IMPLEMENTING WEB APPLICATION

INTEGRATING RAPID API

Date	30th October 2022
Team ID	PNT2022TMID36388
Project Name	News Tracker Application

Config.py

```
class Config:

    NEWS_BASE_URL_SOURCES = 'https://newsapi.org/v2/top-
headlines/sources?apiKey={}'
    NEWS_BASE_EVERYTHING_URL =
'https://newsapi.org/v2/everything?domains={}&apiKey={}'
    NEWS_BASE_HEADLINES_URL = 'https://newsapi.org/v2/top-
headlines?country=us&apiKey={}'
    NEWS_BASE_SOURCE = 'https://newsapi.org/v2/top-
headlines?sources={}&apiKey={}'
    API_KEY = "0b3e6ba5d69e431d93a99caca0163ea6"

class ProdConfig(Config):
    pass

class DevConfig(Config):
    DEBUG = True


config_options= {
    'development': DevConfig,
    'production': ProdConfig
}
```

Model.py

```
from .models import Articles
from .models import Sources
from newsapi import NewsApiClient
from .config  import  Config
import urllib.request,json

api_key=None
base_url=None
base_url_for_everything=None
base_url_top_headlines=None
base_source_list=None
```

```
def publishedArticles():
    newsapi = NewsApiClient(api_key= Config.API_KEY)
    get_articles = newsapi.get_everything(sources= 'cnn, reuters, cnbc, the-
verge, gizmodo, the-next-web, techradar, recode, ars-technica')
    all_articles = get_articles['articles']

    articles_results = []
    source = []
    title = []
    desc = []
    author = []
    img = []
    p_date = []
    url = []

    for i in range(len(all_articles)):
        article = all_articles[i]

        source.append(article['source'])
        title.append(article['title'])
        desc.append(article['description'])
        author.append(article['author'])
        img.append(article['urlToImage'])
        p_date.append(article['publishedAt'])
        url.append(article['url'])
        article_object = Articles(source, title, desc, author, img, p_date,
url)
        articles_results.append(article_object)
        contents = zip(source, title, desc, author, img, p_date, url)
    return  contents

def topHeadlines():
    newsapi = NewsApiClient(api_key= Config.API_KEY)
    top_headlines = newsapi.get_top_headlines(sources= 'cnn, reuters, cnbc,
techcrunch, the-verge, gizmodo, the-next-web, techradar, recode, ars-
technica')
    all_headlines = top_headlines['articles']

    articles_results = []
    source = []
    title = []
    desc = []
    author = []
    img = []
    p_date = []
    url = []
```

```
    for i in range(len(all_headlines)):
        headline = all_headlines[i]

        source.append(headline['source'])
        title.append(headline['title'])
        desc.append(headline['description'])
        author.append(headline['author'])
        img.append(headline['urlToImage'])
        p_date.append(headline['publishedAt'])
        url.append(headline['url'])
        article_object = Articles(source, title, desc, author, img, p_date,
url)
        articles_results.append(article_object)
        contents = zip(source, title, desc, author, img, p_date, url)
    return  contents

def randomArticles():
    newsapi = NewsApiClient(api_key= Config.API_KEY)
    random_articles = newsapi.get_everything(sources= 'the-verge, gizmodo,
the-next-web, recode, ars-technica')
    all_articles = random_articles['articles']

    articles_results = []
    source = []
    title = []
    desc = []
    author = []
    img = []
    p_date = []
    url = []

    for i in range(len(all_articles)):
        article = all_articles[i]

        source.append(article['source'])
        title.append(article['title'])
        desc.append(article['description'])
        author.append(article['author'])
        img.append(article['urlToImage'])
        p_date.append(article['publishedAt'])
        url.append(article['url'])
        article_object = Articles(source, title, desc, author, img, p_date,
url)
        articles_results.append(article_object)
        contents = zip(source, title, desc, author, img, p_date, url)
    return  contents
```

```
def businessArticles():
    newsapi = NewsApiClient(api_key= Config.API_KEY)
    business_articles = newsapi.get_top_headlines(category='business')
    all_articles = business_articles['articles']

    business_articles_results = []
    source = []
    title = []
    desc = []
    author = []
    img = []
    p_date = []
    url = []

    for i in range(len(all_articles)):
        article = all_articles[i]

        source.append(article['source'])
        title.append(article['title'])
        desc.append(article['description'])
        author.append(article['author'])
        img.append(article['urlToImage'])
        p_date.append(article['publishedAt'])
        url.append(article['url'])
        article_object = Articles(source, title, desc, author, img, p_date,
url)
        business_articles_results.append(article_object)
        contents = zip(source, title, desc, author, img, p_date, url)
    return  contents

def techArticles():
    newsapi = NewsApiClient(api_key= Config.API_KEY)
    tech_articles = newsapi.get_top_headlines(category='technology')
    all_articles = tech_articles['articles']

    tech_articles_results = []
    source = []
    title = []
    desc = []
    author = []
    img = []
    p_date = []
    url = []

    for i in range(len(all_articles)):
        article = all_articles[i]
```

```
        source.append(article['source'])
        title.append(article['title'])
        desc.append(article['description'])
        author.append(article['author'])
        img.append(article['urlToImage'])
        p_date.append(article['publishedAt'])
        url.append(article['url'])

        article_object = Articles(source, title, desc, author, img, p_date,
url)

        tech_articles_results.append(article_object)
        contents = zip(source, title, desc, author, img, p_date, url)
    return  contents

def entArticles():
    newsapi = NewsApiClient(api_key= Config.API_KEY)
    ent_articles = newsapi.get_top_headlines(category='entertainment')
    all_articles = ent_articles['articles']

    ent_articles_results = []
    source = []
    title = []
    desc = []
    author = []
    img = []
    p_date = []
    url = []

    for i in range(len(all_articles)):
        article = all_articles[i]

        source.append(article['source'])
        title.append(article['title'])
        desc.append(article['description'])
        author.append(article['author'])
        img.append(article['urlToImage'])
        p_date.append(article['publishedAt'])
        url.append(article['url'])

        article_object = Articles(source, title, desc, author, img, p_date,
url)
        ent_articles_results.append(article_object)
        contents = zip(source, title, desc, author, img, p_date, url)
    return  contents

def scienceArticles():
    newsapi = NewsApiClient(api_key= Config.API_KEY)
```

```
    science_articles = newsapi.get_top_headlines(category='science')
    all_articles = science_articles['articles']

    science_articles_results = []
    source = []
    title = []
    desc = []
    author = []
    img = []
    p_date = []
    url = []

    for i in range(len(all_articles)):
        article = all_articles[i]

        source.append(article['source'])
        title.append(article['title'])
        desc.append(article['description'])
        author.append(article['author'])
        img.append(article['urlToImage'])
        p_date.append(article['publishedAt'])
        url.append(article['url'])

        article_object = Articles(source, title, desc, author, img, p_date,
url)
        science_articles_results.append(article_object)
        contents = zip(source, title, desc, author, img, p_date, url)
    return  contents

def sportArticles():
    newsapi = NewsApiClient(api_key= Config.API_KEY)
    sport_articles = newsapi.get_top_headlines(category='sports')
    all_articles = sport_articles['articles']

    sport_articles_results = []
    source = []
    title = []
    desc = []
    author = []
    img = []
    p_date = []
    url = []

    for i in range(len(all_articles)):
        article = all_articles[i]

        source.append(article['source'])
```

```
        title.append(article['title'])
        desc.append(article['description'])
        author.append(article['author'])
        img.append(article['urlToImage'])
        p_date.append(article['publishedAt'])
        url.append(article['url'])

        article_object = Articles(source, title, desc, author, img, p_date,
url)
        sport_articles_results.append(article_object)
        contents = zip(source, title, desc, author, img, p_date, url)
    return  contents

def healthArticles():
    newsapi = NewsApiClient(api_key= Config.API_KEY)
    health_articles = newsapi.get_top_headlines(category='health')
    all_articles = health_articles['articles']

    health_articles_results = []
    source = []
    title = []
    desc = []
    author = []
    img = []
    p_date = []
    url = []

    for i in range(len(all_articles)):
        article = all_articles[i]

        source.append(article['source'])
        title.append(article['title'])
        desc.append(article['description'])
        author.append(article['author'])
        img.append(article['urlToImage'])
        p_date.append(article['publishedAt'])
        url.append(article['url'])

        article_object = Articles(source, title, desc, author, img, p_date,
url)
        health_articles_results.append(article_object)
        contents = zip(source, title, desc, author, img, p_date, url)
    return  contents

def get_news_source():
  get_news_source_url = 'https://newsapi.org/v2/sources?apiKey=' +
Config.API_KEY
  with urllib.request.urlopen(get_news_source_url) as url:
```

```
    get_news_source_data = url.read()
    get_news_source_response = json.loads(get_news_source_data)

    news_source_results = None

    if  get_news_source_response['sources']:
      news_source_results_list = get_news_source_response['sources']
      news_source_results = process_sources(news_source_results_list)

  return news_source_results

def process_sources(source_list):

  news_source_result = []
  for news_source_item in source_list:
    name = news_source_item.get('name')
    description = news_source_item.get('description')
    url = news_source_item.get('url')

    if name:
      news_source_object = Sources(name, description,url)
      news_source_result.append(news_source_object)
  return news_source_result
```