poke_api_all

June 3, 2024

1. Mengimpor Library

```
[ ]: import requests
     import asyncpg
     import asyncio
```

2. Konfigurasi Database

```
[ ]: DATABASE_URL = "postgresql://poke:p0k3!!123@172.23.1.22/pokebase"
```

3. Pool Koneksi Database Asinkron

```
[ ]: async def get_pool():
         return await asyncpg.create_pool(DATABASE_URL)
```

4. Mengambil dan Menyimpan Data Kemampuan Pokemon

```
[ ]: async def fetch_and_store_ability(pool, pokemon_ability_id):
         try:
             url = f"https://pokeapi.co/api/v2/ability/{pokemon_ability_id}"
             call = requests.get(url)
             call.raise_for_status()  # Raise an HTTPError for bad responses
             response = call.json()

             async with pool.acquire() as conn:
                 for effect_entry in response['effect_entries']:
                     await conn.execute("""
                         INSERT INTO pokemon_effect (pokemon_ability_id, effect,␣
      ↪language, short_effect)
                         VALUES ($1, $2, $3, $4);
                     """, int(pokemon_ability_id), str(effect_entry['effect']),
                     str(effect_entry['language']['name']),␣
      ↪str(effect_entry['short_effect']))

             print(f"Saved ability_id {pokemon_ability_id} to database")
             return {
                 "pokemon_ability_id": pokemon_ability_id,
                 "status": "success"
             }
         except requests.exceptions.RequestException as e:
```

```
        print(f"HTTP error fetching ability_id {pokemon_ability_id}: {str(e)}")
        return {
            "pokemon_ability_id": pokemon_ability_id,
            "error": str(e)
        }
    except Exception as e:
        print(f"Error saving ability_id {pokemon_ability_id}: {str(e)}")
        return {
            "pokemon_ability_id": pokemon_ability_id,
            "error": str(e)
        }
```

5. Fungsi Utama

```
[ ]: async def main():
    pool = await get_pool()
    tasks = [fetch_and_store_ability(pool, pokemon_ability_id) for␣
 ↪pokemon_ability_id in range(1, 1000)]
    results = await asyncio.gather(*tasks)
    print("Completed fetching and storing abilities")
    print(results)
```

6. Menjalankan Skrip

```
[ ]: if __name__ == "__main__":
    asyncio.run(main())
```