

[https://www.youtube.com/watch?v=Pwwz4\\_AvH DU&t=1080s](https://www.youtube.com/watch?v=Pwwz4_AvH DU&t=1080s)

<https://www.digitalocean.com/community/tutorials/build-a-to-do-application-using-django-and-react>

## Python Django + PostgreSQL | REST API +React Tutorial

1. First create folder
2. pip install virtualenvwrapper-win
3. mkvirtualenv apiwork
4. pip install django
5. django-admin startproject firstproject
6. pip install djangorestframework
7. pip install django-cors-headers
8. Inside project
9. Python manage.py startapp apiapp
10. Now register the module in register.py in project

```
11. INSTALLED_APPS = [  
12.     'django.contrib.admin',  
13.     'django.contrib.auth',  
14.     'django.contrib.contenttypes',  
15.     'django.contrib.sessions',  
16.     'django.contrib.messages',  
17.     'django.contrib.staticfiles',  
18.     'rest_framework',  
19.     'apiwork',  
20.     'corsheaders'  
21. ]  
22.  
23. CORS_ORIGIN_ALLOW_ALL = True  
24.  
25. MIDDLEWARE = [  
26.     "corsheaders.middleware.CorsMiddleware",  
27.     "django.middleware.common.CommonMiddleware",  
28.     'django.middleware.security.SecurityMiddleware',  
29.     'django.contrib.sessions.middleware.SessionMiddleware',  
30.     'django.middleware.common.CommonMiddleware',  
31.     'django.middleware.csrf.CsrfViewMiddleware',  
32.     'django.contrib.auth.middleware.AuthenticationMiddleware',  
33.     'django.contrib.messages.middleware.MessageMiddleware',  
34.     'django.middleware.clickjacking.XFrameOptionsMiddleware',  
35. ]  
36.  
37. ROOT_URLCONF = 'frontend.urls'  
38.  
39. TEMPLATES = [  
40.     {  
41.         'BACKEND': 'django.template.backends.django.DjangoTemplates',  
42.         'DIRS': [],  
43.         'APP_DIRS': True,  
44.         'OPTIONS': {  
45.             'context_processors': [  
46.                 'django.template.context_processors.debug',  
47.                 'django.template.context_processors.request',  
48.                 'django.contrib.auth.context_processors.auth',  
49.                 'django.contrib.messages.context_processors.messages',  
50.             ],  
51.         },  
52.     },  
53. ]  
54.  
55. WSGI_APPLICATION = 'firstproject.wsgi.application'  
56.  
57. # Internationalization  
58. # https://docs.djangoproject.com/en/2.2/topics/i18n/
```

```

40.     {
41.         'BACKEND': 'django.template.backends.django.DjangoTemplates',
42.         'DIRS': [],
43.         'APP_DIRS': True,
44.         'OPTIONS': {
45.             'context_processors': [
46.                 'django.template.context_processors.debug',
47.                 'django.template.context_processors.request',
48.                 'django.contrib.auth.context_processors.auth',
49.                 'django.contrib.messages.context_processors.messages',
50.             ],
51.         },
52.     },
53. ]
54.
55. WSGI_APPLICATION = 'frontend.wsgi.application'
56.
57. # Database
58. # https://docs.djangoproject.com/en/3.2/ref/settings/#databases
59.
60. DATABASES = {
61.     'default': {
62.         'ENGINE': 'django.db.backends.postgresql',
63.         'NAME': 'reactdjango',
64.         'USER': 'postgres',
65.         'PASSWORD': 'saikat',
66.         'HOST': 'localhost',
67.         'PORT': '5432'
68.     }
69. }
70. }
71.
72. # Password validation
73. # https://docs.djangoproject.com/en/3.2/ref/settings/#auth-password-
74. # validators
75. AUTH_PASSWORD_VALIDATORS = [
76.     {
77.         'NAME':
78.             'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
79.     },
80.     {
81.         'NAME':
82.             'django.contrib.auth.password_validation.MinimumLengthValidator',

```

```

83.         'NAME':
      'django.contrib.auth.password_validation.CommonPasswordValidator',
84.     },
85.     {
86.         'NAME':
      'django.contrib.auth.password_validation.NumericPasswordValidator',
87.     },
88. ]
89.
90. # Url Address
91. CORS_ORIGIN_WHITELIST = [
92.     'http://localhost:3000'
93. ]

```

11. Create models.py from app

```

from django.db import models

# Create your models here.

class Employee(models.Model):
    full_name = models.CharField(max_length=200)
    email = models.EmailField(max_length=100)
    contact = models.CharField(max_length=500)
    address = models.TextField()

    def __str__(self):
        return self.full_name

```

12. Data base adapter install

-> pip install psycopg2

13. Create database now.

12. Database settings.py from project

```

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'apidb',
        'USER': 'postgres',
        'PASSWORD': 'saikat',
        'HOST': 'localhost',
        'PORT': '5432'
    }
}

```

14. python manage.py makemigrations

15. python manage.py migrate

16. Set in admin.py in app

```
from django.contrib import admin
from .models import Employee

# Register your models here.
admin.site.register(Employee)
```

17. create serializers.py file into app

```
from rest_framework import serializers
from .models import Employee

class EmployeeSerializer(serializers.ModelSerializer):
    class Meta:
        model = Employee
        fields = ('id', 'full_name', 'email', 'contact', 'address')
```

18. Create API methods in views.py from app

```
from django.shortcuts import render
from rest_framework import viewsets
from .serializers import EmployeeSerializer
from .models import Employee

# Create your views here.

class EmployeeView(viewsets.ModelViewSet):
    serializer_class = EmployeeSerializer
    queryset = Employee.objects.all()
```

19. set urls.py in project

```
from django.urls import path, include
from django.contrib import admin
from django.urls import path, include
from rest_framework import routers
from apiwork import views

router = routers.DefaultRouter()
router.register(r'employees', views.EmployeeView, 'employee')

urlpatterns = [
    path('admin/', admin.site.urls),
```

```
path('api/', include(router.urls)),  
]
```

20. run the server

- `-.> python manage.py runserver`

21. API test from postman

Now Start With React Part:

1. Node.js download
2. Go to the project directory create a folder
3. Inside the new folder create react app  
-> `npx create-react-app frontend`
4. Cd frontend
5. Npm start
6. From the bootstrap site download bootstrap.min.css file and include into src folder.
7. Create a components folder into src folder. Add three components Add, Update and List.  
App.js file update here:

App.js:

```
import './App.css';  
import React from 'react';  
import './bootstrap.min.css';  
  
import List from './components/List';  
import Add from './components/Add';  
import Update from './components/Update';  
  
//import {Switch,Route,Link} from 'react-router-dom';  
import { BrowserRouter,Routes ,Route , Link,Switch} from 'react-router-dom';  
//import { BrowserRouter as Router,Routes, Route, Link } from "react-router-dom";  
  
function App() {  
  return (  
    <BrowserRouter>  
  
    <div className="container">  
      <nav className="navbar navbar-expand-lg navbar-dark bg-dark mb-3">  
        <Link className="navbar-brand" to="/">Navbar</Link>  
        <button className="navbar-toggler" type="button" data-  
toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-  
expanded="false" aria-label="Toggle navigation">
```

```

        <span className="navbar-toggler-icon"></span>
      </button>
      <div className="collapse navbar-collapse" id="navbarNav">
        <ul className="navbar-nav">
          <li className="nav-item">
            <Link className="nav-link" to="/">List</Link>
          </li>
          <li className="nav-item">
            <Link className="nav-link" to="/add">Add</Link>
          </li>
        </ul>
      </div>
    </nav>

    <Switch>
      <Route path="/" component={List} exact />
      <Route path="/add" component={Add} />
      <Route path="/update/:id" component={Update} />
    </Switch>

  </div>
</BrowserRouter>
);
}

export default App;

```

List.js:

```

import React from 'react';
import { Link, Route } from 'react-router-dom';
import Update from './Update';

class List extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      data: []
    };
  }

  fetchData() {
    fetch('http://127.0.0.1:8000/api/employees/')
      .then(response => response.json())

```

```

        .then((data)=>{
            this.setState({
                data: data
            });
        });
    }

    deleteData(id){
        fetch('http://127.0.0.1:8000/api/employees/'+id+',',{
            method: 'DELETE',
            body: JSON.stringify(this.state),
        })
        .then(response=>response)
        .then((data)=>{
            if(data){
                this.fetchData();
            }
        });
    }

    sendValue(id){
        this.history.push({
            pathname: 'update/',
            search: '?query=Update',
            state: { id: id }
        })
    }

    componentDidMount(){
        this.fetchData();
    }

    render(){
        const empData=this.state.data;
        console.log(empData)
        const rows=empData.map((emp)=>
            // your link creation

            <tr key={emp.id}>
                <td>{emp.full_name}</td>
                <td>{emp.email}</td>
                <td>{emp.contact}</td>
                <td>{emp.address}</td>
                <td>
                    <Link to={'update/'+emp.id} className="btn btn-info mr-2">Update</Link>
                    <button onClick={()=>this.deleteData(emp.id)}
                        className="btn btn-danger">Delete</button>

```

```

        </td>
      </tr>
    );
    return (
      <table className="table table-bordered">
        <thead>
          <tr>
            <th>Full Name</th>
            <th>Email</th>
            <th>Contact</th>
            <th>Address</th>
            <th>Action</th>
          </tr>
        </thead>
        <tbody>
          {rows}
        </tbody>
      </table>
    );
  }
}

export default List;

```

Add.js:

```

import React from 'react';

class Add extends React.Component{

  constructor(){
    super();
    this.state={
      full_name:'',
      email:'',
      contact:'',
      address:''
    }
    this.handleChange=this.handleChange.bind(this);
    this.submitForm=this.submitForm.bind(this);
  }

  // Input Change Handler
  handleChange(event){
    this.setState({
      [event.target.name]:event.target.value
    });
  }

```



```

}

// Submit Form
submitForm(){
  fetch('http://127.0.0.1:8000/api/employees/',{
    method: 'POST',
    body: JSON.stringify(this.state),
    headers:{
      'Content-type': 'application/json; charset=UTF-8',
    },
  })
  .then(response=>response.json())
  .then(data=>console.log(data));

  this.setState({
    full_name:'',
    email:'',
    contact:'',
    address:''
  });
}

render(){
  return (
    <table className="table table-bordered">
      <tbody>
        <tr>
          <th>Full Name</th>
          <td>
            <input value={this.state.full_name}
name="full_name" onChange={this.handleChange} type="text" className="form-control" />
          </td>
        </tr>
        <tr>
          <th>Email</th>
          <td>
            <input value={this.state.email} name="email"
onChange={this.handleChange} type="text" className="form-control" />
          </td>
        </tr>
        <tr>
          <th>Contact</th>
          <td>
            <input value={this.state.contact} name="contact"
onChange={this.handleChange} type="text" className="form-control" />
          </td>
        </tr>
      </tbody>
    </table>
  )
}

```

```

        <th>Address</th>
        <td>
            <input value={this.state.address} name="address"
onChange={this.changeHandler} type="text" className="form-control" />
        </td>
    </tr>
    <tr>
        <td colspan="2">
            <input type="submit" onClick={this.submitForm}
className="btn btn-dark" />
        </td>
    </tr>
</tbody>
</table>
    );
}
}

export default Add;

```

Update.js:

```

import React from 'react';

class Update extends React.Component{

    constructor(props){
        super(props);
        this.state={
            full_name:'',
            email:'',
            contact:'',
            address:''
        }
        this.changeHandler=this.changeHandler.bind(this);
        this.submitForm=this.submitForm.bind(this);
    }

    // Input Change Handler
    changeHandler(event){
        this.setState({
            [event.target.name]:event.target.value
        });
    }

    // Submit Form
    submitForm(){

```

```

    var id=this.props.match.params.id;
    fetch('http://127.0.0.1:8000/api/employees/'+id+',',{
      method:'PUT',
      body:JSON.stringify(this.state),
      headers:{
        'Content-type': 'application/json; charset=UTF-8',
      },
    })
    .then(response=>response.json())
    .then((data)=>console.log(data));

    this.setState({full_name: '', email: '', contact: '',address:''})
  }

  fetchData(){

    var id=this.props.match.params.id;
    fetch('http://127.0.0.1:8000/api/employees/'+id)
    .then(response=>response.json())
    .then((data)=>{
      this.setState({
        full_name:data.full_name,
        email:data.email,
        contact:data.contact,
        address:data.address
      });
    });
  }

  componentDidMount(){
    //const id = this.props.match.params.id;
    //var id=this.props.match.params.id;
    //console.log("received "+id);
    this.fetchData();
  }

  render(){
    return (

      <table className="table table-bordered">
        <tbody>
          <tr>
            <th>Full Name</th>
            <td>

```

```

        <input value={this.state.full_name}
name="full_name" onChange={this.changeHandler} type="text" className="form-
control" />
    </td>
</tr>
<tr>
    <th>Email</th>
    <td>
        <input value={this.state.email} name="email"
onChange={this.changeHandler} type="text" className="form-control" />
    </td>
</tr>
<tr>
    <th>Contact</th>
    <td>
        <input value={this.state.contact} name="contact"
onChange={this.changeHandler} type="text" className="form-control" />
    </td>
</tr>
<tr>
    <th>Address</th>
    <td>
        <input value={this.state.address} name="address"
onChange={this.changeHandler} type="text" className="form-control" />
    </td>
</tr>
<tr>
    <td colspan="2">
        <input type="submit" onClick={this.submitForm}
className="btn btn-dark" />
    </td>
</tr>
</tbody>
</table>

    );
}
}

export default Update;

```

8. Now install react-router-dom  
Lastest version(Not use):

Install:

->npm install react-router-dom --save

Uninstall:

->npm uninstall react-router-dom

Install stable version(This one):

Install:

➔ npm install [react-router-dom@5.2.0](#)

9. run the server : npm start