Program Listing for File double_buffer_receiver.hpp

Return to documentation for file (`include/holoscan/core/resources/gxf/double_buffer_receiver.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_RESOURCES_GXF_DOUBLE_BUFFER_RECEIVER_HPP #define
HOLOSCAN_CORE_RESOURCES_GXF_DOUBLE_BUFFER_RECEIVER_HPP #include
<string> #include <gxf/std/double_buffer_receiver.hpp> #include "./receiver.hpp"
namespace holoscan { // Forward declarations class
AnnotatedDoubleBufferReceiver; class DoubleBufferReceiver : public Receiver {
public: HOLOSCAN_RESOURCE_FORWARD_ARGS_SUPER(DoubleBufferReceiver,
Receiver) DoubleBufferReceiver() = default; DoubleBufferReceiver(const std::string&
name, nvidia::gxf::DoubleBufferReceiver* component); DoubleBufferReceiver(const
std::string& name, AnnotatedDoubleBufferReceiver* component); const char*
gxf_typename() const override; void setup(ComponentSpec& spec) override; void
track(); nvidia::gxf::DoubleBufferReceiver* get() const; Parameter<uint64_t>
capacity_; Parameter<uint64_t> policy_; private: bool tracking_ = false; }; } //
namespace holoscan #endif/*
HOLOSCAN_CORE_RESOURCES_GXF_DOUBLE_BUFFER_RECEIVER_HPP */
```