books/

Django REST framework

Api Root / Book List

# Book List

GET /demo/books/

HTTP 200 OK  
Allow: GET, POST, HEAD, OPTIONS  
Content-Type: application/json  
Vary: Accept

```
[
  {
    "title": "OS"
  },
  {
    "title": "Network"
  }
]
```

Add authToken

```

first > settings.py > ...
27
28     ALLOWED_HOSTS = []
29
30
31     # Application definition
32     |
33     INSTALLED_APPS = [
34         'rest_framework',
35         'rest_framework.authtoken',
36         'demo',
37         'django.contrib.admin',
38         'django.contrib.auth',
39         'django.contrib.contenttypes',
40         'django.contrib.sessions',
41         'django.contrib.messages',
42         'django.contrib.staticfiles',
43
44     ]

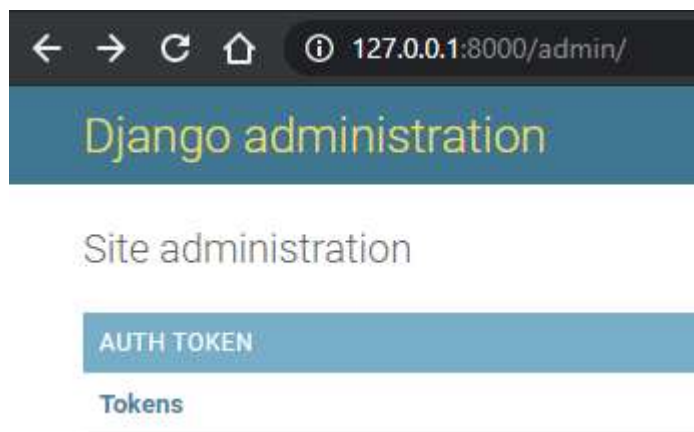
```

Migrate the change

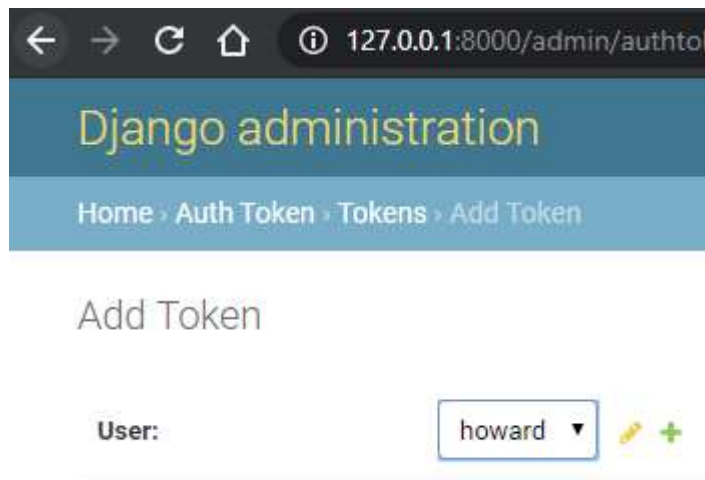
```

PS C:\Users\dryad\OneDrive\Desktop\web> python .\manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, authtoken, contenttypes, demo, sessions
Running migrations:
  Applying authtoken.0001_initial... OK
  Applying authtoken.0002_auto_20160226_1747... OK

```

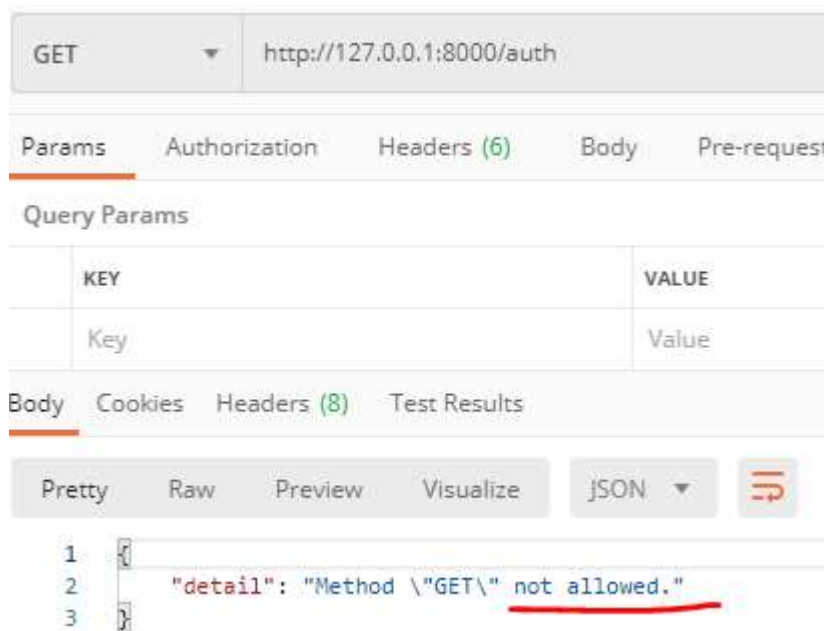


Add token to user



Test permission

```
first > urls.py > ...
17 from django.urls import path, include
18 from rest_framework.auth_token.views import obtain_auth_token
19
20 urlpatterns = [
21     path('demo/', include('demo.urls')),
22     path('admin/', admin.site.urls),
23     path('auth/', obtain_auth_token),
24 ]
25
```



POST http://127.0.0.1:8000/auth/

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

☐ none ☒ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

	KEY	VALUE	DESCRIPTI
<input checked="" type="checkbox"/>	username	howard	
<input checked="" type="checkbox"/>	password	0324	
	Key	Value	Descripti

Body Cookies (1) Headers (8) Test Results Status: 200 OK Time: 335

Pretty Raw Preview Visualize

```
{"token": "aaeb2889535a898097393b414bed7dcaa76fb11b"}
```

Add web permission setting

```
first > settings.py > ...
85
86
87 REST_FRAMEWORK = {
88     'DEFAULT_PERMISSION_CLASSES': [
89         'rest_framework.permissions.IsAuthenticated',
90     ]
91 }
92
```

Now we can't access the books without authentication

GET http://127.0.0.1:8000/demo/books/

Params Authorization Headers (7) Body Pre-request Script

Query Params

KEY	VALUE
Key	Value

Body Cookies (1) Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "detail": "Authentication credentials were not provided."
3 }
```

Set the authentication to view

```
demo > views.py > BookViewSet
4
5 from .models import book
6 from rest_framework import viewsets
7 from .serializers import BookSerializer
8 from rest_framework.authentication import TokenAuthentication
9
10
11 class BookViewSet(viewsets.ModelViewSet):
12     serializer_class = BookSerializer
13     queryset = book.objects.all()
14     authentication_classes = (TokenAuthentication,)
```

Post again with token

GET http://127.0.0.1:8000/demo/books/

Params Authorization Headers (8) Body Pre-request Script Tests Settings

Headers 7 hidden

KEY	VALUE
Authorization	Token aueb2889535a898097393b414bed7dcaa76fb11b
Key	Value

Body Cookies (1) Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "title": "OS"
4   },
5   {
6     "title": "Network"
7   }
8 ]
```

We can also set allow to any(any one can access without token)

```
REST_FRAMEWORK = {
    'DEFAULT_PERMISSION_CLASSES': (
        'rest_framework.permissions.AllowAny',
    )
}
```

But set the permission in the view (BookViewSet this specific view will need token)

```
from rest_framework.authentication import TokenAuthentication
from rest_framework.permissions import IsAuthenticated

class BookViewSet(viewsets.ModelViewSet):
    serializer_class = BookSerializer
    queryset = Book.objects.all()
    authentication_classes = (TokenAuthentication,)
    permission_classes = (IsAuthenticated,)
```

Create the data relation (one to one)



```

demo > models.py > ...
1  from django.db import models
2
3
4  class bookNumber(models.Model):
5      isbn = models.CharField(max_length=10, blank=True)
6
7  class book(models.Model):
8      title = models.CharField(max_length=36)
9      number = models.OneToOneField(bookNumber, null=True, blank=True, on_delete=models.CASCADE)
10     def __str__(self):
11         return self.title
12
13

```

Make migrations

```

PS C:\Users\dryad\OneDrive\Desktop\web> python .\manage.py makemigrations
Migrations for 'demo':
  demo\migrations\0002_auto_20200518_1838.py
    - Create model bookNumber
    - Add field number to book

```

Apply migrations

```

PS C:\Users\dryad\OneDrive\Desktop\web> python .\manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, authtoken, contenttypes, demo, sessions
Running migrations:
  Applying demo.0002_auto_20200518_1838... OK

```

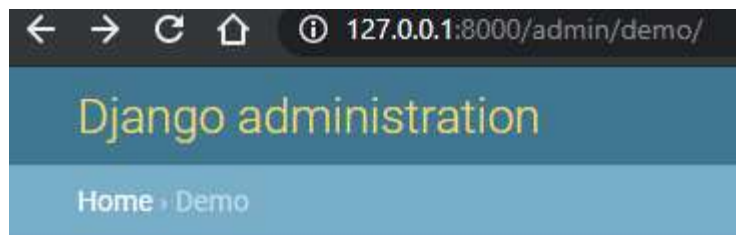
Assign the bookNumber to admin

```

demo > admin.py > bookAdmin
1  from django.contrib import admin
2  from .models import book, bookNumber
3
4  # Register your models here.
5  #admin.site.register(book)
6  @admin.register(book)
7  class bookAdmin(admin.ModelAdmin):
8      #fields = ['title']
9      list_display = ['title']
10     list_filter = ['title']
11     search_fields = ['title']
12  admin.site.register(bookNumber)

```





Demo administration



Change book


**Title:**

---

**Number:**   

Add number

Select book number to change

Action:    0 of 1 selected

<input type="checkbox"/>	BOOK NUMBER
<input type="checkbox"/>	bookNumber object (1)

---

1 book number

Add to serializer

```
demo > serializers.py > BookNumberSerializer > Meta
1 from rest_framework import serializers
2 from .models import book, bookNumber
3
4 class BookNumberSerializer(serializers.ModelSerializer):
5     class Meta:
6         model = bookNumber
7         fields = ['id', 'isbn']
8 class BookSerializer(serializers.ModelSerializer):
9     number = BookNumberSerializer(many = False)
10    class Meta:
11        model = book
12        fields = ['id', 'title', 'number']
```

GET http://127.0.0.1:8000/demo/books/

Params Authorization Headers (8) Body Pre-request

Headers 7 hidden

KEY
<input checked="" type="checkbox"/> Authorization
Key

Body Cookies (1) Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```

1 [
2   {
3     "id": 1,
4     "title": "OS",
5     "number": null
6   },
7   {
8     "id": 2,
9     "title": "Network",
10    "number": {
11      "id": 1,
12      "isbn": "1111111111"
13    }
14  }
15 ]
```

## One to many

```
demo > models.py > ...
1  from django.db import models
2
3
4  class bookNumber(models.Model):
5      isbn = models.CharField(max_length=10, blank=True)
6
7  class book(models.Model):
8      title = models.CharField(max_length=36)
9      number = models.OneToOneField(bookNumber, null=True, blank=True, on_delete=models.CASCADE)
10     def __str__(self):
11         return self.title
12
13     class char(models.Model):
14         name = models.CharField(max_length=30)
15         book = models.ForeignKey(book, on_delete=models.CASCADE, related_name='char')
16
```

## migrate

```
PS C:\Users\dryad\OneDrive\Desktop\web> python .\manage.py makemigrations
Migrations for 'demo':
  demo\migrations\0003_char.py
    - Create model char
PS C:\Users\dryad\OneDrive\Desktop\web> python .\manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, authtoken, contenttypes, demo, sessions
Running migrations:
  Applying demo.0003_char... OK
```

```
demo > admin.py > ...
1  from django.contrib import admin
2  from .models import book, bookNumber, char
3
4  # Register your models here.
5  #admin.site.register(book)
6  @admin.register(book)
7  class bookAdmin(admin.ModelAdmin):
8      #fields = ['title']
9      list_display = ['title']
10     list_filter = ['title']
11     search_fields = ['title']
12     admin.site.register(bookNumber)
13     admin.site.register(char)
```

# Django administration

Home > Demo > Chars > Add char

## Add char

Name:

Howard

Book:

OS



GET http://127.0.0.1:8000/demo/books/

Params Authorization Headers (8) Body Pre-request Script

Headers 7 hidden

KEY
<input checked="" type="checkbox"/> Authorization
Key

Body Cookies (1) Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "id": 1,
4     "title": "OS",
5     "number": null,
6     "char": [
7       {
8         "id": 1,
9         "name": "Howard"
10      },
11      {
12        "id": 2,
13        "name": "Maeve"
14      }
15    ]
16  },
17  {
18    "id": 2,
19    "title": "Network",
20    "number": {
21      "id": 1,
22      "isbn": "1111111111"
23    },
24    "char": []
25  }
26 ]
```

Many to many

```

demo > models.py > author
1  from django.db import models
2
3
4  class bookNumber(models.Model):
5      isbn = models.CharField(max_length=10, blank=True)
6
7  class book(models.Model):
8      title = models.CharField(max_length=36)
9      number = models.OneToOneField(bookNumber, null=True, blank=True, on_delete=models.CASCADE)
10     def __str__(self):
11         return self.title
12
13     class char(models.Model):
14         name = models.CharField(max_length=30)
15         book = models.ForeignKey(book, on_delete=models.CASCADE, related_name='char')
16
17     class author(models.Model):
18         name = models.CharField(max_length=30)
19         book = models.ManyToManyField(book, related_name='author')

```

```

demo > admin.py > ...
1  from django.contrib import admin
2  from .models import book, bookNumber, char, author
3
4  # Register your models here.
5  #admin.site.register(book)
6  @admin.register(book)
7  class bookAdmin(admin.ModelAdmin):
8      #fields = ['title']
9      list_display = ['title']
10     list_filter = ['title']
11     search_fields = ['title']
12
13     admin.site.register(bookNumber)
14     admin.site.register(char)
15     admin.site.register(author)

```

```

PS C:\Users\dryad\OneDrive\Desktop\web> python .\manage.py makemigrations
Migrations for 'demo':
  demo\migrations\0004_auto_20200518_1953.py
    - Alter field book on char
    - Create model author
PS C:\Users\dryad\OneDrive\Desktop\web> python .\manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, authtoken, contenttypes, demo, sessions
Running migrations:
  Applying demo.0004_auto_20200518_1953... OK

```



```

demo > serializers.py > BookSerializer > Meta
1  from rest_framework import serializers
2  from .models import book, bookNumber, char, author
3
4
5  class authorSerrializer(serializers.ModelSerializer):
6      class Meta:
7          model = author
8          fields = ['id', 'name']
9  class CharSerrializer(serializers.ModelSerializer):
10     class Meta:
11         model = char
12         fields = ['id', 'name']
13 class BookNumberSerrializer(serializers.ModelSerializer):
14     class Meta:
15         model = bookNumber
16         fields = ['id', 'isbn']
17 class BookSerrializer(serializers.ModelSerializer):
18     number = BookNumberSerrializer(many=False)
19     char = CharSerrializer(many=True)
20     author = authorSerrializer(many=True)
21     class Meta:
22         model = book
23         fields = ['id', 'title', 'number', 'char', 'author']
24

```



# Django administration

Home › Demo › Authors › Add author

## Add author

Name:

Howard

Book:



Hold down "Control", or "Command" on a Mac, to select more than one.

```
GET http://127.0.0.1:8000/demo/books/

Body Cookies (1) Headers (8) Test Results

Pretty Raw Preview Visualize JSON

1 [
2   {
3     "id": 1,
4     "title": "OS",
5     "number": null,
6     "char": [
7       {
8         "id": 1,
9         "name": "Howard"
10      },
11      {
12        "id": 2,
13        "name": "Maeve"
14      }
15    ],
16    "author": [
17      {
18        "id": 1,
19        "name": "Howard"
20      }
21    ],
22  },
23  {
24    "id": 2,
25    "title": "Network",
26    "number": {
27      "id": 1,
28      "isbn": "1111111111"
29    },
30    "char": [],
31    "author": [
32      {
33        "id": 1,
34        "name": "Howard"
35      }
36    ]
37  }
]
```

Share the API with port to React

pip install wheel

pip install django-cors-headers

```
=INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'rest_framework',  
    'rest_framework.authtoken',  
    'api',  
    ✓ 'corsheaders',  
]  
✓ CORS_ORIGIN_WHITELIST = [  
    'http://localhost:3000',  
    'http://localhost:8000',  
    'http://localhost:8080',  
]  
=MIDDLEWARE = [  
    'django.middleware.security.SecurityMiddleware',  
    'django.contrib.sessions.middleware.SessionMiddleware',  
    'django.middleware.common.CommonMiddleware',  
    'django.middleware.csrf.CsrfViewMiddleware',  
    'django.contrib.auth.middleware.AuthenticationMiddleware',  
    'django.contrib.messages.middleware.MessageMiddleware',  
    ✓ 'django.middleware.clickjacking.XFrameOptionsMiddleware',  
    ✓ 'corsheaders.middleware.CorsMiddleware',  
]
```