

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

```
type json =  
  | JsonNull  
  | JsonBool of bool  
  | JsonInt of int  
  | JsonFloat of float  
  | JsonString of string  
  | JsonArray of json list  
  | 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 `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 `to-string : json -> string` という関数の 2 つが公開されています。

```
let json-str = `
{
  "foo" : "foo",
  "bar" : {
    "bar1" : [1,2,3],
    "bar2" : [
      {"baz1" : 123, "baz2" : -1.2},
      {"baz3" : null, "baz4" : true}
    ]
  }
}
`

let json = json-str |> JSON.parse |> Result.unwrap
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。

2. `json5.satyg` で定義されているもの

グローバル空間に

```
type json5 =
| Json5Null
| Json5Bool of bool
| Json5Int of int
```

```
| Json5Float of float
| Json5NaN
| Json5Infinity
| Json5NegInfinity
| Json5String of string
| Json5Array of json5 list
| 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 `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 `to-string :
json -> string` という関数の 2 つが公開されています。

```
let json5-str = `
{
  // comments
  "foo" : "foo",
  /*
    block comments
```

```
    / hoge * fuga/  
  */  
  bar : {  
    "bar1" : [1,2,3,],  
    "bar2" : [  
      {"baz1" : 123, "baz2" : -Infinity},  
      {"baz3" : null, "baz4" : true, "baz" : 0xabcd}  
    ]  
  },  
}  
`~`  
  
let json5 = json5-str |> JSON5.parse |> Result.unwrap  
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。

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。