

ERRORS MODULE

The module documentation

A module for generating program errors

Author

Rômulo Peres de Moraes

Table of Contents

1. The purposes of this document.....	3
2. Errors module.....	3
3. The layout of the documentation and the module.....	3
4. Interfaces.....	3
4.1. Public interfaces.....	3
char* generateErrorMessage(ErrorID errorID, struct ErrorMessageArguments arguments, char* errorOutput);.....	3

1. The purposes of this document

This document has the purpose of describe the module and its interfaces with the goal of improve the Developer Experience (DX) once the developers have to create new features or even for maintenance.

2. Errors module

The errors module is a component from the Rclock that store and generate all errors that might happen on run-time. This module works using errors ID that are available in a program scope, when a module reports an error, the error is an ID that is useful to fetch the message using this module.

3. The layout of the documentation and the module

The module is divided into two parts, the public code and the private code, they will be placed inside the directories public/ and private/ respectively. All private interface names shall begin with an underscore. The public interfaces may use the private interfaces, the private interfaces may use another private interfaces, however, a private interface can't use a public interface.

4. Interfaces

4.1. Public interfaces

char* generateErrorMessage(ErrorID errorID, struct ErrorMessageArguments arguments, char* errorOutput);

Purposes:

This function receives an error ID and generate its meaning.

Postconditions:

The generated string is available in the ***errorOutput**, the same pointer is also returned from the function.

Special considerations:

The errors are stored at the beginning of the module's public file, they are fetched based on their Ids.

The **ErrorMessageArguments** is a struct that makes the error message more customizable and it is mainly used when the given color for a specific clock's digit is unavailable.

An error that doesn't require an **ErrorMessageArguments** struct can pass the macro **USELESS_ERROR_MESSAGE_ARGUMENTS**, it is a uninitialized struct that is ignored.

4.1. Private interfaces