

My Project

Generated by Doxygen 1.8.17

1 Class Index	1
1.1 Class List	1
2 File Index	3
2.1 File List	3
3 Class Documentation	5
3.1 action Struct Reference	5
3.2 action_t Struct Reference	5
3.2.1 Detailed Description	5
3.3 semaine Struct Reference	6
4 File Documentation	7
4.1 action.h File Reference	7
4.1.1 Detailed Description	8
4.1.2 Macro Definition Documentation	9
4.1.2.1 listAction_t	9
4.1.3 Function Documentation	9
4.1.3.1 checkDay()	9
4.1.3.2 checkHour()	9
4.1.3.3 compareAction()	10
4.1.3.4 equalAction()	10
4.1.3.5 findAction()	10
4.1.3.6 freeAction()	11
4.1.3.7 freeListAction()	11
4.1.3.8 initListAction()	12
4.1.3.9 insertAction()	12
4.1.3.10 newAction()	12
4.1.3.11 prettyPrintListAction()	13
4.1.3.12 suprAction()	13
Index	15

Chapter 1

Class Index

1.1 Class List

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

action	5
action_t	5
semaine	6

Chapter 2

File Index

2.1 File List

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

action.h	Header pour la structure SDD action	7
liste.h	??
semaine.h	??

Chapter 3

Class Documentation

3.1 action Struct Reference

Collaboration diagram for action:



Public Attributes

- char **jour**
- char **heure** [3]
- char **nom** [10]
- struct [action](#) * **suiv**

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

- [action.h](#)

3.2 action_t Struct Reference

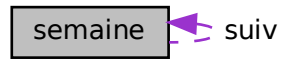
3.2.1 Detailed Description

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

- [action.h](#)

3.3 semaine Struct Reference

Collaboration diagram for semaine:



Public Attributes

- char **annee** [5]
- char **numSemaine** [3]
- struct [semaine](#) * **suiv**
- [listAction_t](#) **listeAction**

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

- liste.h
- semaine.h

Chapter 4

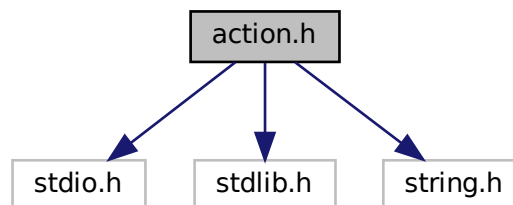
File Documentation

4.1 action.h File Reference

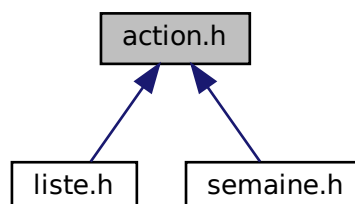
header pour la structure SDD action

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

Include dependency graph for action.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct [action](#)

Macros

- `#define listAction_t action_t *`

Typedefs

- `typedef struct action action_t`

Functions

- [listAction_t](#) [initListAction](#) ()
Fonction créant une nouvelle instance d'une liste d'action.
- [action_t](#) * [newAction](#) (char jour, char heure[2], char nom[10])
- void [freeAction](#) ([action_t](#) *act)
libère l'action (simple free)
- void [freeListAction](#) ([listAction_t](#) listAct)
libère la liste fournit en paramètre
- void [insertAction](#) ([action_t](#) *nouvAction, [listAction_t](#) *la)
insert l'action nouvAction dans la liste listAct en respectant l'ordre
- [action_t](#) ** [findAction](#) (char heure[2], char jour, [listAction_t](#) *la)
recherche l'adresse de l'action avec l'heure et le jour donnait en argument
- void [suprAction](#) ([listAction_t](#) *la, char heure[2], char jour)
supprime l'action du jour et heure indiqué dans la liste listAct
- int [compareAction](#) ([action_t](#) *a, char heure[2], char jour)
retourne 1 si l'action est plus tard ou égale à l'heure et le jour donné en argument
- int [equalAction](#) ([action_t](#) *a, char heure[2], char jour)
retourne 1 si l'action est égale à l'heure et le jour donné en argument
- int [checkDay](#) (char jour)
Vérifie si le jour est cohérent.
- int [checkHour](#) (char heure[2])
Vérifie si l'heure est cohérent.
- void [prettyPrintListAction](#) ([listAction_t](#) listAct)
Affichage de la liste d'action de façon plus lisible.

4.1.1 Detailed Description

header pour la structure SDD action

Author

Marie Bornet et Cliquot Théo

Date

15 février 2022

Header contenant toute les fonctions de bases pour la structure action (Init, new, find, insert, supr ...)

4.1.2 Macro Definition Documentation

4.1.2.1 listAction_t

```
#define listAction_t action_t *
```

Simplification, une liste d'action est en réalité un pointeur vers l'action qui correspond à la tête de liste

4.1.3 Function Documentation

4.1.3.1 checkDay()

```
int checkDay (  
    char jour )
```

Vérifie si le jour est cohérent.

Parameters

<i>jour</i>	: le jour à vérifié
-------------	---------------------

Returns

1 si cohérent, 0 sinon

4.1.3.2 checkHour()

```
int checkHour (  
    char heure[2] )
```

Vérifie si l'heure est cohérent.

Parameters

<i>jour</i>	: l'heure à vérifié
-------------	---------------------

Returns

1 si cohérent, 0 sinon

4.1.3.3 compareAction()

```
int compareAction (
    action_t * a,
    char heure[2],
    char jour )
```

retourne 1 si l'action est plus tard ou égale à l'heure et le jour donné en argument

Parameters

<i>act</i>	: action à comparer
<i>heure</i>	: heure de référence
<i>jour</i>	: jour de référence

Returns

1 si $act \geq$, 0 sinon

4.1.3.4 equalAction()

```
int equalAction (
    action_t * a,
    char heure[2],
    char jour )
```

retourne 1 si l'action est égale à l'heure et le jour donné en argument

Parameters

<i>act</i>	: action à comparer
<i>heure</i>	: heure de référence
<i>jour</i>	: jour de référence

Returns

1 si $act \geq$, 0 sinon

4.1.3.5 findAction()

```
action_t** findAction (
    char heure[2],
    char jour,
    listAction_t * la )
```

recherche l'adresse de l'action avec l'heure et le jour donnait en argument

Parameters

<i>heure</i>	: l'heure (00 à 24),
<i>jour</i>	: le jour (1 à 7),
<i>listAct</i>	: la liste où il faut chercher

Returns

un pointeur de pointeur vers la cellule précédent celle recherché où la première plus grande, NULL si heure et jour ne sont pas correcte

4.1.3.6 freeAction()

```
void freeAction (
    action_t * a )
```

libère l'action (simple free)

Parameters

<i>act</i>	: action à free
------------	-----------------

Returns

void

4.1.3.7 freeListAction()

```
void freeListAction (
    listAction_t a )
```

libère la liste fournit en paramètre

Parameters

<i>listAct</i>	: la liste a libéré
----------------	---------------------

Returns

void

4.1.3.8 initListAction()

```
listAction_t initListAction ( )
```

Fonction créant une nouvelle instance d'une liste d'action.

Parameters

<i>None</i>	
-------------	--

Returns

Null (liste d'action vide)

4.1.3.9 insertAction()

```
void insertAction (
    action_t * nouvAction,
    listAction_t * la )
```

insert l'action nouvAction dans la liste listAct en respectant l'ordre

Parameters

<i>nouvAction</i>	: l'action à insérer
<i>listAct</i>	: la liste où il faut insérer

Returns

void

4.1.3.10 newAction()

```
action_t* newAction (
    char jour,
    char heure[2],
    char nom[10] )
```

Parameters

<i>jour</i>	: jour de l'action,
<i>heure</i>	: heure de l'action,
<i>nom</i>	: nom de l'action

Returns

l'action remplie avec les arguments donnée

4.1.3.11 prettyPrintListAction()

```
void prettyPrintListAction (
    listAction_t listAct )
```

Affichage de la liste d'action de façon plus lisible.

Parameters

<i>listAct</i>	: la liste à afficher
----------------	-----------------------

Returns

void

EFFET DE BORD ::: Print la liste

4.1.3.12 suprAction()

```
void suprAction (
    listAction_t * la,
    char heure[2],
    char jour )
```

supprime l'action du jour et heure indiqué dans la liste listAct

Parameters

<i>jour</i>	: le jour de l'action à supprimer (0 à 7)
<i>heure</i>	: l'heure de l'action à supprimer (0 à 24)
<i>listAct</i>	: la liste d'action ou il faut supprimer

Returns

void

Si l'action n'est pas contenu dans la liste, la fonction ne fera rien dans ce cas

Index

- action, [5](#)
- action.h, [7](#)
 - checkDay, [9](#)
 - checkHour, [9](#)
 - compareAction, [9](#)
 - equalAction, [10](#)
 - findAction, [10](#)
 - freeAction, [11](#)
 - freeListAction, [11](#)
 - initListAction, [11](#)
 - insertAction, [12](#)
 - listAction_t, [9](#)
 - newAction, [12](#)
 - prettyPrintListAction, [13](#)
 - suprAction, [13](#)
- action_t, [5](#)

- checkDay
 - action.h, [9](#)
- checkHour
 - action.h, [9](#)
- compareAction
 - action.h, [9](#)

- equalAction
 - action.h, [10](#)

- findAction
 - action.h, [10](#)
- freeAction
 - action.h, [11](#)
- freeListAction
 - action.h, [11](#)

- initListAction
 - action.h, [11](#)
- insertAction
 - action.h, [12](#)

- listAction_t
 - action.h, [9](#)

- newAction
 - action.h, [12](#)

- prettyPrintListAction
 - action.h, [13](#)

- semaine, [6](#)
- suprAction
 - action.h, [13](#)