HTTPServe

0.1

Generated by Doxygen 1.6.1

Sun Feb 14 01:15:21 2010

Contents

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Chapter 3

Class Documentation

3.1 fsm_state Struct Reference

Stores information regarding state.

`#include <fsm.h>`

Public Attributes

- char * name
- void(* function)(int, void **)
- struct fsm_state * next

3.1.1 Detailed Description

Stores information regarding state.

3.1.2 Member Data Documentation

3.1.2.1 void(* fsm_state::function)(int, void **)

stores the function pointer for the state

3.1.2.2 char* fsm_state::name

Stores the name of the state

3.1.2.3 struct fsm_state* fsm_state::next [read]

pointer to the next state

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

- fsm.h

Chapter 4

File Documentation

4.1 fsm.c File Reference

an implementation for a FSM in C for HTTPServe, this file contains implementation of definations in "fsm.h". `#include "fsm.h"`

```
#include <stdlib.h>
```

```
#include <string.h>
```

Functions

- int fsm_init ()
- int fsm_main (char *def_state)
- int fsm_add (char *state, void(*fun)(int, void **))
- int fsm_remove (char *state)
- int fsm_to_state (char *state, int num, void **arg)
- int fsm_default (char *state, void(*fun)(int, void **))
- void fsm_terminate ()

4.1.1 Detailed Description

an implementation for a FSM in C for HTTPServe, this file contains implementation of definations in "fsm.h".

Author:

Ankur Shrivastava

4.1.2 Function Documentation

4.1.2.1 int fsm_add (char * *state*, void(*)(int, void **) *fun*)

Function to add a new state to the FSM.

Parameters:

state name of the state to be added.

fun name of the function to be executed for this state

4.1.2.2 int fsm_default (char * *state*, void(*)(int, void **) *fun*)

Function to add a default state to FSM.

Adds a default state to FSM, this is the function called at the start of the FSM or in case of error, with the appropriate error code

Parameters:

state name of the state to be added.

fun name of the function to be executed for this state

4.1.2.3 int fsm_init ()

Function to initialize the FSM

4.1.2.4 int fsm_main (char * *def_state*)

The FSM entry point, this is where execution of code begins in FSM.

Parameters:

def_state the default state of FSM

4.1.2.5 int fsm_remove (char * *state*)

Function to remove a state from the FSM.

Parameters:

state name of state to be removed

4.1.2.6 void fsm_terminate ()

Function for FSM termination

4.1.2.7 int fsm_to_state (char * *state*, int *num*, void ** *arg*)

Function to change state.

changes state to the new specified state, if the state does not exist returns error, state change is not triggered till function calling fsm_to_state returns

Parameters:

state name of state to chnage to

num number of arguments

arg arguments

4.2 fsm.h File Reference

an implementation for a FSM in C for HTTPServe, this file contains all definations required for FSM.

Classes

- struct fsm_state

  Stores information regarding state.

Functions

- int fsm_init ()
- int fsm_main ()
- int fsm_add (char *state, void(*fun)(int, void **))
- int fsm_default (char *state, void(*fun)(int, void **))
- int fsm_remove (char *state)
- int fsm_to_state (char *state, int num, void **arg)
- void fsm_terminate ()

Variables

- struct fsm_state * fsm_base
- char * fsm_cur_state
- int fsm_arg_num
- void ** fsm_arg_value

4.2.1 Detailed Description

an implementation for a FSM in C for HTTPServe, this file contains all definations required for FSM.

Author:

Ankur Shrivastava

4.2.2 Function Documentation

4.2.2.1 int fsm_add (char * *state*, void(*)(int, void **) *fun*)

Function to add a new state to the FSM.

Parameters:

state name of the state to be added.

fun name of the function to be executed for this state

4.2.2.2 int fsm_default (char * *state*, void(*)(int, void **) *fun*)

Function to add a default state to FSM.

Adds a default state to FSM, this is the function called at the start of the FSM or in case of error, with the appropriate error code

Parameters:

state name of the state to be added.

fun name of the function to be executed for this state

4.2.2.3 int fsm_init ()

Function to initialize the FSM

4.2.2.4 int fsm_main ()

The FSM entry point, this is where execution of code begins in FSM.

4.2.2.5 int fsm_remove (char * *state*)

Function to remove a state from the FSM.

Parameters:

state name of state to be removed

4.2.2.6 void fsm_terminate ()

Function for FSM termination

4.2.2.7 int fsm_to_state (char * *state*, int *num*, void ** *arg*)

Function to change state.

changes state to the new specified state, if the state does not exist returns error, state change is not triggered till function calling fsm_to_state returns

Parameters:

state name of state to chnage to

num number of arguments

arg arguments

4.2.3 Variable Documentation

4.2.3.1 int fsm_arg_num

stores the number of argument passed to the nest state

4.2.3.2 void** fsm_arg_value

stores the values of arguments passed to the next state

4.2.3.3 struct fsm_state * fsm_base

base pointer, storing the default state information

4.2.3.4 char* fsm_cur_state

stores pointer of current FSM state

4.3 main.c File Reference

```
#include "fsm.h"
#include <stdio.h>
```

Functions

- void abc (int val, void **arg)
- void xyz (int val, void **arg)
- int main ()

4.3.1 Function Documentation

4.3.1.1 void abc (int *val*, void ** *arg*)

4.3.1.2 int main ()

4.3.1.3 void xyz (int *val*, void ** *arg*)

Index