

SenML on RIOT

Base Information

```
/*! struct that contains base information */  
typedef struct {  
    uint8_t          version;  
    char             *base_name;  
    double           base_time;  
    char             *base_unit;  
    double           base_value;  
} senml_base_info_t;
```

Record

```
/*! struct that contains the values of a SenML record */  
typedef struct {  
    char                *name;  
    char                *unit;  
    double              time;  
    double              update_time;  
    double              value_sum;  
    senml_value_type_t value_type;  
    union {  
        double          f;  //!< A float value  
        char            *s;  //!< A string value  
        bool            b;   //!< A boolean value  
        char            *d;  //!< A data value  
    } value;  
} senml_record_t;
```

Pack

```
/*! struct that holds a SenML pack (optional base info and 1..n records) */  
typedef struct {  
    senml_base_info_t    *base_info;  
    senml_record_t       *records;  
    size_t               num;  
} senml_pack_t;
```

Usage - Decoding

Input:

```
[  
  { "bn": "urn:dev:ow:10e2073a01080063",  
    "bt": 1276020076.001,  
    "bu": "A",  
    "bver": 5,  
    "n": "voltage", "u": "V", "v": 120.1 },  
  { "n": "current", "t": -3, "v": 0.14e1 },  
  { "n": "current", "t": -2, "v": 1.5 },  
  { "n": "current", "t": -1, "v": 1.6 },  
  { "n": "current", "t": 0, "v": 1.7 }  
]
```

Usage - Decoding

```
senml_base_info_t base_info;  
senml_record_t    records[5];  
senml_pack_t      pack;
```

```
memset(&base_info, 0, sizeof(base_info));  
memset(records,   0, sizeof(records));
```

```
pack.base_info = base_info;  
pack.records   = records;  
pack.num       = 5;
```

```
senml_decode_json_s(input, &pack);
```

Usage - Decoding

Output:

```
base_info.version    = 5;  
base_info.base_name  = "urn:dev:ow:10e2073a01080063";  
base_info.base_time  = 1276020076.001;  
base_info.base_unit  = "A";
```

```
records[0].name      = "voltage";  
records[0].unit      = "V";  
records[0].value_type = SENML_TYPE_FLOAT;  
records[0].value.f    = 120.1;
```

...

Usage - Encoding

Input:

```
base_info.version    = 5;  
base_info.base_name  = "urn:dev:ow:10e2073a01080063";  
base_info.base_time  = 1276020076.001;  
base_info.base_unit  = "A";
```

```
records[0].name      = "voltage";  
records[0].unit       = "V";  
records[0].value_type = SENML_TYPE_FLOAT;  
records[0].value.f    = 120.1;
```

...

Usage - Encoding

```
char output[512];
```

```
senml_encode_json_s(&pack, output, 512);
```

Output:

```
[  
  { ...  
    "n": "voltage", "u": "V", "v": 120.100000 },  
  { "n": "current", "t": -3.000000, "v": 0.140000 },  
  ...  
]
```

Issues

JSON:

optimize representation of integers and floats (trailing zeros)

Issues

JSON:

optimize representation of integers and floats (trailing zeros)

CBOR:

optimize representation of integers and floats (trailing zeros)

Issues

JSON:

optimize representation of integers and floats (trailing zeros)

CBOR:

optimize representation of integers and floats (trailing zeros)
definite or indefinite length arrays and maps?

Issues

JSON:

optimize representation of integers and floats (trailing zeros)

CBOR:

optimize representation of integers and floats (trailing zeros)
definite or indefinite length arrays and maps?

XML / EXI:

not aware of any lightweight implementations
does anybody use this..?

Performance

```
sizeof senml_base_info_t == 28  
sizeof senml_record_t    == 44  
sizeof senml_pack_t      == 12
```

RAM usage of `senml_encode_json`:
 < 100 bytes

RAM usage of `senml_decode_json`:
 depends on length of input string and JSON library,
 here roughly 20 bytes per JSON token plus some
 20 bytes or so for the rest

Measurements for CBOR are expected to be similar

Performance

Required memory for example application:

text	data	bss	dec	hex
28368	188	2748	31304	7a48

ROM = text + data = 28 kB

RAM = data + bss = 2.9 kB

...but much this is for the JSON parser,
„pure“ SenML much less

More Information

Current Draft:

<https://tools.ietf.org/html/draft-ietf-core-senml-02>

PR #5544

<https://github.com/RIOT-OS/RIOT/pull/5544>

Check out

- /sys/include/senml.h
- /sys/senml/senml.c
- /examples/senml_json
- /tests/senml_json