

TP2_a

Generated by Doxygen 1.8.11

Contents

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

TLex	Structure contenant tous les parametres/donnees pour l'analyse lexicale	??
TSymbole	Union permettant de manipuler un entier/reel/chaine pour la table des symboles	??

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

tp2_a.c	Analyseur lexical pour le langage JSON	??
-------------------------	--	----

Chapter 3

Data Structure Documentation

3.1 TLex Struct Reference

structure contenant tous les parametres/donnees pour l'analyse lexicale

Data Fields

- char * [data](#)
- char * [startPos](#)
- int [nbLignes](#)
- [TSymbole](#) * [tableSymboles](#)
- int [nbSymboles](#)

3.1.1 Detailed Description

structure contenant tous les parametres/donnees pour l'analyse lexicale

3.1.2 Field Documentation

3.1.2.1 char* TLex::data

chaîne a parcourir

3.1.2.2 int TLex::nbLignes

nb de lignes analysees

3.1.2.3 int TLex::nbSymboles

taille du tableau tableSymboles

3.1.2.4 `char* TLex::startPos`

position de depart pour la prochaine analyse

3.1.2.5 `TSymbole* TLex::tableSymboles`

tableau des symboles : chaines/entier/reel

The documentation for this struct was generated from the following file:

- [tp2_a.c](#)

3.2 TSymbole Union Reference

union permettant de manipuler un entier/reel/chaine pour la table des symboles

Data Fields

- int `type`
- union {
 - int `entier`
 - float `reel`
 - char * `chaine`
- } `val`

3.2.1 Detailed Description

union permettant de manipuler un entier/reel/chaine pour la table des symboles

3.2.2 Field Documentation

3.2.2.1 `int TSymbole::type`

l'un des 3 types suivants : JSON_STRING/JSON_INT_NUMBER/JSON_REAL_NUMBER

3.2.2.2 `union { ... } TSymbole::val`

valeur associer a un element de la table des symboles

The documentation for this union was generated from the following file:

- [tp2_a.c](#)

Chapter 4

File Documentation

4.1 tp2_a.c File Reference

analyseur lexical pour le langage JSON

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>
#include <stdlib.h>
#include <assert.h>
```

Data Structures

- union [TSymbole](#)
union permettant de manipuler un entier/reel/chaine pour la table des symboles
- struct [TLex](#)
structure contenant tous les parametres/donnees pour l'analyse lexicale

Macros

- `#define JSON_LEX_ERROR -1`
- `#define JSON_TRUE 1`
- `#define JSON_FALSE 2`
- `#define JSON_NULL 3`
- `#define JSON_LCB 4`
- `#define JSON_RCB 5`
- `#define JSON_LB 6`
- `#define JSON_RB 7`
- `#define JSON_COMMA 8`
- `#define JSON_COLON 9`
- `#define JSON_STRING 10`
- `#define JSON_INT_NUMBER 11`
- `#define JSON_REAL_NUMBER 12`

Functions

- int `isSep` (const char `_symb`)
fonction qui teste si un symbole fait partie des separateurs
- `TLex * initLexData` (char * `_data`)
fonction qui reserve la memoire et initialise les donnees pour l'analyseur lexical
- void `deleteLexData` (`TLex **` `_lexData`)
fonction qui supprime de la memoire les donnees pour l'analyseur lexical
- void `printLexData` (`TLex *` `_lexData`)
fonction qui affiche les donnees pour l'analyseur lexical
- void `addIntSymbolToLexData` (`TLex *` `_lexData`, const int `_val`)
fonction qui ajoute un symbole entier a la table des symboles
- void `addRealSymbolToLexData` (`TLex *` `_lexData`, const float `_val`)
fonction qui ajoute un symbole reel a la table des symboles
- void `addStringSymbolToLexData` (`TLex *` `_lexData`, char * `_val`)
fonction qui ajoute une chaine de caracteres a la table des symboles
- void `parseString` (`TLex *` `_lexData`)
fonction qui parse une chaine de caractère et l'ajoute à la table des symbole
- int `parseNumber` (`TLex *` `_lexData`)
fonction qui parse un nombre (entier ou réel) et l'ajoute à la table des symbole
- int `lex` (`TLex *` `_lexData`)
- int `main` ()
fonction principale

4.1.1 Detailed Description

analyseur lexical pour le langage JSON

Author

NM

Version

0.1

Date

25/11/2015

4.1.2 Macro Definition Documentation

4.1.2.1 `#define JSON_COLON 9`

entite lexicale :

4.1.2.2 `#define JSON_COMMA 8`

entite lexicale ,

4.1.2.3 #define JSON_FALSE 2

entite lexicale false

4.1.2.4 #define JSON_INT_NUMBER 11

entite lexicale nombre entier

4.1.2.5 #define JSON_LB 6

entite lexicale [

4.1.2.6 #define JSON_LCB 4

entite lexicale {

4.1.2.7 #define JSON_LEX_ERROR -1

code d'erreur lexicale

4.1.2.8 #define JSON_NULL 3

entite lexicale null

4.1.2.9 #define JSON_RB 7

entite lexicale]

4.1.2.10 #define JSON_RCB 5

entite lexicale }

4.1.2.11 #define JSON_REAL_NUMBER 12

entite lexicale nombre reel

4.1.2.12 #define JSON_STRING 10

entite lexicale chaine de caracteres

4.1.2.13 #define JSON_TRUE 1

entite lexicale true

4.1.3 Function Documentation

4.1.3.1 void addIntSymbolToLexData (TLex * _lexData, const int _val)

fonction qui ajoute un symbole entier a la table des symboles

Parameters

--	--

4.1.3.2 void addRealSymbolToLexData (TLex * *_lexData*, const float *_val*)

fonction qui ajoute un symbole reel a la table des symboles

Parameters

--	--

4.1.3.3 void addStringSymbolToLexData (TLex * *_lexData*, char * *_val*)

fonction qui ajoute une chaine de caracteres a la table des symboles

Parameters

--	--

4.1.3.4 void deleteLexData (TLex ** *_lexData*)

fonction qui supprime de la memoire les donnees pour l'analyseur lexical

Parameters

--	--

4.1.3.5 TLex * initLexData (char * *_data*)

fonction qui reserve la memoire et initialise les donnees pour l'analyseur lexical

Parameters

in	<i>_data</i>	chaine a analyser
----	--------------	-------------------

Returns

pointeur sur la structure de donnees creee

4.1.3.6 int isSep (const char *_symb*)

fonction qui teste si un symbole fait partie des separateurs

Parameters

in	<i>_symb</i>	symbole a analyser
----	--------------	--------------------

Returns

1 (vrai) si *_symb* est un separateur, 0 (faux) sinon

4.1.3.7 int parseNumber (TLex * *_lexData*)

fonction qui parse un nombre (entier ou réel) et l'ajoute à la table des symbole

Parameters

--	--

4.1.3.8 void parseString (TLex * *_lexData*)

fonction qui parse une chaine de caractère et l'ajoute à la table des symbole

Parameters

--	--

4.1.3.9 void printLexData (TLex * *_lexData*)

fonction qui affiche les donnees pour l'analyseur lexical

Parameters

in	<i>_lexData</i>	donnees de l'analyseur lexical
----	-----------------	--------------------------------

Returns

neant

