

# HTTPServe

0.1

Generated by Doxygen 1.6.1

Sun Feb 14 01:15:21 2010



# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class List . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	fsm_state Struct Reference . . . . .	5
3.1.1	Detailed Description . . . . .	5
3.1.2	Member Data Documentation . . . . .	5
3.1.2.1	function . . . . .	5
3.1.2.2	name . . . . .	5
3.1.2.3	next . . . . .	5
<b>4</b>	<b>File Documentation</b>	<b>7</b>
4.1	fsm.c File Reference . . . . .	7
4.1.1	Detailed Description . . . . .	7
4.1.2	Function Documentation . . . . .	7
4.1.2.1	fsm_add . . . . .	7
4.1.2.2	fsm_default . . . . .	8
4.1.2.3	fsm_init . . . . .	8
4.1.2.4	fsm_main . . . . .	8
4.1.2.5	fsm_remove . . . . .	8
4.1.2.6	fsm_terminate . . . . .	8
4.1.2.7	fsm_to_state . . . . .	8
4.2	fsm.h File Reference . . . . .	9
4.2.1	Detailed Description . . . . .	9
4.2.2	Function Documentation . . . . .	9
4.2.2.1	fsm_add . . . . .	9

4.2.2.2	fsm_default	10
4.2.2.3	fsm_init	10
4.2.2.4	fsm_main	10
4.2.2.5	fsm_remove	10
4.2.2.6	fsm_terminate	10
4.2.2.7	fsm_to_state	10
4.2.3	Variable Documentation	10
4.2.3.1	fsm_arg_num	10
4.2.3.2	fsm_arg_value	11
4.2.3.3	fsm_base	11
4.2.3.4	fsm_cur_state	11
4.3	main.c File Reference	12
4.3.1	Function Documentation	12
4.3.1.1	abc	12
4.3.1.2	main	12
4.3.1.3	xyz	12

# Chapter 1

## Class Index

### 1.1 Class List

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

[fsm\\_state](#) (Stores information regarding state ) . . . . . 5



# Chapter 2

## File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

<a href="#">fsm.c</a> (Implementation for a FSM in C for HTTPServe, this file contains implementation of definitions in "fsm.h" ) . . . . .	7
<a href="#">fsm.h</a> (Implementation for a FSM in C for HTTPServe, this file contains all definitions required for FSM ) . . . . .	9
<a href="#">main.c</a> . . . . .	12





# 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 definitions in "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 definitions 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 change 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 change 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 next 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

- abc
  - main.c, [12](#)
- fsm.c, [7](#)
  - fsm\_add, [7](#)
  - fsm\_default, [8](#)
  - fsm\_init, [8](#)
  - fsm\_main, [8](#)
  - fsm\_remove, [8](#)
  - fsm\_terminate, [8](#)
  - fsm\_to\_state, [8](#)
- fsm.h, [9](#)
  - fsm\_add, [9](#)
  - fsm\_arg\_num, [10](#)
  - fsm\_arg\_value, [10](#)
  - fsm\_base, [11](#)
  - fsm\_cur\_state, [11](#)
  - fsm\_default, [9](#)
  - fsm\_init, [10](#)
  - fsm\_main, [10](#)
  - fsm\_remove, [10](#)
  - fsm\_terminate, [10](#)
  - fsm\_to\_state, [10](#)
- fsm\_add
  - fsm.c, [7](#)
  - fsm.h, [9](#)
- fsm\_arg\_num
  - fsm.h, [10](#)
- fsm\_arg\_value
  - fsm.h, [10](#)
- fsm\_base
  - fsm.h, [11](#)
- fsm\_cur\_state
  - fsm.h, [11](#)
- fsm\_default
  - fsm.c, [8](#)
  - fsm.h, [9](#)
- fsm\_init
  - fsm.c, [8](#)
  - fsm.h, [10](#)
- fsm\_main
  - fsm.c, [8](#)
  - fsm.h, [10](#)
- fsm\_remove
  - fsm.c, [8](#)
  - fsm.h, [10](#)
- fsm\_state, [5](#)
- function, [5](#)
  - name, [5](#)
  - next, [5](#)
- fsm\_terminate
  - fsm.c, [8](#)
  - fsm.h, [10](#)
- fsm\_to\_state
  - fsm.c, [8](#)
  - fsm.h, [10](#)
- function
  - fsm\_state, [5](#)
- main
  - main.c, [12](#)
- main.c, [12](#)
  - abc, [12](#)
  - main, [12](#)
  - xyz, [12](#)
- name
  - fsm\_state, [5](#)
- next
  - fsm\_state, [5](#)
- xyz
  - main.c, [12](#)