GetPokemonDetailsQueryHandler.cs

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


namespace Application.Pokemons.Queries.GetPokemonDetails;

public class GetPokemonDetailsQueryHandler :
IRequestHandler<GetPokemonDetailsQuery, ErrorOr<PokemonDetailedResponse>>
{
    private readonly PokeApiClient _pokeApiService;

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

    public async Task<ErrorOr<PokemonDetailedResponse>>
Handle(GetPokemonDetailsQuery request, CancellationToken cancellationToken)
    {
        // Get the main Pokemon data
        var pokemon = await _pokeApiService.GetResourceAsync<Pokemon>
(request.Name, cancellationToken);

        // Extract all unique move names
        var moveNames = pokemon.Moves.Select(m => m.Move.Name).ToList();

        // Create tasks for all move requests
        var moveTasks = moveNames.Select(name =>
            _pokeApiService.GetResourceAsync<Move>(name, cancellationToken));

        // Wait for all moves to complete
        var moves = await Task.WhenAll(moveTasks);

        // Create a dictionary for quick lookup
        var moveDict = moves.ToDictionary(m => m.Name, m => m);

        // Create move details
        var moveDetails = pokemon.Moves
            .Select(moveEntry => {
                var move = moveDict[moveEntry.Move.Name];
                return new MoveDetail(
                    Name: move.Name,
                    Type: move.Type.Name,
                    Power: move.Power,
                    Accuracy: move.Accuracy,
                    PP: move.Pp,
                    DamageClass: move.DamageClass.Name
                );
            })
            .ToList();

        var response = new PokemonDetailedResponse(
            Id: pokemon.Id,
            Name: pokemon.Name,
            Types: pokemon.Types.Select(t => t.Type.Name).ToList(),
            Moves: moveDetails,
            PokemonStats: pokemon.Stats.Select(t => new StatDetail(t.Stat.Name,
t.BaseStat)).ToList(),
            SpriteUrl: pokemon.Sprites.FrontDefault);

        return response;
    }
}
```

Path: `./Backend/Application/Pokemons/Queries/GetPokemonDetails/GetPokemonDetailsQueryHandler.cs`