

Django+React(step_by_step)

Django

```
d:\BEREGSZASZI\fantasy>py -m venv f_venv

d:\BEREGSZASZI\fantasy>cd f_venv

d:\BEREGSZASZI\fantasy\f_venv>Scripts\activate

(f_venv) d:\BEREGSZASZI\fantasy\f_venv>cd ..

(f_venv) d:\BEREGSZASZI\fantasy>pip install django

(f_venv) d:\BEREGSZASZI\fantasy>django-admin startproject config

(f_venv) d:\BEREGSZASZI\fantasy>rename config backend

(f_venv) d:\BEREGSZASZI\fantasy>cd backend

(f_venv) d:\BEREGSZASZI\fantasy\backend>py manage.py migrate

(f_venv) d:\BEREGSZASZI\fantasy\backend>code .

(f_venv) d:\BEREGSZASZI\fantasy\backend>py manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
May 19, 2024 - 19:02:11
Django version 5.0.6, using settings 'config.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

[19/May/2024 19:02:24] "GET / HTTP/1.1" 200 10629

(f_venv) d:\BEREGSZASZI\fantasy\backend>npx create-react-app frontend

(f_venv) d:\BEREGSZASZI\fantasy\backend>cd frontend

(f_venv) d:\BEREGSZASZI\fantasy\backend\frontend>pip install pillow

(f_venv) d:\BEREGSZASZI\fantasy\backend\frontend>pip install djangoestframework
```

config>setting.py

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
```

```
'django.contrib.staticfiles',  
'rest_framework',  
]
```

```
(f_venv) d:\BEREGSZASZI\fantasy\backend>py manage.py startapp fantasy
```

config>setting.py

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'rest_framework',  
    'fantasy',  
]
```

```
(f_venv) d:\BEREGSZASZI\fantasy\backend>py manage.py createsuperuser  
Username (leave blank to use 'user'): admin  
Email address:  
Password: admin  
Password (again): admin  
The password is too similar to the username.  
This password is too short. It must contain at least 8 characters.  
This password is too common.  
Bypass password validation and create user anyway? [y/N]: y  
Superuser created successfully.
```

```
(f_venv) d:\BEREGSZASZI\fantasy\backend>py manage.py runserver  
CTRL+C+C
```

fantasy>models.py

```
from django.db import models  
  
class ItemType(models.Model):  
    name = models.CharField(max_length=255)  
  
    def __str__(self):  
        return self.name
```

fantasy>admin.py

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

admin.site.register(ItemType)
```

```
(f_venv) d:\BEREGSZASZI\fantasy\backend>py manage.py makemigrations
```

```
(f_venv) d:\BEREGSZASZI\fantasy\backend>py manage.py migrate
```

<http://localhost:8000/admin/>

Felvesszük az itemtípusokat.

fantasy>models.py

```
from django.db import models

class ItemType(models.Model):
    name = models.CharField(max_length=255)

    def __str__(self):
        return self.name

class Item(models.Model):
    name = models.CharField(max_length=255)
    item_type = models.ForeignKey(ItemType, on_delete=models.CASCADE)
    description = models.TextField(null=True, blank=True)
    image = models.ImageField(upload_to="static/images/")
    date = models.DateField(auto_now_add=True)

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

fantasy>admin.py

```
from django.contrib import admin
from .models import ItemType
from .models import Item

admin.site.register(ItemType)
admin.site.register(Item)
```

```
(f_venv) d:\BEREGSZASZI\fantasy\backend>py manage.py makemigrations
```

```
(f_venv) d:\BEREGSZASZI\fantasy\backend>py manage.py migrate
```

<http://localhost:8000/admin/>

Felvesszük az itemeket.

Hogy a képek is megjeleníthetők legyenek admin felületről(a linkre kattintva):

config>settings.py

```
STATIC_URL = 'static/'

STATICFILES_DIRS = [
    BASE_DIR / 'static',
]
```

API-k készítése

fantasy>serializers.py (létre kell hozni!)

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

class ItemSerializers(ModelSerializer):
    class Meta:
        model = Item
        fields = '__all__'
```

fantasy>views.py

```
from rest_framework.decorators import api_view
from .models import Item
from .serializers import ItemSerializer
from rest_framework.response import Response

@api_view(['GET'])
def getAllItems(request):
    items = Item.objects.all()
    ser = ItemSerializer(items, many=True)

    return Response(ser.data)
```

config>urls.py

```
from django.contrib import admin
from django.urls import path
from fantasy import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('api/items/', views.getAllItems)
]
```

A serializers.py-ban beállítjuk a mélységet:

fantasy>serializers.py

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

class ItemSerializer(ModelSerializer):
    class Meta:
        model = Item
        fields = '__all__'
        depth = 1
```

A React-os alkalmazás futtatása a backend 8000-es portján

```
(f_venv) d:\BEREGSZASZI\fantasy\backend\frontend>npm run build
```

Létrejön a build könyvtár a frontend mappában.

config>settings.py

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [ BASE_DIR / 'frontend/build/' ],
```

...majd ugyanitt:

```
STATICFILES_DIRS = [
    BASE_DIR / 'static',
    BASE_DIR / 'frontend/build/static',
]
```

Készítünk egy view-ot, ami visszaadja ezt (frontend/build/index.html) az oldalt

config>urls.py

```
from django.contrib import admin
from django.urls import path
from fantasy import views

from django.shortcuts import render

def indexPage(request):
    return render(request, 'index.html')

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', indexPage, name="index"),
```

```
    path('api/items/', views.getAllItems),  
  ]
```

Innentől kezdve a 8000-es porton a frontend utoljára buildelt változata fog látszani.

React

```
PS D:\BEREGSZASZI\fantasy\backend\frontend> npm start
```

A 3000-es porton fut live-ben a frontendünk. Itt fejlesztünk, majd ha kész, akkor lebuildeljük és a 8000-esen lesz a végtermék.

Létrehozunk két fájlt az src>components mappában: itemSection, ItemCard.

frontend>src>App.js

```
import React from 'react'  
import ItemSection from './components/itemSection'  
  
const App = () => {  
  return (  
    <div>  
      <ItemSection />  
    </div>  
  )  
}  
  
export default App
```

frontend>src>index.js

```
import React from 'react';  
import ReactDOM from 'react-dom/client';  
import './index.css';  
import App from './App';  
  
const root = ReactDOM.createRoot(document.getElementById('root'));  
root.render(  
  <React.StrictMode>  
    <App />  
  </React.StrictMode>  
);
```

frontend>src>components>itemSection.js

```
import React, {useState, useEffect} from 'react'  
import ItemCard from './ItemCard'  
  
const ItemSection = () => {
```

```

const [items, setItems] = useState([])

useEffect(()=>{
  fetch('api/items/')
    .then(res => res.json())
    .then(data => setItems(data))
}, [])

return (
  <section className='item-section'>
    {items.map( (item, idx) => <ItemCard key={idx} {...item}/> )}
  </section>
)
}

export default ItemSection

```

frontend>src>components>ItemCards.js

```

import React from 'react'

const ItemCard = (props) => {
  return (
    <div>
      <img src={props.image}/>
      <h2>{props.name}</h2>
      <p>{props.description}</p>
      <strong>{props.item_type.name}</strong>
    </div>
  )
}

export default ItemCard

```

frontend>src>index.css

```

body{
  margin: 0;
  font-family: system-ui, -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu, Cantarell, 'Open Sans', 'Helvetica Neue', sans-serif;
  padding: 20px;
}

.item-section{
  padding: 10px;
  display: grid;
  grid-template-columns: repeat(1, 1fr);
  gap: 20px;
}

.item-section>div{
  border: 1px solid black;
}

```

```

padding: 10px;
border-radius: 5px;
display: flex;
flex-direction: column;
justify-content: start;
align-items: center;
gap: 20px;
}
.item-section p{
    text-align: justify;
}

.item-section img{
    width: 100%;
    max-width: 50vw;
    border: 1px solid black;
}

@media only screen and (min-width: 500px) {
    .item-section{
        padding: 10px;
        display: grid;
        grid-template-columns: repeat(2, 1fr);
        gap: 20px;
    }
}

@media only screen and (min-width: 800px) {
    .item-section{
        padding: 10px;
        display: grid;
        grid-template-columns: repeat(4, 1fr);
        gap: 20px;
    }
}

```