Program Listing for File gxf_resource.hpp

Return to documentation for file (include/holoscan/core/gxf/gxf_resource.hpp)

/* * SPDX-FileCopyrightText: Copyright (c) 2022-2024 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_GXF_GXF_RESOURCE_HPP #define HOLOSCAN_CORE_GXF_GXF_RESOURCE_HPP #include <string> #include <gxf/core/component.hpp> #include "../resource.hpp" #include "./gxf_component.hpp" #include "./gxf_utils.hpp" namespace holoscan::gxf { class GXFResource : public holoscan::Resource, public gxf::GXFComponent { public: HOLOSCAN_RESOURCE_FORWARD_ARGS_SUPER(GXFResource, holoscan::Resource) GXFResource() = default; GXFResource(const std::string& name, nvidia::gxf::Component* component); void initialize() override; protected: *// Make GXFExecutor a friend class so it can call protected initialization methods* friend class holoscan::gxf::GXFExecutor; *// Operator::initialize_resources() needs to call add_to_graph_entity()* friend class holoscan::Operator; virtual void add_to_graph_entity(Operator* op); void set_parameters() override; bool handle_dev_id(std::optional<int32_t>& dev_id_value); std::string gxf_typename_ = "unknown_gxf_typename"; }; } *// namespace holoscan::gxf* #endif/* HOLOSCAN_CORE_GXF_GXF_RESOURCE_HPP */