

Toelichting werking Pokemon basis applicatie

Pokemondata

Binnen de map Resources vind je een tekstdocument PokemonData.txt. Dit document bevat alle pokemondata in een JSON-formaat:

```
{Property1:
[
  {
    "id": "001",
    "name": "Bulbasaur",
    "img": "http://img.pokemondb.net/artwork/bulbasaur.jpg",
    "type": [
      "Grass",
      "Poison"
    ],
    "stats": {
      "hp": "45",
      "attack": "49",
      "defense": "49",
      "spattack": "65",

      Etc...

    ...
  }
]
}
```

Dit bestand wordt steeds mee naar de Debug map gekopieerd (in een map Resources) zodat het programma hier altijd beschikking tot heeft.

Tekstbestand parsen

De JSON-data in het bestand wordt ingelezen en geparsed mbv van de Nuget bibliotheek: Newtonsoft.Json. Deze bibliotheek laat toe om JSON te parsen naar respectievelijke objecten van een specifiek type.

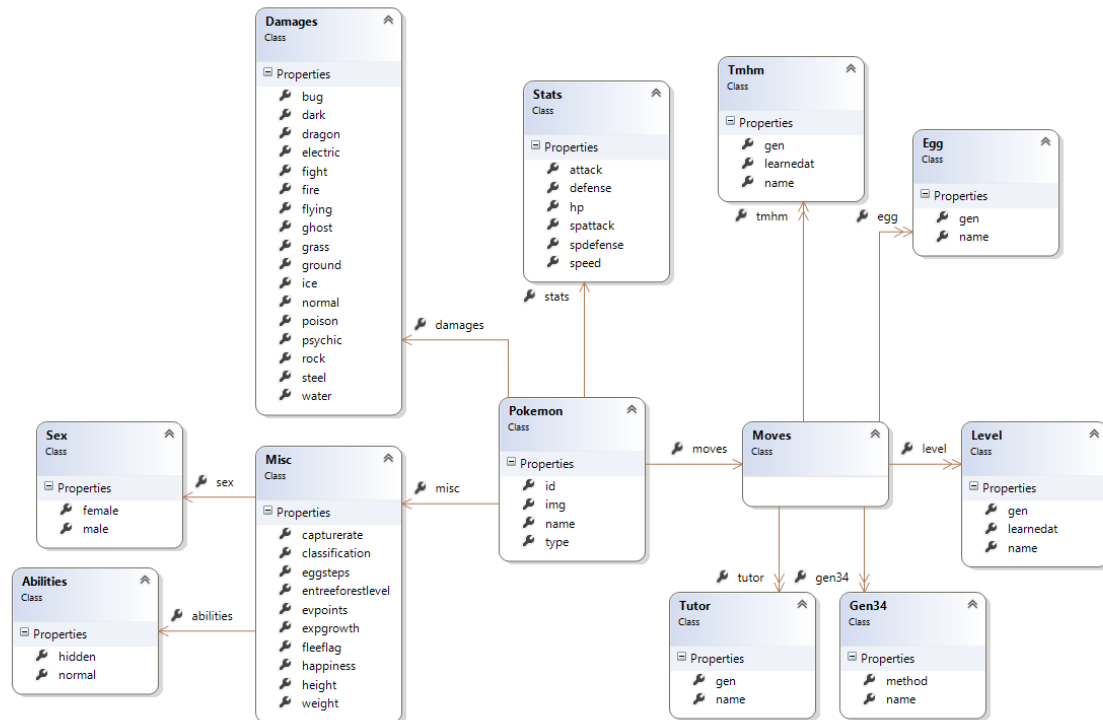
Voorbeelden van hoe je kan werken JSON.NET staan hier:

- <https://codevan1001nacht.wordpress.com/2014/02/09/json-parsen-met-json-net/>
- <http://www.youtube.com/watch?v=PLMbV4eBPEg>

De effectieve parsing met JSON.NET gebeurt in de DataAccessLayer(DAL)

Pokemonklasse

Pokemon.cs bevat de effectieve Pokemon klassen waarnaar JSON.NET zal parsen.



Opgelet

PAS DE NAMEN VAN DE PROPERTIES NIET AAN, anders zal de parsing niet meer werken. Wil je alsnog de namen veranderen dan plaats je boven de property het volgende attribuut waarbij de **OriginalPropertynaam** gelijk moet zijn aan de originele naam die de property had (en deze is dus ook gelijk aan de JSON naam in het tekstbestand):

```
[JsonProperty(PropertyName = "OriginalPropertynaam")]
```

PokemonDAL

In de map PokemonDAL van je solution staat één klasse: DAL.cs . Deze klasse zal je data parsen en aan de rest van het programma aanbieden als ene lijst van pokemons. Voorts laat de DAL toe om ook een Pokemon te verkrijgen met een specifiek Id.

Deze twee methoden zijn:

```
public static List<Pokemon> GetAllPokemon()  
public static Pokemon GetPokemon(int nr)
```

PokemonDAL gebruiken

Beide methoden zijn static gemaakt waarbij het singletonpattern werd toegepast op de private allpokemons lijst. Dit tweede voorkomt dat het hele tekstbestand maar eenmalig wordt uitgelezen , waarna de geparseerde lijst in het geheugen blijft staan tot het programma afsluit.

De 2 methoden werden statisch gemaakt zodat je geen DAL object elders moet aanmaken om de data te verkrijgen.

Je kan dus van elders de data als volgt verkrijgen:

```
List<Pokemon> result = PokemonDAL.DAL.GetAllPokemon();
```

Of indien je enkel de Pokemon met id 25 wenst:

```
Pokemon single = PokemonDAL.DAL.GetPokemon(25);
```