Class InputContext

Table of contents

- Defined in File io_context.hpp

Inheritance Relationships

Derived Type

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

Class Documentation

class InputContext

Class to hold the input context.

This class provides the interface to receive the input data from the operator.

Subclassed by holoscan::gxf::GXFInputContext

Public Functions

inline InputContext(ExecutionContext *execution_context, Operator *op, std::unordered_map<std::string, std::shared_ptr<IOSpec>> &inputs)

Construct a new InputContext object.

Parameters

- **execution_context** – The pointer to the execution context.
- **op** – The pointer to the operator that this context is associated with.
- **inputs** – The references to the map of the input specs.

inline InputContext(ExecutionContext *execution_context, Operator *op)

Construct a new InputContext object.

inputs for the InputContext will be set to op->spec()->inputs()

Parameters

- **execution_context** – The pointer to GXF execution runtime
- **op** – The pointer to the operator that this context is associated with.

inline ExecutionContext *execution_context() const

Get pointer to the execution context.

Returns

The pointer to the execution context.

inline Operator *op() const

Return the operator that this context is associated with.

Returns

The pointer to the operator.

inline std::unordered_map<std::string, std::shared_ptr<IOSpec>> &inputs() const

Return the reference to the map of the input specs.

Returns

The reference to the map of the input specs.

inline bool empty(const char *name = nullptr)

Return whether the input port has any data.

For parameters with std::vector<IOSpec*> type, if all the inputs are empty, it will return true. Otherwise, it will return false.

Parameters

name – The name of the input port to check.

Returns

True, if it has no data, otherwise false.

template<typename DataT>
inline holoscan::expected<DataT, holoscan::RuntimeError> receive(const char *name = nullptr)

Receive a message from the input port with the given name.

If the operator has a single input port, the name of the input port can be omitted.

If the input port with the given name and type (`DataT`) is available, it will return the data from the input port. Otherwise, it will return an object of the holoscan::unexpected class which will contain the error message. The error message can be access by calling the `what()` method of the holoscan::unexpected object.

Example:

```
class PingRxOp : public holoscan::ops::GXFOperator { public: HOLOSCAN_OPERATOR_FORWARD_ARGS_SUPER(PingRxOp, holoscan::ops::GXFOperator) PingRxOp() = default; void setup(OperatorSpec& spec) override { spec.input<std::shared_ptr<ValueData>>("in"); } void compute(InputContext& op_input, OutputContext&, ExecutionContext&) override { auto value = op_input.receive<std::shared_ptr<ValueData>>("in"); if (value.has_value()) { HOLOSCAN_LOG_INFO("Message received (value: {})", value->data()); } } };
```

Template Parameters

DataT – The type of the data to receive.

Parameters

name – The name of the input port to receive the data from.

Returns

The received data.

Protected Functions

inline virtual bool empty_impl(const char *name = nullptr)

The implementation of the `empty` method.

Parameters

name – The name of the input port

Returns

True if the input port is empty or by default. Otherwise, false.

inline virtual std::any receive_impl(const char *name = nullptr, bool no_error_message = false)

The implementation of the `receive` method.

Depending on the type of the data, this method receives a message from the input port with the given name.

Parameters

- **name** – The name of the input port.
- **no_error_message** – Whether to print an error message when the input port is not found.

Returns

The data received from the input port.

Protected Attributes

ExecutionContext *execution_context_ = nullptr

The execution context that is associated with.

Operator *op_ = nullptr

The operator that this context is associated with.

std::unordered_map<std::string, std::shared_ptr<IOSpec>> &inputs_

The inputs.