Training Report

on

Python Training/ Trimp

Submitted in partial fulfillment of the requirements
For the award of the degree of

BACHELOR OF TECHNOLOGY

In

BTECH IT

Submitted By
Parkash
2818397

2018-2022

Panipat Institute of Engineering & Technology,
(Affiliated to Kurukshetra University Kurukshetra, India)
Samalkha, Panipat

2018-2022

CANDIDATE’S DECLARATION

I hereby declare that the work presented in this training report entitled **“Python Training /Trimp”**, submitted in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in **BTech IT**, submitted to **Panipat Institute of Engineering & Technology, Kurukshetra University, Kurukshetra, India** is an authentic record of my own work carried out during the period from **May 2021** to **July 2021** under the guidance of **Payal Chhabra**. The work reported in this project report has not been submitted by me for the award of any other degree or diploma.

Date: 06/12/21 PARKASH

Place: PIET 2818397

CERTIFICATE

This is to certify that the summer training report entitled **"Python Training /Trimp"** done by **Parkash, 2818397. Trimp** is an authentic work carried out by him at **PIET** under my guidance. The matter embodied in this **Python Training /Trimp** work has not been submitted earlier for the award of any degree or diploma to the best of my knowledge and belief.

Date: 06/12/21 **Payal Chhabra**

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech. Project undertaken during B. Tech. Final Year. We/I owe special debt of gratitude to Assistant Professor Mr. Sourabh Gupta, Department of Information Technology, PIET, Samalkha for his/her constant support and guidance throughout the course of our work. His/ Her sincerity, thoroughness and perseverance have been a constant source of inspiration for us/me. It is only his cognizant efforts that our/my endeavors have seen light of the day.

We/I also take the opportunity to acknowledge the contribution of Mr. Rattan Deep Aneja, Head, Department of Information Technology, PIET, Samalkha for his full support and assistance during the development of the project.

We/I also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation during the development of our/my project. Last but not the least, we/I acknowledge our friends for their contribution in the completion of the project.

ABSTRACT

Trimp is online tool for URL shortener. It uses react for frontend and Django for backend. I have created a complete documentation for using both for personal and commercial use. It is free to use, and you can use it for your projects but for commercial use you must purchase License. It uses **swagger API**, REDOC.

Redoc is an open-source tool for generating documentation from **OpenAPI** (fka Swagger) definitions.

By default, **Redoc** offers a three-panel, responsive layout:

The left panel contains a search bar and navigation menu.

The central panel contains the documentation.

The right panel contains request and response examples.

INTRODUCTION

Trimp is online tool for URL shortener. It uses react for frontend and Django for backend. I have created a complete documentation for using both for personal and commercial use. It is free to use, and you can use it for your projects but for commercial use you must purchase License. It uses swagger API, REDOC.

Redoc is an open-source tool for generating documentation from **OpenAPI** (fka Swagger) definitions.
By default, **Redoc** offers a three-panel, responsive layout:
The left panel contains a search bar and navigation menu.
The central panel contains the documentation.
The right panel contains request and response examples.

Swagger™ is a project used to describe and document RESTful APIs.
The Swagger specification defines a set of files required to describe such an API. These files can then be used by the Swagger-UI project to display the API and Swagger-**Codegen** to generate clients in various languages. Additional utilities can also take advantage of the resulting files, such as testing tools.

SOFTWARE AND HARDWARE REQUIREMENTS

Min requirements: - 2GB Ram, 32-bit operating system, 1GHz Processor, 128mb video memory

Software and Tools:

React:

React is a free and open-source front-end JavaScript library for building user interfaces based on UI components. We used React for creating frontend of our store.

Redux:

Redux is an open-source JavaScript library for managing and centralizing application state. It is most used with libraries such as React or Angular for building user interfaces. We have used it for managing our state and account details at client side.

Django:

Django is a Python-based free and open-source web framework that follows the model–template–views architectural pattern. We have used it for creating and managing backend of our app. This is also responsible for managing static and media files.

Django REST framework:

Django REST framework is a powerful and flexible toolkit for building Web APIs. Some reasons you might want to use REST framework: The Web browsable API is a huge usability win for your developers. Authentication policies including packages for OAuth1a and OAuth2. We used it for creating our API end points for easily communicate with our frontend.

CODING/SOURCE CODE

BACKEND:

URLS:

```
from django.contrib import admin
from django.urls import path, include
from django.conf import settings
from django.conf.urls.static import static
from django.urls.conf import re_path
from django.views.generic.base import RedirectView
from api.views import Redirector
from rest_framework import permissions
from drf_yasg.views import get_schema_view
from drf_yasg import openapi
from django.views.generic.base import TemplateView

schema_view = get_schema_view(
    openapi.Info(
        title="Trimp API",
        default_version='v1',
        description="Trimp is online tool for url
shortening. You can use it for your projects but for
comercial use you have to purchase License, For purchase
license you can contact us at
parkashsatiyaar0008@gmail.com",

terms_of_service="https://www.google.com/policies/terms/",

contact=openapi.Contact(email="parkashsatiyaar0008@gmail.co
m"),
        license=openapi.License(name="BSD License"),
    ),
    public=True,
    permission_classes=(permissions.AllowAny,),
)

urlpatterns = [
    re_path(r'^swagger(?P<format>\.json|\.yaml)$',
            schema_view.without_ui(cache_timeout=0),
name='schema-json'),
    re_path(r'^swagger/$', schema_view.with_ui('swagger',
            cache_timeout=0), name='schema-swagger-ui'),
    re_path(r'^redoc/$', schema_view.with_ui('redoc',
            cache_timeout=0), name='schema-redoc'),
    path('api-auth/', include('rest_framework.urls')),
    path('admin/', admin.site.urls),
    path('api/', include('api.urls')),
    path('<str:short_char>/', Redirector.as_view(),
name="redirector"),
] + static(settings.STATIC_URL,
document_root=settings.STATIC_ROOT) +
static(settings.MEDIA_URL,
document_root=settings.MEDIA_ROOT)

urlpatterns.append(path('', TemplateView.as_view(
    template_name="index.html"), name="home"))
urlpatterns.append(re_path('^.*',
RedirectView.as_view(url=f"/swagger/")))
```

API/URLS:

```
from rest_framework.routers import DefaultRouter
from .views import ShortenerListAPIView
router = DefaultRouter()
router.register('', ShortenerListAPIView)
urlpatterns = router.urls
```

VIEWS:

```
from django.shortcuts import redirect
from rest_framework.viewsets import ModelViewSet
from rest_framework.views import View
from rest_framework.permissions import AllowAny
from api.models import Urls
from api.serializer import UrlSerializer
```

```python
# Create your views here.


class ShortenerListAPIView(ModelViewSet):
    queryset = Urls.objects.all()
    serializer_class = UrlSerializer
    permission_classes = [AllowAny]


class Redirector(View):
    def get(self, request, short_char, *args, **kwargs):
        if not Urls.objects.filter(
                short_char=short_char).exists():
            return redirect('home')
        redirect_link = Urls.objects.filter(
            short_char=short_char).first().original_url
        return redirect(redirect_link)
```

Models:

```python
import string
import random
from django.db import models

# Create your models here.


class Urls(models.Model):
    original_url = models.URLField(max_length=2000)
    short_char = models.CharField(max_length=6, blank=True,
null=True)

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

```
def shortner(self):
        while True:
            ran_string =
''.join(random.choices(string.ascii_uppercase +

string.digits, k=6))
            if not
Urls.objects.filter(short_char=ran_string).exists():
                return ran_string

    def save(self, *args, **kwargs):
        short_string = self.shortner()
        self.short_char = short_string
        super(Urls, self).save(*args, **kwargs)
```

SERIALIZERS:

```
from rest_framework.serializers import ModelSerializer
from .models import Urls


class UrlSerializer(ModelSerializer):
    class Meta:
        model = Urls
        fields = "__all__"
```

SETTINGS:

```
import os
from pathlib import Path

# Build paths inside the project like this: BASE_DIR /
'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent
```

```python
# Quick-start development settings - unsuitable for
production
# See
https://docs.djangoproject.com/en/3.2/howto/deployment/chec
klist/

# SECURITY WARNING: keep the secret key used in production
secret!
SECRET_KEY = 'django-insecure-ky%u1fbb+$iunt(9oywoeafv%-
!xnolx(kv!e+ofpit-%f=b#j'

# SECURITY WARNING: don't run with debug turned on in
production!
DEBUG = True

ALLOWED_HOSTS = ['*']


# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'drf_yasg',
    'corsheaders',
    'rest_framework',
    'api',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'corsheaders.middleware.CorsMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',

'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'backend.urls'

TEMPLATES = [
    {
        'BACKEND':
'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR,
"templates/build")],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',

'django.template.context_processors.request',

'django.contrib.auth.context_processors.auth',

'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'backend.wsgi.application'


# Database
#
https://docs.djangoproject.com/en/3.2/ref/settings/#databas
es

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}


# Password validation
# https://docs.djangoproject.com/en/3.2/ref/settings/#auth-
password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
'django.contrib.auth.password_validation.UserAttributeSimil
arityValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.MinimumLengthValid
ator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.CommonPasswordVali
dator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.NumericPasswordVal
idator',
    },
]


# Internationalization
# https://docs.djangoproject.com/en/3.2/topics/i18n/

LANGUAGE_CODE = 'en-us'
```

```
TIME_ZONE = 'UTC'

USE_I18N = True

USE_L10N = True

USE_TZ = True


# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/3.2/howto/static-files/

STATIC_URL = '/static/'
STATIC_ROOT = os.path.join(BASE_DIR, 'static_media')

STATICFILES_DIRS = [
    os.path.join(BASE_DIR, "static"),
    os.path.join(BASE_DIR, "templates/build/static"),
]


# media root

MEDIA_ROOT = os.path.join(BASE_DIR, 'media')

MEDIA_URL = '/media/'

# Default primary key field type
#
https://docs.djangoproject.com/en/3.2/ref/settings/#default
-auto-field

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'


REST_FRAMEWORK = {
    # Use Django's standard `django.contrib.auth`
permissions,
    # or allow read-only access for unauthenticated users.
    'DEFAULT_PERMISSION_CLASSES': [

'rest_framework.permissions.DjangoModelPermissionsOrAnonRea
dOnly'
    ]
}

# CORS_ORIGIN_WHITELIST = (
#      'http://localhost:3000',
# )

CORS_ORIGIN_ALLOW_ALL = True
```

REQUIREMNETS:

```
asgiref==3.4.1
autopep8==1.6.0
certifi==2021.10.8
charset-normalizer==2.0.9
coreapi==2.3.3
coreschema==0.0.4
Django==3.2.9
django-cors-headers==3.10.0
djangorestframework==3.12.4
drf-yasg==1.20.0
idna==3.3
inflection==0.5.1
itypes==1.2.0
Jinja2==3.0.3
MarkupSafe==2.0.1
openapi-codec==1.3.2
packaging==21.3
pycodestyle==2.8.0
pyparsing==3.0.6
pytz==2021.3
requests==2.26.0
ruamel.yaml==0.17.17
ruamel.yaml.clib==0.2.6
```

```
simplejson==3.17.6
sqlparse==0.4.2
toml==0.10.2
uritemplate==4.1.1
urllib3==1.26.7
```

FRONTEND:

APP.JS:

```
import React from "react";
import "./App.css";
import NavBar from "./components/NavBar";
import HomeScreen from "./screens/HomeScreen";

function App() {
  return (
    <div className="trimp">
      <NavBar />
      <HomeScreen />
    </div>
  );
}

export default App;
```

STYLES:

```
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

body {
  font-family: "Roboto", sans-serif;
}
```

```css
.navbar-bg {
  background: #6e3cbc !important;
}

.navbar-brand {
  font-size: 2rem;
  background: #7267cb;
  padding: 10px 30px;
  border-radius: 50px;
}
.navbar-brand:hover {
  box-shadow: 1px 2px 4px 0px rgba(0, 0, 0, 0.459);
}

.box {
  max-width: 800px;
  border-radius: 50px;
  height: 55vh;
  background: #fff;
  box-shadow: 1px 0px 4px 0px rgba(0, 0, 0, 0.459);
}

.header {
  padding-bottom: 100px !important;
  background: #6e3cbc;
  border-radius: 0 0 50% 50%;
  box-shadow: 1px 2px 4px 0px rgba(0, 0, 0, 0.459);
}

/* .box-d {
    height: 50vh;
} */

.heading {
  margin-top: 10px;
  align-items: center;
  color: #7267cb;
  text-shadow: #655ab9;
```

```css
}

.url {
  max-width: 400px;
  border-radius: 50px;
  margin: 30px;
}

.button {
  background: #7267cb;
  padding: 10px 20px;
  border-radius: 50px;
  color: #fff;
  box-shadow: 0px 2px 4px 0px rgba(0, 0, 0, 0.459);
  border: none;
  text-decoration: none;
}
.button:hover {
  background: #655ab9;
  padding: 10px 20px;
  border-radius: 50px;
  color: rgba(255, 255, 255, 0.945);
}

input {
  border: 1px solid #655ab9 !important;
}

.link-box {
  padding: 10px;
  border-radius: 10px;
  border: 1px solid #655ab931;
}

@media screen and (max-width: 500px) {
  .link-copy {
    margin: 0 -15px;
    font-size: 0.9rem;
    word-wrap: break-word;
  }
  .link-box {
    width: 100% !important;
  }
}
```

HOMESCREEN.JS:

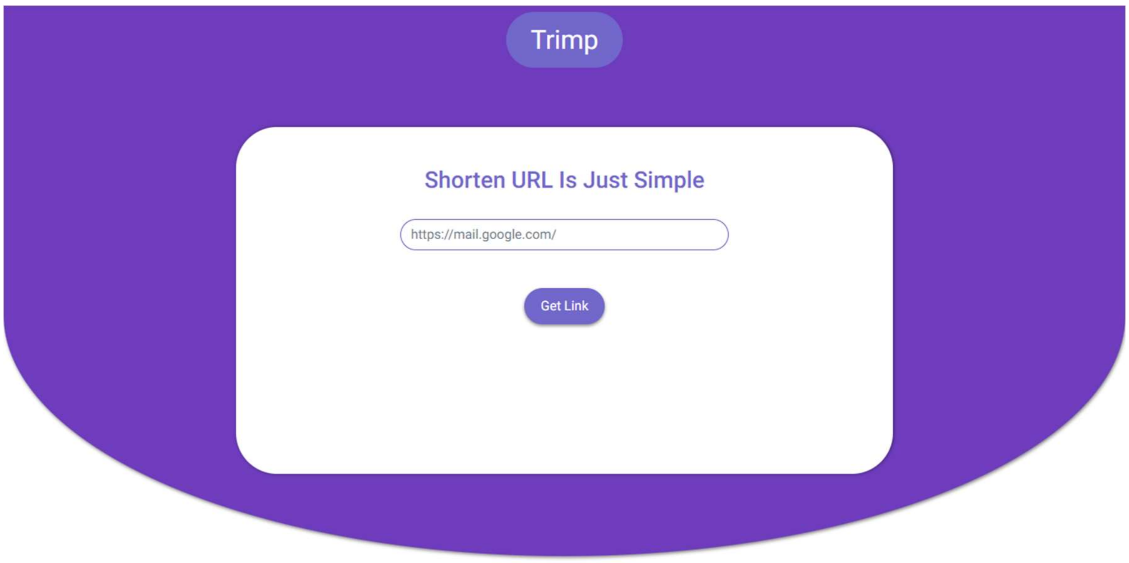
```
import { useState } from "react";
import axios from "axios";
const HomeScreen = () => {
  const [text, settext] = useState([]);
  const [short_str, setshortStr] = useState("");
  function CopyToClipboard(id) {
    const str = document.getElementById(id).innerText;
    const el = document.createElement("textarea");
    el.value = str;
    el.setAttribute("readonly", "");
    el.style.position = "absolute";
    el.style.left = "-9999px";
    document.body.appendChild(el);
    el.select();
    document.execCommand("copy");
    document.body.removeChild(el);
  }

  const createShortLink = () => {
    if (text.length > 8) {
      axios
        .post("/api/", {
          original_url: text,
        })
        .then(function (response) {
          setshortStr(response.data.short_char);
        })
        .catch(function (error) {
          console.log(error);
        });
    }
  };

  return (
    <div className="header p-5">
      <div className="text-center mx-auto box mx-3">
        <h3 className="heading pt-5 m-3">Shorten URL Is
Just Simple</h3>
        <input
          placeholder="https://mail.google.com/"
          type="text"
          className="form-control url  mx-auto"
          value={text}
          onChange={(e) => settext(e.target.value)}
        />
        <button className="button mt-3" onClick={() =>
createShortLink()}>
          Get Link
        </button>
        {short_str.length > 5 && (
          <div className="link-box w-75 row mx-auto my-3">
            <div className="col p-2">
              <span id="copy-text text-center">
                <a
                  className="link-copy text-center"

href={`https://trimp.pythonanywhere.com/${short_str}/`}

>{`https://trimp.pythonanywhere.com/${short_str}/`}</a>
              </span>
            </div>
            <div className="col p-2">
              <a
                href="#copied"
                className="button copy"
                onClick={() => CopyToClipboard("copy-
text")}
              >
                copy
              </a>
            </div>
          </div>
        )}
      </div>
    </div>
  );
};

export default HomeScreen;
```

NAVBAR:

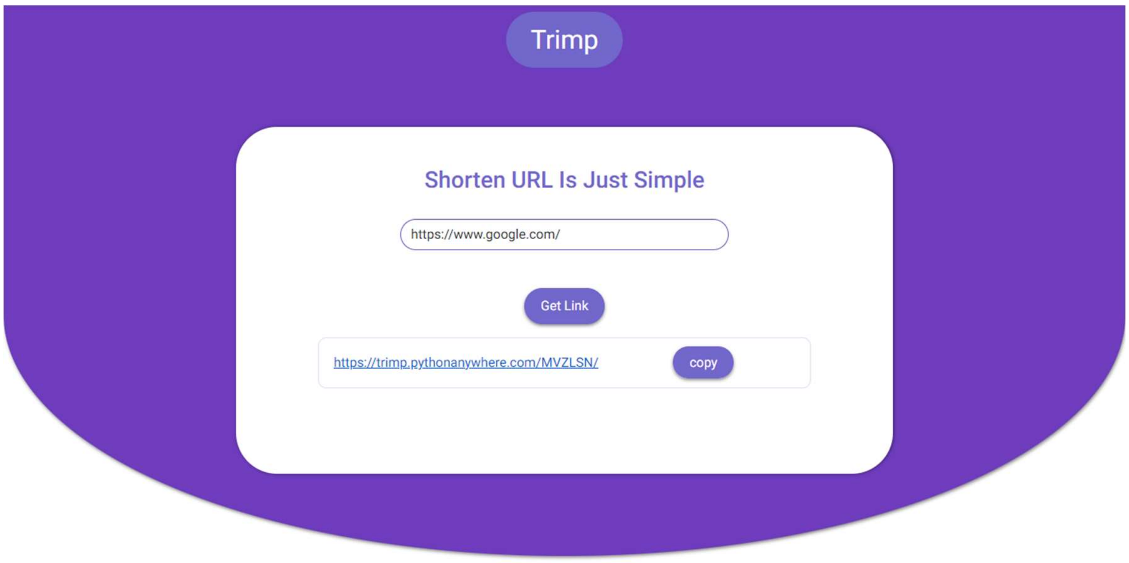
```
import React from "react";

const NavBar = () => {
  return (
    <div>
      <nav className="navbar navbar-expand-sm navbar-dark
bg-dark navbar-bg">
        <div className="container">
          <a className="navbar-brand mx-auto" href="/">
            Trimp
          </a>
        </div>
      </nav>
    </div>
  );
};

export default NavBar;
```

OUTPUT

HOME:

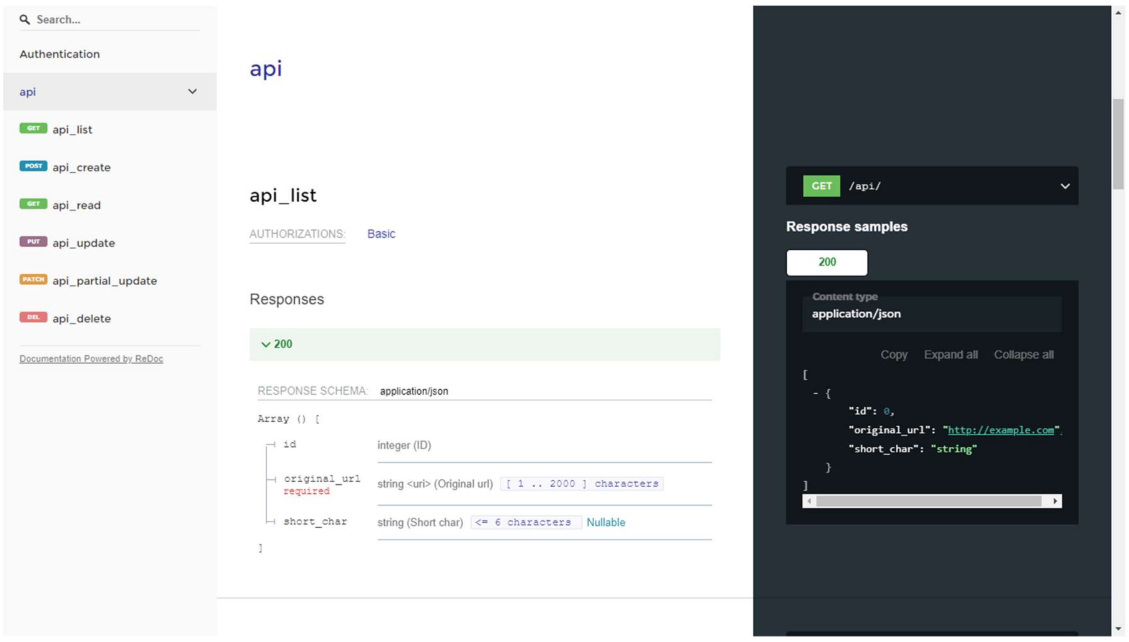


GENERATED URL:


Trimp
Shorten URL Is Just Simple
https://www.google.com/
Get Link
https://trimp.pythonanywhere.com/MVZLSN/
copy

https://trimp.pythonanywhere.com/MVZLSN/

API DOCUMNETATION:



api

api_list

AUTHORIZATIONS: Basic

Responses

200

RESPONSE SCHEMA: application/json

```
Array () [
  id             integer (ID)
  original_url   string <uri> (Original url)  [ 1 .. 2000 ] characters
  required
  short_char     string (Short char)  <= 6 characters  Nullable
]
```

GET /api/

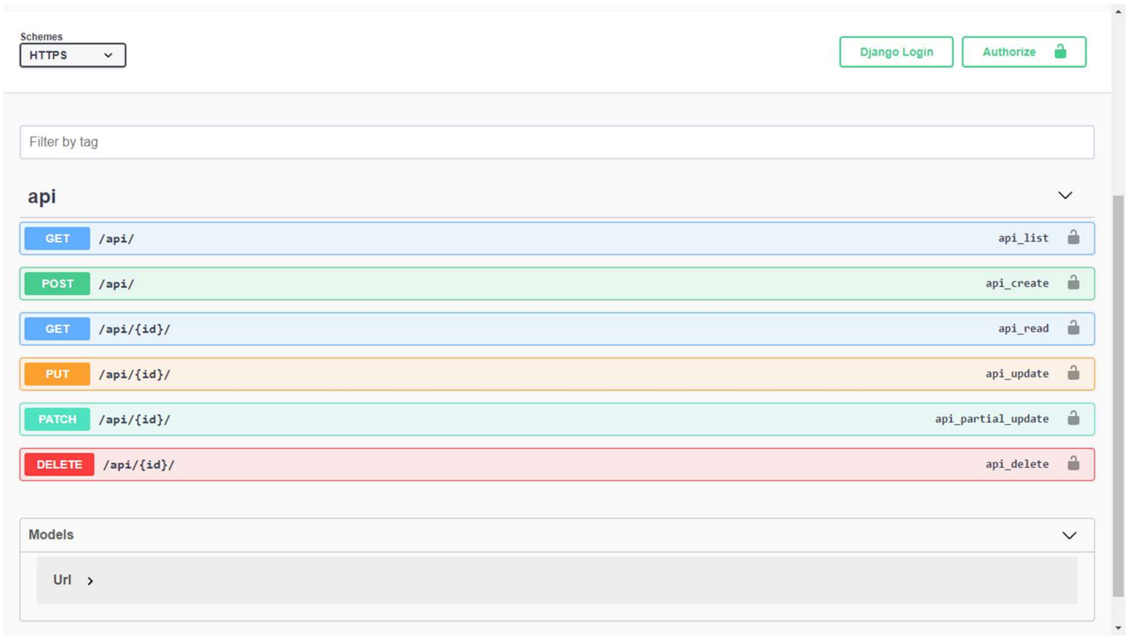
Response samples

200

Content type
application/json

```
[
  - {
        "id": 0,
        "original_url": "http://example.com",
        "short_char": "string"
    }
]
```

API ROUTES:



api

GET	/api/	api_list
POST	/api/	api_create
GET	/api/{id}/	api_read
PUT	/api/{id}/	api_update
PATCH	/api/{id}/	api_partial_update
DELETE	/api/{id}/	api_delete

Models

Url