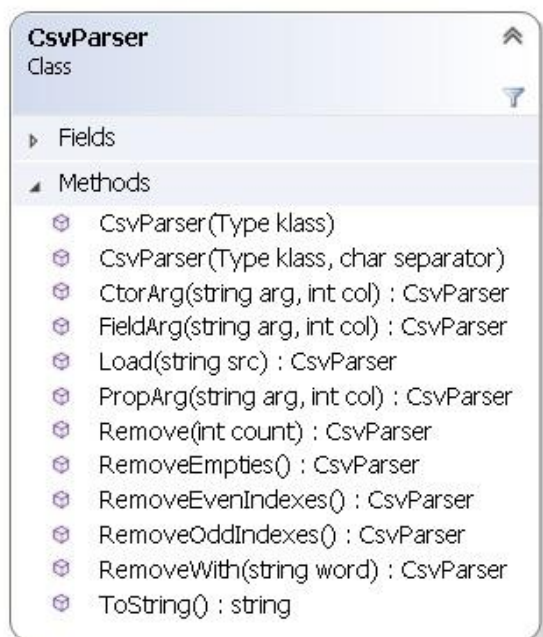


A biblioteca Csvier oferece processamento de dados em formato CSV. Esta biblioteca disponibiliza uma classe CsvParser com a API pública apresentada no diagrama seguinte:



O exemplo seguinte demonstra um caso de utilização desta classe para processamento de dados meteorológicos fornecidas pelo serviço [world weather online](http://api.worldweatheronline.com/).

```
string weatherJanuaryInLisbon = ...;
CsvParser pastWeather = new CsvParser(typeof(WeatherInfo))
    .CtorArg("date", 0)
    .CtorArg("tempC", 2)
    .PropArg("precipMM", 11)
    .PropArg("desc", 10);
object[] items = pastWeather
    .Load(weatherJanuaryInLisbon)
    .RemoveWith("#")
    .Remove(1)
    .RemoveEvenIndexes()
    .Parse();
```

Pode obter os dados referentes ao exemplo anterior a partir do seguinte URL substituindo \*\*\* por uma chave que deve obter ao registar-se no serviço [world weather online](http://api.worldweatheronline.com/).

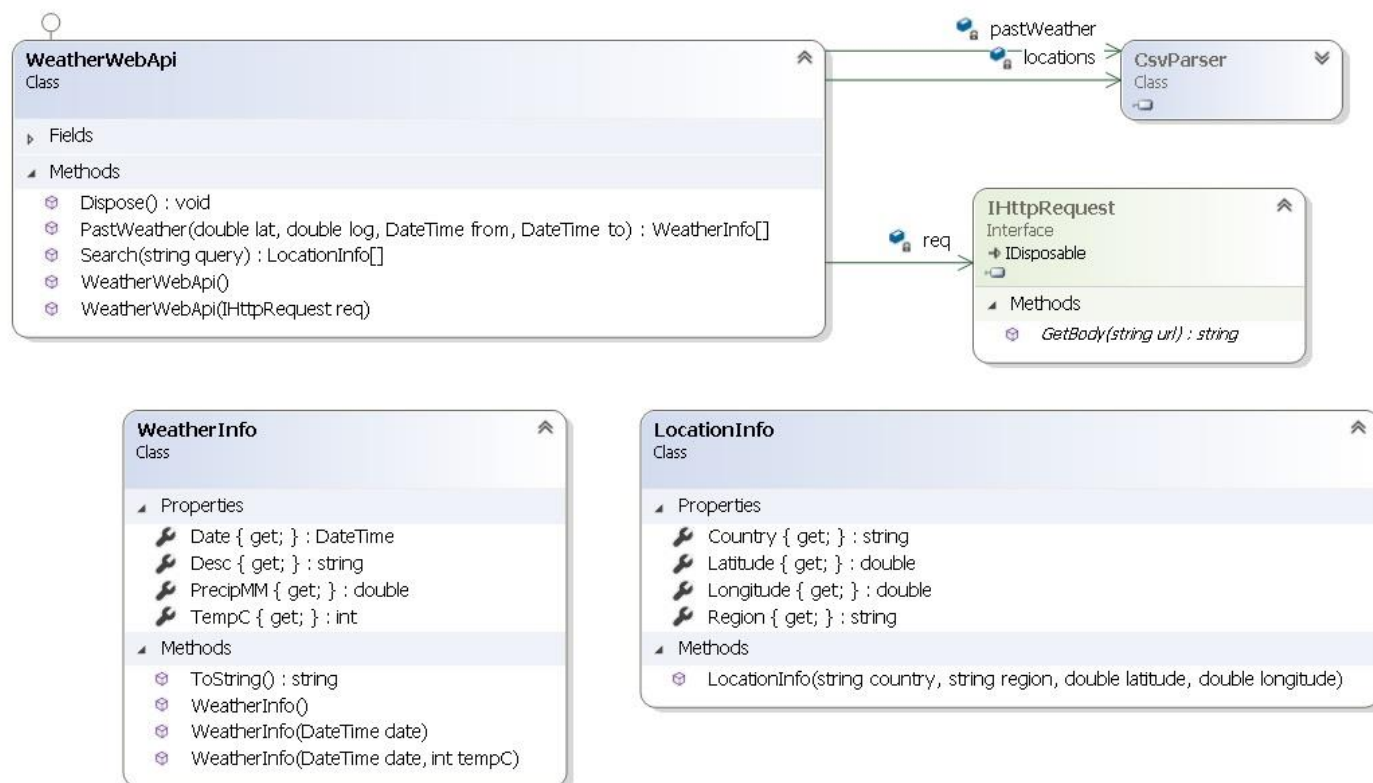
[http://api.worldweatheronline.com/premium/v1/past-weather.ashx?q=37.017,-7.933&date=2019-01-01&enddate=2019-01-30&tp=24&format=csv&key=\\*\\*\\*\\*\\*](http://api.worldweatheronline.com/premium/v1/past-weather.ashx?q=37.017,-7.933&date=2019-01-01&enddate=2019-01-30&tp=24&format=csv&key=*****)

Os métodos de CsvParser têm o seguinte comportamento:

- CsvParser(Type klass, char separator) – klass representa a classe alvo a instanciar e separator o caracter separador das colunas nos dados CSV. Este último é vírgula (,) por omissão.
- CtorArg(string name, int col) – associa o parâmetro name do construtor com a coluna de índice col dos dados CSV.
- PropArg(string name, int col) – associa a propriedade name com a coluna de índice col dos dados CSV.
- FieldArg(string name, int col) – associa o campo name com a coluna de índice col dos dados CSV.
- Load(string data) – lê e carrega os dados de data dividindo por linhas.
- RemoveWith(string word) – Remove todas as linhas que comecem com a string word.
- Remove(int count) – Remove um número de linhas iniciais igual count.

- `RemoveEvenIndexes()` – Remove as linhas de índice par (0, 2, 4,...)
- `RemoveOddIndexes()` – Remove as linhas de índice ímpar (1, 3, 5,...)
- `RemoveEmpties()` – Remove as linhas vazias, iguais a "".
- `Parse():object[]` – Retorna um array com instâncias do tipo `klass` especificado no construtor, correspondente aos dados carregados no método `Parse()` e mantidos no estado actual. As instâncias devem ser criadas usando o construtor com os parâmetros especificados no método `CtorArg` (na respectiva ordem) e inicializando os campos e propriedades que foram especificados em `PropArg()` e `FieldArg()`.  
Assuma que os tipos de parâmetros do construtor, campos ou propriedades, têm sempre um método estático `Parse(string)`, que converte uma string num valor do respectivo tipo.

A biblioteca `Clima` disponibiliza a classe `WeatherWebApi` que permite consultar as informações climáticas de World Weather Online e pesquisar localizações também a partir desta Web API. Para tal a classe `WeatherWebApi` usa o `CsvParser` para processamento dos dados CSV e uma classe auxiliar `HttpRequest` para realizar os pedidos HTTP à Web API.



A correspondência entre os parâmetros do construtor, propriedades ou campos e as colunas dos dados CSV é especificada nas classes por intermédio de *custom attributes*.

A API de `CsvParser` é genérica no tipo `CsvParser<T>`. O parâmetro `klass` do seu construtor pode ser determinado a partir de `T`. Por sua vez o método `Parse()` pode retornar `T[]`.

`CsvParser` oferece ainda um método `Parse(Func<string, T> parser)` em que a função `parser` é a responsável por fazer o processamento da informação de cada linha e instanciação do tipo `T`.

Na utilização do `Parse(Func<string, T> parser)` NÃO é usada reflexão na instanciação de `T`.

O resultado do processamento das linhas é mantido na forma de uma instância de `IEnumerable`. Repare que o tipo `String` também implementa `IEnumerable<Char>` pelo que não existe a utilização do método `Split` e NÃO são usados *arrays* auxiliares.

O método `ToEnumerable(Func<string, T> parser)` funciona em alternativa ao `Parse` e retorna uma instância de `IEnumerable<T>` *lazy*.