Class MultiThreadScheduler

Table of contents

- Defined in File multithread_scheduler.hpp

Inheritance Relationships

Base Type

- `public holoscan::gxf::GXFScheduler` (Class GXFScheduler)

Class Documentation

class MultiThreadScheduler : public holoscan::gxf::GXFScheduler

Public Functions

template<typename ArgT, typename ...ArgsT, typename = std::enable_if_t<!std::is_base_of_v<::holoscan::Scheduler, std::decay_t<ArgT>> && (std::is_same_v<::holoscan::Arg, std::decay_t<ArgT>> || std::is_same_v<::holoscan::ArgList, std::decay_t<ArgT>>)>>
inline MultiThreadScheduler(ArgT &&arg, ArgsT&&... args)

MultiThreadScheduler() = default

inline virtual const char *gxf_typename() const override

Get the type name of the GXF scheduler.

The returned string is the type name of the GXF scheduler and is used to create the GXF scheduler.

Example: “nvidia::holoscan::GreedyScheduler”

Returns

The type name of the GXF scheduler.

inline virtual std::shared_ptr<Clock> clock() override

Get the Clock used by the scheduler.

Returns

The Clock used by the scheduler.

virtual void setup(ComponentSpec &spec) override

Define the scheduler specification.

Parameters

spec – The reference to the component specification.

virtual void initialize() override

Initialize the scheduler.

This function is called after the scheduler is created by holoscan::Fragment::make_scheduler().

inline int64_t worker_thread_number()

inline bool stop_on_deadlock()

inline int64_t check_recession_period_ms()

inline int64_t stop_on_deadlock_timeout()

inline int64_t max_duration_ms()

nvidia::gxf::MultiThreadScheduler *get() const