



## **Program Listing for File operator\_wrapper.hpp**

[Return to documentation for file \(](#)

[gxf\\_extensions/gxf\\_holoscan\\_wrapper/operator\\_wrapper.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
GXF_HOLOSCAN_WRAPPER_OPERATOR_WRAPPER_HPP #define
GXF_HOLOSCAN_WRAPPER_OPERATOR_WRAPPER_HPP #include <list> #include
<memory> #include "holoscan/core/operator.hpp" #include
"holoscan/core/parameter.hpp" #include "operator_wrapper_fragment.hpp"
#include "gxf/std/codelet.hpp" #include "gxf/core/parameter_parser_std.hpp"
namespace holoscan::gxf { class OperatorWrapper : public nvidia::gxf::Codelet {
public: OperatorWrapper(); virtual ~OperatorWrapper() = default; virtual const char*
holoscan_typename() const = 0; gxf_result_t initialize() override; gxf_result_t
deinitialize() override; gxf_result_t registerInterface(nvidia::gxf::Registrar* registrar)
override; gxf_result_t start() override; gxf_result_t tick() override; gxf_result_t stop()
override; struct GXFParameter { nvidia::gxf::Parameter<YAML::Node> param;
holoscan::ArgType arg_type; holoscan::Parameter<void*>* param_ptr; }; protected:
std::shared_ptr<Operator> op_; OperatorWrapperFragment fragment_;
std::list<GXFParameter> parameters_; }; } // namespace holoscan::gxf #endif/*
GXF_HOLOSCAN_WRAPPER_OPERATOR_WRAPPER_HPP */
```

© Copyright 2022-2024, NVIDIA.. PDF Generated on 06/06/2024