

json Manual

Naoki Kaneko

このライブラリは JSON 文字列をパースしたり、逆に JSON データ構造を JSON 文字列に変換したりするものです。

`json.satyg` というファイルでは JSON の規格に準拠したパーサと文字列化関数が定義されています。そして、`json5.satyg` というファイルでは JSON5 の規格に準拠したパーサと文字列化関数が定義されています。

1. `json.satyg` で定義されているもの

グローバル空間に

```
1 type json =  
2   | JsonNull  
3   | JsonBool of bool  
4   | JsonInt of int  
5   | JsonFloat of float  
6   | JsonString of string  
7   | JsonArray of json list  
8   | JsonObject of (string * json) list
```

が定義されています。上から順に

- (1) `JsonNull` : `null` を表す
- (2) `JsonBool` : `true` もしくは `false` を格納する
- (3) `JsonInt` : 自然数を格納する
- (4) `JsonFloat` : 小数を格納する
- (5) `JsonString` : 文字列を格納する
- (6) `JsonArray` : リストを格納する

(7) JsonObject : object を格納する

という役割を持っています。

json.satyg では JSON モジュールが定義されています。

JSON モジュールでは文字列をパースして json データを生成する `parse : json json-parse-error result` という関数（ここでの `result` は `satysfi-base` の `base.satyg` で定義されているもの）と、json データを文字列に変換する `to-string : json -> string` という関数が公開されています。

```
1  let json-str = `  
2  {  
3    "foo" : "foo",  
4    "bar" : {  
5      "bar1" : [1,2,3],  
6      "bar2" : [  
7        {"baz1" : 123, "baz2" : -1.2},  
8        {"baz3" : null, "baz4" : true}  
9      ]  
10   }  
11 }  
12 `  
13 let json = json-str |> JSON.parse |> Result.unwrap  
14 let str = JSON.to-string json
```

というコードがあったとき、`str` は `{"foo": "foo", "bar": {"bar1": [1,2,3], "bar2": [{"baz1": 123, "baz2": -1.2}, {"baz3": null, "baz4": true}]}}` になります。

また、json データから目的のデータを取り出すために

- `is-null : json -> bool`
- `is-bool : json -> bool`
- `as-bool : json -> bool option`
- `is-int : json -> bool`
- `as-int : json -> int option`

- `is-float : json -> bool`
- `as-float : json -> float option`
- `is-string : json -> bool`
- `as-string : json -> string option`
- `is-array : json -> bool`
- `as-array : json -> (json list) option`
- `is-object : json -> bool`
- `as-object : json -> ((string * json) list) option`

という各種関数が定義されています。

2. `json5.satyg` で定義されているもの

グローバル空間に

```
1  type json5 =  
2    | Json5Null  
3    | Json5Bool of bool  
4    | Json5Int of int  
5    | Json5Float of float  
6    | Json5NaN  
7    | Json5Infinity  
8    | Json5NegInfinity  
9    | Json5String of string  
10   | Json5Array of json5 list  
11   | Json5Object of (string * json5) list
```

が定義されています。上から順に

- (1) `Json5Null` : `null` を表す
- (2) `Json5Bool` : `true` もしくは `false` を格納する
- (3) `Json5Int` : 自然数を格納する
- (4) `Json5Float` : 小数を格納する
- (5) `Json5NaN` : `NaN` を表す

- (6) `Json5Infinity` : 正の無限大を表す
- (7) `Json5NegInfinity` : 負の無限大を表す
- (8) `Json5String` : 文字列を格納する
- (9) `Json5Array` : リストを格納する
- (10) `Json5Object` : `object` を格納する

という役割を持っています。

`json5.satyg` では JSON5 モジュールが定義されています。

JSON5 モジュールでは、文字列から `json5` データを生成する `parse : json5 json5-parse-error result` という関数（ここでの `result` は `satysfi-base` の `base.satyg` で定義されているもの）と、`json5` データを文字列に変換する `to-string : json -> string` という関数が公開されています。

```

1  let json5-str = `
2  {
3    // comments
4    "foo" : "foo",
5    /*
6      block comments
7      / hoge * fuga/
8    */
9    bar : {
10     "bar1" : [1,2,3,],
11     "bar2" : [
12       {"baz1" : 123, "baz2" : -Infinity},
13       {"baz3" : null, "baz4" : true, "baz" : 0xabcd}
14     ]
15   },
16 }
17 `
18 let json5 = json5-str |> JSON5.parse |> Result.unwrap
19 let str = JSON5.to-string json5

```

というコードがあったとき、`str` は `{"foo": "foo", "bar": {"bar1": [1,2,3],`

`"bar2": [{ "baz1": 123, "baz2": -Infinity }, { "baz3": null, "baz4": true, "baz": 43981 }]] }` になります。

また、`json5` データから目的のデータを取り出すために

- `is-null : json5 -> bool`
- `is-bool : json5 -> bool`
- `as-bool : json5 -> bool option`
- `is-int : json5 -> bool`
- `as-int : json5 -> int option`
- `is-float : json5 -> bool`
- `as-float : json5 -> float option`
- `is-nan : json5 -> bool`
- `is-infinity : json5 -> bool`
- `is-neg-infinity : json5 -> bool`
- `is-string : json5 -> bool`
- `as-string : json5 -> string option`
- `is-array : json5 -> bool`
- `as-array : json5 -> (json5 list) option`
- `is-object : json5 -> bool`
- `as-object : json5 -> ((string * json5) list) option`

という各種関数が定義されています。

3. 必要なバージョンや依存など

`satysfi-dist` と `satysfi-base` の二つに依存します。また、必要なバージョンは `SATYSFI` の `v0.0.5` 以上が必要になります。