

Data de Entrega: 13 de Abril de 2015

Objectivo: implementar no package `jsonzai` a funcionalidade de conversão de *strings* Json em objectos Java. Este package deverá ser reutilizado em `weathergw` para processamento da resposta HTTP em formato Json da API do serviço *Worl Weather Online*. O trabalho deve ser **entregue com o package `jsonzai`, os respectivos testes unitários e utilização da funcionalidade de `jsonzai` no `weathergw`.**

Implemente no package `pt.isel.mpd.jsonzai` uma classe `JsonParser` com os métodos de instância:

- `<T> T toObject(String src, Class<T> dest).`
- `<T> List<T> toList(String src, Class<T> dest).`

Por exemplo, dado o pedido HTTP ao URI: <https://api.github.com/users/achiu>, que retorna uma resposta com o conteúdo da Figura 1, então a seguinte utilização do método `toObject`:

```
JsonParser parser = new JsonParser();
GithubUser user = parser.<GithubUser>toObject(response, GithubUser.class)
```

retorna uma instância de `GithubUser` com a definição da Figura 2

<pre>{ login: "achiu", id: 24772, type: "User", name: "Arthur Chiu", company: "GitHub", location: "San Francisco, CA", email: "achiu@github.com", hireable: false, public_repos: 52, public_gists: 37, followers: 198, following: 38, created_at: "2008-09-16T03:24:44Z", updated_at: "2015-03-26T06:29:14Z" }</pre>	<pre>public class GithubUser { String login; int id; String email; String location; }</pre>
--	---

Figura 1

Figura 2

O processo de conversão de Json em objectos, procura para cada par nome-valor da *string* Json fonte um campo com o mesmo nome na classe destino, afectando esse campo com o respectivo valor.

São suportados apenas campos de tipos primitivos, `String`, `java.util.Date` e `java.util.List`:

- Se um campo é do tipo `List` então pressupõe que o respectivo valor na *string* Json representa um *array*.
- Se um campo é do tipo de outra classe então pressupõe que o respectivo valor na *string* Json representa um objecto.

Por exemplo, dado o pedido HTTP ao URI: <https://api.github.com/users/achiu/repos>, que retorna uma resposta com o conteúdo da Figura 3, então a seguinte utilização do método `toList`:

```
List<GithubRepo> repos = parser.<GithubRepo>toList(response, GithubRepo.class)
```

retorna uma instância compatível com `List` contendo elementos do tipo `GithubRepo` com a definição da Figura 4.

<pre>[{ id: 363183, name: "achiu.github.com", full_name: "achiu/achiu.github.com", owner: {...}, private: false, ... }, {...}, ...]</pre>	<pre>class GithubRepo { String id; String name; String fullName; GithubUser owner; }</pre>
---	--

Figura 3

Figura 4