

Unittests

Создано системой Doxygen 1.9.4

1 Иерархический список классов	1
1.1 Иерархия классов	1
2 Алфавитный указатель классов	3
2.1 Классы	3
3 Список файлов	5
3.1 Файлы	5
4 Классы	7
4.1 Класс MyTestReporter	7
4.1.1 Подробное описание	8
4.1.2 Методы	8
4.1.2.1 ReportFailure()	8
4.1.2.2 ReportSummary()	8
4.1.2.3 ReportTestFinish()	9
4.1.2.4 ReportTestStart()	9
5 Файлы	11
5.1 Файл main.cpp	11
5.1.1 Подробное описание	12
5.1.2 Функции	12
5.1.2.1 main()	12
Предметный указатель	15

Глава 1

Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

UnitTest::TestReporter	
MyTestReporter	7

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

[MyTestReporter](#)

Класс логгера для вывода детализированной информации о тестах [7](#)

Глава 3

Список файлов

3.1 Файлы

Полный список документированных файлов.

[main.cpp](#)

Тесты для проверки функциональности различных компонентов 11

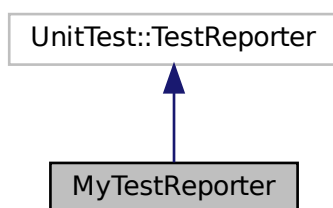
Глава 4

Классы

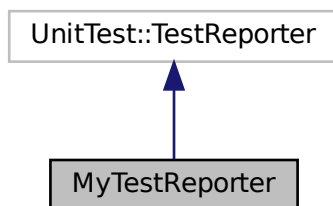
4.1 Класс MyTestReporter

Класс логгера для вывода детализированной информации о тестах.

Граф наследования:MyTestReporter:



Граф связей класса MyTestReporter:



Открытые члены

- void [ReportTestStart](#) (UnitTest::TestDetails const &test) override
Сообщает о начале выполнения теста.
- void [ReportTestFinish](#) (UnitTest::TestDetails const &test, float secondsElapsed) override
Сообщает об окончании выполнения теста.
- void [ReportFailure](#) (UnitTest::TestDetails const &test, char const *failure) override
Сообщает о сбое теста.
- void [ReportSummary](#) (int totalTestCount, int failedTestCount, int failureCount, float secondsElapsed) override
Сообщает об общей сводке выполнения тестов.

4.1.1 Подробное описание

Класс логгера для вывода детализированной информации о тестах.

4.1.2 Методы

4.1.2.1 ReportFailure()

```
void MyTestReporter::ReportFailure (
    UnitTest::TestDetails const & test,
    char const * failure ) [inline], [override]
```

Сообщает о сбое теста.

Аргументы

test	Детали теста.
failure	Описание сбоя.

4.1.2.2 ReportSummary()

```
void MyTestReporter::ReportSummary (
    int totalTestCount,
    int failedTestCount,
    int failureCount,
    float secondsElapsed ) [inline], [override]
```

Сообщает об общей сводке выполнения тестов.

Аргументы

totalTestCount	Общее количество тестов.
----------------	--------------------------

Аргументы

failedTestCount	Количество неудачных тестов.
failureCount	Общее количество сбоев.
secondsElapsed	Общее время выполнения тестов в секундах.

4.1.2.3 ReportTestFinish()

```
void MyTestReporter::ReportTestFinish (
    UnitTest::TestDetails const & test,
    float secondsElapsed ) [inline], [override]
```

Сообщает об окончании выполнения теста.

Аргументы

test	Детали теста.
secondsElapsed	Время выполнения теста в секундах.

4.1.2.4 ReportTestStart()

```
void MyTestReporter::ReportTestStart (
    UnitTest::TestDetails const & test ) [inline], [override]
```

Сообщает о начале выполнения теста.

Аргументы

test	Детали теста.
------	---------------

Объявления и описания членов класса находятся в файле:

- [main.cpp](#)

Глава 5

Файлы

5.1 Файл main.cpp

Тесты для проверки функциональности различных компонентов.

```
#include <UnitTest++/UnitTest++.h>
#include "../server/source/modules/crutils.h"
#include "../server/source/modules/network.h"
#include "../server/source/modules/interface.h"
#include "../server/source/modules/exceptions.h"
#include <iostream>
#include <fstream>
#include <map>
```

Граф включаемых заголовочных файлов для main.cpp:



Классы

- class `MyTestReporter`

Класс логгера для вывода детализированной информации о тестах.

Функции

- TEST (InterfaceInitTest)
Тест для проверки парсинга аргументов командной строки.
- TEST (InvalidArgsTest)
Тест для проверки обработки исключений при неверных аргументах.
- TEST (GenerateSaltTest)
Тест для проверки генерации соли.
- TEST (CalculateHashTest)

- Тест для проверки вычисления хеша.
 - TEST (CalculateHashKnownDataTest)
Дополнительный тест для проверки корректности вычисления хеша с известными данными.
 - TEST (CalculateHashDifferentDataTest)
Тест для проверки, что хеш не повторяется на разных данных.
 - TEST (CalculateHashSameDataTest)
Тест для проверки, что хеш повторяется на одинаковых данных.
 - TEST (GenerateDifferentSaltTest)
Тест для проверки, что соль генерируется разная.
 - TEST (NetworkInitializationTest)
Тест для проверки создания сокета и ожидания соединений.
 - TEST (NetworkSuccessTest)
Тест для проверки успешного взаимодействия.
- int [main](#) (int argc, char *argv[])
Главная функция тестирования.

5.1.1 Подробное описание

Тесты для проверки функциональности различных компонентов.

Этот файл содержит тесты для проверки парсинга аргументов командной строки, генерации соли, вычисления хеша, и работы сетевого взаимодействия.

Дата

23.11.2024

Версия

1.0

Автор

Мамелин Д. А.

5.1.2 Функции

5.1.2.1 main()

```
int main (  
    int argc,  
    char * argv[] )
```

Главная функция тестирования.

Инициализирует объект [MyTestReporter](#) и запускает тесты с использованием `UnitTest++`.

Аргументы

argc	Количество аргументов командной строки.
argv	Аргументы командной строки.

Возвращает

Код завершения программы. 0 - успешное завершение, 1 - ошибка.

Предметный указатель

main

main.cpp, [12](#)

main.cpp, [11](#)

main, [12](#)

MyTestReporter, [7](#)

ReportFailure, [8](#)

ReportSummary, [8](#)

ReportTestFinish, [9](#)

ReportTestStart, [9](#)

ReportFailure

MyTestReporter, [8](#)

ReportSummary

MyTestReporter, [8](#)

ReportTestFinish

MyTestReporter, [9](#)

ReportTestStart

MyTestReporter, [9](#)