



Program Listing for File server.hpp

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

`include/holoscan/core/services/app_driver/server.hpp`)

```
/* * SPDX-FileCopyrightText: Copyright (c) 2023 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_SERVICES_APP_DRIVER_SERVER_HPP #define
HOLOSCAN_CORE_SERVICES_APP_DRIVER_SERVER_HPP #include
<condition_variable> #include <memory> #include <mutex> #include <string>
#include <thread> #include <unordered_map> #include <vector> #include
"holoscan/core/app_driver.hpp" // Forward declaration of gRPC server namespace
grpc { class Server; } // namespace grpc namespace holoscan::service { constexpr
int32_t kDefaultAppDriverPort = 8765; constexpr int32_t
kDefaultHealthCheckingPort = 8777; class AppDriverServer { public: explicit
AppDriverServer(holoscan::AppDriver* app_driver, bool need_driver = true, bool
need_health_check = true); virtual ~AppDriverServer(); void start(); void stop(); void
wait(); void notify(); std::unique_ptr<AppWorkerClient>& connect_to_worker(const
std::string& worker_address); bool close_worker_connection(const std::string&
worker_address); std::vector<std::string> get_worker_addresses() const; std::size_t
num_worker_connections() const; private: void run(); std::unique_ptr<grpc::Server>
server_; std::unique_ptr<std::thread> server_thread_; std::condition_variable cv_;
std::mutex mutex_; std::mutex join_mutex_; bool should_stop_ = false;
holoscan::AppDriver* app_driver_ = nullptr; bool need_driver_ = false; bool
need_health_check_ = false; std::unordered_map<std::string,
std::unique_ptr<AppWorkerClient>> worker_clients_; }; } // namespace
holoscan::service #endif/* HOLOSCAN_CORE_SERVICES_APP_DRIVER_SERVER_HPP */
```

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