parser 入門

～使ってみようlibxml2～

Knct-SG #1

@y1r96

parser is なに

parserとは

- parse (動詞)
  文法に従って分析する、品詞を記述する、構文解析する、などの意味を持つ英単語。プログラムのソースコードやXML文書など、一定の文法に従って記述された複雑な構造のテキスト文書を解析し、プログラムで扱えるようなデータ構造の集合体に変換することなどを指す。

- parserはこのparseをするもの

? ? ?

json(後述),HTML,XMLなどを
コンピュータで扱いやすいデータ構造に変換する

使用例

- Twitter Client (json parserが入ってる)
- Web Browser (HTML parserが入ってる)
- etc...

json? HTML?

json

- json(JavaScript Object Notation)
- JavaScriptにおけるオブジェクトの表記法をベースとした軽量なデータ記述言語

jsonを見てみよう

emacs


```
{
    "search_metadata": {
        "completed_in": 0.14,
        "count": 10,
        "max_id": 397990857845903360,
        "max_id_str": "397990857845903360",
        "next_results": "?max_id=397975510593986559&q=y1r96&lang=ja&count=10&include_entities=1",
        "query": "y1r96",
        "refresh_url": "?since_id=397990857845903360&q=y1r96&lang=ja&include_entities=1",
        "since_id": 0,
        "since_id_str": "0"
    },
    "statuses": [
        {
            "contributors": null,
            "coordinates": null,
            "created_at": "Wed Nov 06 07:36:29 +0000 2013",
            "entities": {
                "hashtags": [],
                "symbols": [],
                "urls": [
                    {
                        "display_url": "xmlsoft.org/tutorial/",
                        "expanded_url": "http://xmlsoft.org/tutorial/",
                        "indices": [
                            16,
                            38
                        ],
                        "url": "http://t.co/DKU5S4OMr5"
                    }
                ],
                "user_mentions": []
            },
            "favorite_count": 0,
            "favorited": false,
            "geo": null,
            "id": 397990857845903360,
            "id_str": "397990857845903360",
            "in_reply_to_screen_name": null,
            "in_reply_to_status_id": null,
            "in_reply_to_status_id_str": null,
            "in_reply_to_user_id": null,
            "in_reply_to_user_id_str": null,
            "lang": "ja",
            "metadata": {
                "iso_language_code": "ja",
                "result_type": "recent"
            },
            "place": null,
            "possibly_sensitive": false,
            "retweet_count": 0,
            "retweeted": false,
            "source": "<a href="http://yoshika23218.com/" rel="nofollow">twitcle plus</a>",
            "text": "Libxml Tutorial http://t.co/DKU5S4OMr5 分かりやすい。何とかなりそうな気がしてきた。\
",
            "truncated": false,
            "user": {
                "contributors_enabled": false,
                "created_at": "Mon Oct 19 12:59:15 +0000 2009",
                "default_profile": false,
                "default_profile_image": false,
                "description": "高専生",
                "entities": {
                    "description": {
                        "urls": []
                    },
                    "url": {
                        "urls": [
                            {
                                "display_url": "blog.y1r.org",
                                "expanded_url": "http://blog.y1r.org/",
                                "indices": [
                                    0,
                                    22
                                ],
                                "url": "http://t.co/9LGQNdk4vN"
```


-UUU:----F1 twitter.json Top (1,0) (Javascript)

parseする #とは

parse しよう

例えばPHP

$jset = json_decode($json);

↓

echo $jset[“search_metadata”][“count”];

↓

10

emacs


```
{
    "search_metadata": {
        "completed_in": 0.14,
        "count": 10,
        "max_id": 397990857845903360,
        "max_id_str": "397990857845903360",
        "next_results": "?max_id=397975510593986559&q=y1r96&lang=ja&count=10&include_entities=1",
        "query": "y1r96",
        "refresh_url": "?since_id=397990857845903360&q=y1r96&lang=ja&include_entities=1",
        "since_id": 0,
        "since_id_str": "0"
    },
    "statuses": [
        {
            "contributors": null,
            "coordinates": null,
            "created_at": "Wed Nov 06 07:36:29 +0000 2013",
            "entities": {
                "hashtags": [],
                "symbols": [],
                "urls": [
                    {
                        "display_url": "xmlsoft.org/tutorial/",
                        "expanded_url": "http://xmlsoft.org/tutorial/",
                        "indices": [
                            16,
                            38
                        ],
                        "url": "http://t.co/DKU5S4OMr5"
                    }
                ],
                "user_mentions": []
            },
            "favorite_count": 0,
            "favorited": false,
            "geo": null,
            "id": 397990857845903360,
            "id_str": "397990857845903360",
            "in_reply_to_screen_name": null,
            "in_reply_to_status_id": null,
            "in_reply_to_status_id_str": null,
```

＿人人人人人＿
＞　すごい　＜
￣Y^Y^Y^Y^Y￣

Parserの種類

- DOM (Document Object Model) API
- SAX (Simple API for XML)

DOM API

- Document Object Model をつくる
- さっきのやつ
- 全体をparseして、あとから要素にaccessする
- 全体のツリー構造を作るのでメモリを食う
- 複雑な動作を実行しやすい

SAX

- Simple API for XML
- 最初から順番にparseし、
  要素を見つけるたびに処理を行う
- 省メモリ
- 複雑な操作を実行しにくい

代表的なparser実装

- HTML,XML Parser libxml2 (C言語で実装)
- 色々な言語の標準ライブラリに既に実装済
  - JavaScript JSON.parse()
  - PHP json_decode()
  - etc...

parserでできること

parserでできること

- 要素を書き出し
  libxml2ならxmlSaveFormatFile()とかでできる

- DOMツリーを書き換えて、
  元のXMLに上書き保存

- etc...

libxml2をつかってみよう

libxml2

↓

HTMLをparse

↓

foo
のhogeとfooを抜き出す

動かしてみよう

- curl www.yahoo.co.jp -o index.html
- gcc getURLbyHTML.c -I/usr/local/opt/libxml2/include/libxml2 -lxml2
- ./a.out index.html

raw.github.com/y1r/Knct-SG/master/1/y1r/getURLbyHTML.c

```
libxml/tree.h
struct _xmlDoc {
    void           *_private;/* application data */
    xmlElementType  type;       /* XML_DOCUMENT_NODE, must be second ! */
    char           *name;    /* name/filename/URI of the document */
    struct _xmlNode *children;    /* the document tree */
    struct _xmlNode *last;   /* last child link */
    struct _xmlNode *parent; /* child->parent link */
    struct _xmlNode *next;   /* next sibling link  */
    struct _xmlNode *prev;   /* previous sibling link  */
    struct _xmlDoc  *doc;    /* autoreference to itself */

    /* End of common part */
    int             compression;/* level of zlib compression */
    int             standalone; /* standalone document (no external refs)
                                   1 if standalone="yes"
                                   0 if standalone="no"
                                  -1 if there is no XML declaration
                                  -2 if there is an XML declaration, but no
                                   standalone attribute was specified */
    struct _xmlDtd  *intSubset;   /* the document internal subset */
    struct _xmlDtd  *extSubset;   /* the document external subset */
    struct _xmlNs   *oldNs;  /* Global namespace, the old way */
    const xmlChar  *version; /* the XML version string */
    const xmlChar  *encoding;   /* external initial encoding, if any */
    void           *ids;        /* Hash table for ID attributes if any */
    void           *refs;       /* Hash table for IDREFs attributes if any */
    const xmlChar  *URL;/* The URI for that document */
    int             charset;    /* encoding of the in-memory content
                                   actually an xmlCharEncoding */
    struct _xmlDict *dict;      /* dict used to allocate names or NULL */
    void           *psvi;    /* for type/PSVI informations */
    int             parseFlags;   /* set of xmlParserOption used to parse the
                                   document */
    int             properties;   /* set of xmlDocProperties for this document
                                   set at the end of parsing */
};
```

```
libxml/tree.h
typedef struct _xmlNode xmlNode;
typedef xmlNode *xmlNodePtr;
struct _xmlNode {
    void           *_private;  /* application data */
    xmlElementType   type; /* type number, must be second ! */
    const xmlChar   *name;      /* the name of the node, or the entity */
    struct _xmlNode *children; /* parent->childs link */
    struct _xmlNode *last; /* last child link */
    struct _xmlNode *parent;   /* child->parent link */
    struct _xmlNode *next; /* next sibling link  */
    struct _xmlNode *prev; /* previous sibling link  */
    struct _xmlDoc  *doc;  /* the containing document */

    /* End of common part */
    xmlNs           *ns;        /* pointer to the associated namespace */
    xmlChar         *content;   /* the content */
    struct _xmlAttr *properties;/* properties list */
    xmlNs           *nsDef;     /* namespace definitions on this node */
    void            *psvi; /* for type/PSVI informations */
    unsigned short   line; /* line number */
    unsigned short   extra;/* extra data for XPath/XSLT */
};
```

```
libxml/xpath.h
typedef struct _xmlXPathObject xmlXPathObject;
typedef xmlXPathObject *xmlXPathObjectPtr;
struct _xmlXPathObject {
    xmlXPathObjectType type;
    xmlNodeSetPtr nodesetval;
    int boolval;
    double floatval;
    xmlChar *stringval;
    void *user;
    int index;
    void *user2;
    int index2;
};
```

libxml/xpath.h

```
struct _xmlXPathContext {
    xmlDocPtr doc;                      /* The current document */
    xmlNodePtr node;                    /* The current node */

    int nb_variables_unused;            /* unused (hash table) */
    int max_variables_unused;           /* unused (hash table) */
    xmlHashTablePtr varHash;            /* Hash table of defined variables */

    int nb_types;                       /* number of defined types */
    int max_types;                      /* max number of types */
    xmlXPathTypePtr types;              /* Array of defined types */

    int nb_funcs_unused;         /* unused (hash table) */
    int max_funcs_unused;        /* unused (hash table) */
    xmlHashTablePtr funcHash;           /* Hash table of defined funcs */

    int nb_axis;                 /* number of defined axis */
    int max_axis;                       /* max number of axis */
    xmlXPathAxisPtr axis;        /* Array of defined axis */

    /* the namespace nodes of the context node */
    xmlNsPtr *namespaces;        /* Array of namespaces */
    int nsNr;                           /* number of namespace in scope */
    void *user;                         /* function to free */

    /* extra variables */
    int contextSize;                    /* the context size */
    int proximityPosition;              /* the proximity position */

    /* extra stuff for XPointer */
    int xptr;                           /* is this an XPointer context? */
    xmlNodePtr here;                    /* for here() */
    xmlNodePtr origin;                  /* for origin() */

    /* the set of namespace declarations in scope for the expression */
    xmlHashTablePtr nsHash;              /* The namespaces hash table */
    xmlXPathVariableLookupFunc varLookupFunc;/* variable lookup func */
    void *varLookupData;         /* variable lookup data */

    /* Possibility to link in an extra item */
    void *extra;                        /* needed for XSLT */

    /* The function name and URI when calling a function */
    const xmlChar *function;
    const xmlChar *functionURI;

    /* function lookup function and data */
    xmlXPathFuncLookupFunc funcLookupFunc;/* function lookup func */
    void *funcLookupData;        /* function lookup data */

    /* temporary namespace lists kept for walking the namespace axis */
    xmlNsPtr *tmpNsList;         /* Array of namespaces */
    int tmpNsNr;                 /* number of namespaces in scope */

    /* error reporting mechanism */
    void *userData;                     /* user specific data block */
    xmlStructuredErrorFunc error;       /* the callback in case of errors */
    xmlError lastError;                   /* the last error */
    xmlNodePtr debugNode;        /* the source node XSLT */

    /* dictionary */
    xmlDictPtr dict;                      /* dictionary if any */

    int flags;                            /* flags to control compilation */

    /* Cache for reusal of XPath objects */
    void *cache;
};
```

xmlChar *

- libxml2で用いられる文字
- typedef char xmlChar
- UTF-8であることを明示的に示すため？

参考

- Libxml Tutorial
  http://xmlsoft.org/tutorial/index.html
- libxml2チュートリアル
  http://wiki.livedoor.jp/koba24505/d/libxml2%A5%C1%A5%E5%A1%BC%A5%C8%A5%EA%A5%A2%A5%EB

Knct-SG #2

“未定”