

## GetAllPokemonsQueryHandler.cs

```
using ErrorOr;
using MediatR;
using PokeApiNet;
using Contracts.Pokemons;

namespace Application.Pokemons.Queries.GetAllPokemons;

public class GetAllPokemonsQueryHandler : IRequestHandler<GetAllPokemonsQuery,
ErrorOr<List<PokemonBasicResponse>>>
{
    private readonly PokeApiClient _pokeApiService;

    public GetAllPokemonsQueryHandler(PokeApiClient pokeApiService)
    {
        _pokeApiService = pokeApiService;
    }

    public async Task<ErrorOr<List<PokemonBasicResponse>>>
Handle(GetAllPokemonsQuery request, CancellationToken cancellationToken)
    {
        // limit 4. Generation 493
        // limit 5. Generation 649
        // request.Limit
        NamedApiResourceList<Pokemon> pokemons = await
_pokeApiService.GetNamedResourcePageAsync<Pokemon>(
            limit: request.Limit,
            offset: request.Page * request.Limit,
            cancellationToken
        );

        var pokemonTasks = pokemons.Results.Select(async pokemonResource =>
        {
            var pokemon = await _pokeApiService.GetResourceAsync<Pokemon>
(pokemonResource.Name);
            var response = new PokemonBasicResponse(
                Id: pokemon.Id,
                Name: pokemon.Name,
                Types: pokemon.Types.Select(t => t.Type.Name).ToList(),
                SpriteUrl: pokemon.Sprites.Other.OfficialArtwork.FrontDefault
            );
            return response;
        });

        var pokemonResponses = await Task.WhenAll(pokemonTasks);
        var resultList = pokemonResponses.ToList();

        return resultList;
    }
}
```

---

Path: [./Backend/Application/Pokemons/Queries/GetAllPokemons/GetAllPokemonsQueryHandler.cs](#)