



## **Class Resource**

# Table of contents

Inheritance Relationships

---

Class Documentation

---

- Defined in [File resource.hpp](#)

## Inheritance Relationships

### Base Type

- `public holoscan::Component` ([Class Component](#))

### Derived Types

- `public holoscan::Endpoint` ([Class Endpoint](#))
- `public holoscan::gxf::GXFResource` ([Class GXFResource](#))

## Class Documentation

class Resource : public holoscan::[Component](#)

Base class for all resources.

Resources such as system memory or a GPU memory pool that an [Operator](#) needs to perform its job. Resources are allocated during the initialization phase of the application. This matches the semantics of GXF's Memory [Allocator](#) or any other components derived from the [Component](#) class in GXF.

Subclassed by [holoscan::Endpoint](#), [holoscan::gxf::GXFResource](#)

Public Types

enum class ResourceType

[Resource](#) type used for the initialization of the resource.

*Values:*

enumerator kNative

Native resource.

enumerator kGXF

GXF resource.

## Public Functions

Resource() = default

Resource(Resource&&) = default

```
template<typename ArgT, typename ...ArgsT, typename =  
std::enable_if_t<!std::is_base_of_v<::holoscan::Resource, 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 explicit Resource(ArgT &&arg, ArgsT&&... args)
```

Construct a new Resource object.

### Parameters

**args** – The arguments to be passed to the resource.

~Resource() override = default

```
inline ResourceType resource_type() const
```

Get the resource type.

### Returns

The resource type.

```
inline Resource &name(const std::string &name) &
```

Set the name of the resource.

### Parameters

**name** – The name of the resource.

### Returns

The reference to the resource.

inline [Resource](#) &&name(const std::string &name) &&

Set the name of the resource.

Parameters

**name** – The name of the resource.

Returns

The reference to the resource.

inline [Resource](#) &fragment([Fragment](#) \*fragment)

Set the fragment of the resource.

Parameters

**fragment** – The pointer to the fragment of the resource.

Returns

The reference to the resource.

inline [Resource](#) &spec(const std::shared\_ptr<[ComponentSpec](#)> &spec)

Set the component specification to the resource.

Parameters

**spec** – The component specification.

Returns

The reference to the resource.

inline [ComponentSpec](#) \*spec()

Get the component specification of the resource.

Returns

The pointer to the component specification.

inline std::shared\_ptr<[ComponentSpec](#)> spec\_shared()

Get the shared pointer to the component spec.

Returns

The shared pointer to the component spec.

inline virtual void setup([ComponentSpec](#) &spec)

Define the resource specification.

Parameters

**spec** – The reference to the component specification.

virtual void initialize() override

Initialize the component.

This method is called only once when the component is created for the first time, and use of light-weight initialization.

virtual YAML::Node to\_yaml\_node() const override

Get a YAML representation of the resource.

Returns

YAML node including spec of the resource in addition to the base component properties.

Protected Functions

void update\_params\_from\_args()

Update parameters based on the specified arguments.

virtual void set\_parameters()

Set the parameters based on defaults (sets GXF parameters for GXF components)

```
void update_params_from_args(std::unordered_map<std::string, ParameterWrapper>
&params)
```

Update parameters based on the specified arguments.

Protected Attributes

```
ResourceType resource_type_ = ResourceType::kNative
```

The type of the resource.

```
bool is_initialized_ = false
```

Whether the resource is initialized.

Friends

```
friend class holoscan::NetworkContext
```

```
friend class holoscan::Scheduler
```

```
friend class holoscan::Operator
```

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