



Program Listing for File gxf_scheduler.hpp

[Return to documentation for file \(include/holoscan/core/gxf/gxf_scheduler.hpp\)](#)

```
/* * SPDX-FileCopyrightText: Copyright (c) 2023-2024 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_SCHEDULER_HPP #define
HOLOSCAN_CORE_GXF_GXF_SCHEDULER_HPP #include <memory> #include
<string> #include <utility> #include "../scheduler.hpp" #include
"./gxf_component.hpp" #include "gxf/std/clock.hpp" #include
"../resources/gxf/clock.hpp" namespace holoscan::gxf { // note: in GFX there is also a
System class that inherits from Component // and is the parent of Scheduler class
GXFScheduler : public holoscan::Scheduler, public GXFComponent { public:
HOLOSCAN_SCHEDULER_FORWARD_ARGS_SUPER(GXFScheduler,
holoscan::Scheduler) GXFScheduler() = default; virtual std::shared_ptr<Clock> clock()
= 0; virtual const char* gxf_typename() const = 0; virtual nvidia::gxf::Clock*
gxf_clock(); protected: // Make Fragment a friend class so it can call reset_graph_entities
friend class holoscan::Fragment; void set_parameters() override; void
reset_graph_entities() override; }; } // namespace holoscan::gxf #endif/*
HOLOSCAN_CORE_GXF_GXF_SCHEDULER_HPP */
```

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