Program Listing for File ucx_context.hpp

Return to documentation for file (`include/holoscan/core/network_contexts/gxf/ucx_context.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_NETWORK_CONTEXT_GXF_UCX_CONTEXT_HPP #define
HOLOSCAN_CORE_NETWORK_CONTEXT_GXF_UCX_CONTEXT_HPP #include
<cstdint> #include <memory> #include <string> #include
<gxf/ucx/ucx_context.hpp> #include "../../gxf/gxf_network_context.hpp" #include
"../../resources/gxf/ucx_entity_serializer.hpp" namespace holoscan { class
UcxContext : public gxf::GXFNetworkContext { public:
HOLOSCAN_NETWORK_CONTEXT_FORWARD_ARGS_SUPER(UcxContext,
gxf::GXFNetworkContext) UcxContext() = default; const char* gxf_typename() const
override { return "nvidia::gxf::UcxContext"; } // // Finds transmitters and receivers
passes the network context to transmitter // // and receivers and make connection
between them // virtual Expected<void> addRoutes(const Entity& entity) = 0; // // Closes
the connection between transmitters and receivers // virtual Expected<void>
removeRoutes(const Entity& entity) = 0; std::shared_ptr<UcxEntitySerializer>
entity_serializer() { return entity_serializer_; } void setup(ComponentSpec& spec)
override; void initialize() override; nvidia::gxf::UcxContext* get() const; private:
Parameter<std::shared_ptr<UcxEntitySerializer>> entity_serializer_; // TODO: support
GPUDevice nvidia::gxf::Resource //
nvidia::gxf::Resource<nvidia::gxf::Handle<nvidia::gxf::GPUDevice>> gpu_device_; }; } //
namespace holoscan #endif/*
HOLOSCAN_CORE_NETWORK_CONTEXT_GXF_UCX_CONTEXT_HPP */
```