

# Criação de endpoints utilizando Aggregate + Ajustes na API



DEVinHouse

Parcerias para desenvolver a sua carreira

**SENAI**

<LAB365>

# AGENDA

- Ajustes na configuração da aplicação
- Criação de endpoints com Aggregate

# Ajustes na configuração da aplicação

Criar novo modelo de dados chamado comments.py dentro da models

```
def create_collection_comments(mongo_client):  
    comments_validator = {  
        "$jsonSchema": {  
            "bsonType": "object",  
            "required": ["_id", "name", "email", "movie_id", "text", "date"],  
            "properties": {  
                "_id": {  
                    "bsonType": "objectId",  
                    "description": "Chave definida da collection"  
                },  
                "name": {  
                    "bsonType": "string",  
                    "description": "Nome do usuário",  
                },  
            },  
        },  
    }
```

# Ajustes na configuração da aplicação

```
"email": {  
  "bsonType": "string",  
  "description": "Duração",  
},  
"movie_id": {  
  "bsonType": "objectId",  
  "description": "Vínculo da collection movie"  
},  
"text": {  
  "bsonType": "string",  
  "description": "Comentário",  
},
```

# Ajustes na configuração da aplicação

```
"date": {  
    "bsonType": "date",  
    "description": "Data do comentário"  
}  
},  
}  
}  
  
try:  
    mongo_client.create_collection("comments")  
except Exception as e:  
    print(e)  
  
mongo_client.command("collMod", "comments", validator=comments_validator)
```

# Ajustes na configuração da aplicação

Atualização da models de movies.py

```
def create_collection_movies(mongo_client):  
    movies_validator = {  
        "$jsonSchema": {  
            "bsonType": "object",  
            "required": [  
                "_id",  
                "plot",  
                "genres",  
                "runtime",
```

# Ajustes na configuração da aplicação

```
"cast",
```

```
"num_mflix_comments",
```

```
"title",
```

```
"fullplot",
```

```
"countries",
```

```
"released",
```

```
"directors",
```

```
"rated",
```

```
"awards",
```

```
"lastupdated",
```

```
"year",
```

```
"imdb",
```

```
"type",
```

```
"tomatoes"
```

```
],
```

# Ajustes na configuração da aplicação

```
}
```

```
}
```

```
try:
```

```
    mongo_client.create_collection("movies")
```

```
except Exception as e:
```

```
    print(e)
```

```
mongo_client.command("collMod", "movies", validator=movies_validator)
```



# Ajustes na configuração da aplicação

Atualização do app.py

```
from src.app import app, mongo_client
from src.app.models.comments import create_collection_comments
from src.app.models.movies import create_collection_movies
from src.app.routes import routes
from flask.cli import with_appcontext
import click

routes(app)

@click.command(name='create_collections')
@with_appcontext
def call_command():
    create_collection_movies(mongo_client)
    create_collection_comments(mongo_client)

app.cli.add_command(call_command)
```

# Ajustes na configuração da aplicação

Atualização do `__init__.py` da pasta `src.app`

REMOVER AS SEGUINTE IMPORTAÇÕES

```
from src.app.models.movies import create_collection_movies
```

```
from src.app.models.cast import create_collection_cast
```

REMOVER AS SEGUINTE LINHAS DE CÓDIGO

```
create_collection_movies(mongo_client)
```

```
create_collection_cast(mongo_client)
```

# Criação de endpoints com Aggregate

Atualizar endpoint para aggregate.

```
@movies.route("/get_all_movies", methods = ["GET"])
def get_all_movies():
    movies = mongo_client.movies.aggregate([
        {
            "$limit": 5
        }
    ])
```

# Criação de endpoints com Aggregate

Criar arquivo na pasta controller, chamado comments.py e escrever o seguinte código.

```
from flask import Blueprint
from flask.wrappers import Response
from src.app import mongo_client
from bson import json_util

comments = Blueprint("comments", __name__, url_prefix="/comments")

@comments.route("/get_all", methods = ["GET"])
def get_all():
    comments = mongo_client.comments.aggregate([
        {
            "$limit": 5
        }
    ])
```

# Criação de endpoints com Aggregate

```
return Response(  
    response=json_util.dumps({'records' : comments}),  
    status=200,  
    mimetype="application/json"  
)
```



# DEVinHouse

Parcerias para desenvolver a sua carreira

**OBRIGADO!**



<LAB365>