Program Listing for File message.hpp

Return to documentation for file (`include/holoscan/core/message.hpp`)

```
/* * SPDX-FileCopyrightText: Copyright (c) 2022-2023 NVIDIA CORPORATION & AFFILIATES. All rights reserved. * SPDX-License-Identifier: Apache-2.0 * * Licensed under the Apache License, Version 2.0 (the "License"); * you may not use this file except in compliance with the License. * You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License. */ #ifndef HOLOSCAN_CORE_MESSAGE_HPP #define HOLOSCAN_CORE_MESSAGE_HPP #include <any> #include <memory> #include <utility> #include "./common.hpp" namespace holoscan { class Message { public: Message() = default; template <typename typeT, typename = std::enable_if_t<!std::is_same_v<std::decay_t<typeT>, Message>>> explicit Message(typeT&& value) : value_(std::forward<typeT>(value)) {} template <typename ValueT> void set_value(ValueT&& value) { value_ = std::forward<ValueT>(value); } std::any value() const { return value_; } template <typename ValueT> std::shared_ptr<ValueT> as() const { try { return std::any_cast<std::shared_ptr<ValueT>>(value_); } catch (const std::bad_any_cast& e) { HOLOSCAN_LOG_ERROR("The message doesn't have a value of type '{}': {}", typeid(std::decay_t<ValueT>).name(), e.what()); return nullptr; } } private: std::any value_; }; } // namespace holoscan #endif/* HOLOSCAN_CORE_MESSAGE_HPP */
```