# Namespace Cobilas.GodotEngine.Component

## Classes

InternalComponentHub
: Inner class for handling IComponentHub.

NullComponentHub
: Represents a null ComponentHub.

RequireComponentAttribute
: Signals to the AddRequireComponent(Node?) method which components to add to the Godot.Node object.

## Interfaces

IComponentHub
: An interface to transform a Godot.Node object into a pseudo Component.

IInternalComponentHub
: Interface for inner class for handling IComponentHub.

# Interface IComponentHub

Namespace: Cobilas.GodotEngine.Component

Assembly: com.cobilas.godot.icomponent.dll

An interface to transform a Godot.Node object into a pseudo Component.

```
public interface IComponentHub : IEnumerable<Node>, IEnumerable
```

**Inherited Members**
IEnumerable<Node>.GetEnumerator()

## Properties

### ComponentsCount

The number of child objects.

```
int ComponentsCount { get; }
```

#### Property Value

int

Returns the number of child objects.

### Parent

The parent object.

```
Node? Parent { get; }
```

#### Property Value

Node

Returns the parent object.

## ParentComponent

The parent object as IComponentHub.

```
IComponentHub? ParentComponent { get; }
```

### Property Value

IComponentHub

Returns the parent object as IComponentHub.

# Methods

## AddComponent(Type?)

Allows you to add a component by specifying its type.

```
Node? AddComponent(Type? component)
```

### Parameters

`component` Type

The type to be added.

### Returns

Node

Returns the type that was added.

## AddComponent<TypeComponent>()

Allows you to add a component by specifying its type.

```
TypeComponent? AddComponent<TypeComponent>() where TypeComponent : Node
```

### Returns

TypeComponent

Returns the type that was added.

### Type Parameters

`TypeComponent`

The type to be added.

## AddComponents(params Type[]?)

Allows you to add multiple components by specifying their type.

```
void AddComponents(params Type[]? components)
```

### Parameters

`components` Type[]

The types to be added.

## AddNodeComponent(Node?)

Allows you to add a Godot.Node object to the component list.

```
void AddNodeComponent(Node? component)
```

### Parameters

`component` Node

The Godot.Node object to add.

## AddNodeComponents(params Node[]?)

Allows you to add multiple Godot.Node objects to the component list.

```
void AddNodeComponents(params Node[]? components)
```

### Parameters

`components` Node[]

The Godot.Node objects to add.

## GetComponent(Type?)

Gets the component by the specified type.

```
Node? GetComponent(Type? component)
```

### Parameters

`component` Type

The type to be obtained.

### Returns

Node

Returns the component type as node.

## GetComponent(Type?, bool)

Gets the component by the specified type.

```
Node? GetComponent(Type? component, bool recursive)
```

### Parameters

`component` Type

The type to be obtained.

`recursive` bool

Allows searching in sub-children.

### Returns

Node

Returns the component type as node.

## GetComponent<TypeComponent>()

Gets the component by the specified type.

```
TypeComponent? GetComponent<TypeComponent>() where TypeComponent : Node
```

### Returns

TypeComponent

Returns the component type as node.

### Type Parameters

`TypeComponent`

The type to be obtained.

## GetComponent<TypeComponent>(bool)

Gets the component by the specified type.

```
TypeComponent? GetComponent<TypeComponent>(bool recursive) where TypeComponent : Node
```

### Parameters

`recursive` bool

Allows searching in sub-children.

### Returns

TypeComponent

Returns the component type as node.

### Type Parameters

`TypeComponent`

The type to be obtained.

## GetComponents(Type?)

Gets components by the specified type.

```
Node[]? GetComponents(Type? component)
```

### Parameters

`component` Type

The type to be obtained.

### Returns

Node[]

Returns the component types as a node list.

## GetComponents(Type?, bool)

Gets components by the specified type.

```
Node[]? GetComponents(Type? component, bool recursive)
```

### Parameters

`component` Type

The type to be obtained.

`recursive` bool

Allows searching in sub-children.

### Returns

Node[]

Returns the component types as a node list.

## GetComponents<TypeComponent>()

Gets components by the specified type.

```
TypeComponent[]? GetComponents<TypeComponent>() where TypeComponent : Node
```

### Returns

TypeComponent[]

Returns the component types as a node list.

### Type Parameters

`TypeComponent`

The type to be obtained.

## GetComponents<TypeComponent>(bool)

Gets components by the specified type.

```
TypeComponent[]? GetComponents<TypeComponent>(bool recursive) where TypeComponent : Node
```

### Parameters

`recursive` bool

Allows searching in sub-children.

### Returns

TypeComponent[]

Returns the component types as a node list.

### Type Parameters

`TypeComponent`

The type to be obtained.

## RemoveComponent(Node?)

Allows you to remove a Godot.Node object from the list of components.

```
bool RemoveComponent(Node? component)
```

### Parameters

`component` Node

The Godot.Node object to remove.

### Returns

bool

Returns `true` if the operation is successful.

## RemoveComponents(params Node[]?)

Allows you to remove several Godot.Node objects from the list of components.

```
void RemoveComponents(params Node[]? components)
```

## Parameters

`components` Node[]

The Godot.Node objects to be removed.

# Interface IInternalComponentHub

Namespace: Cobilas.GodotEngine.Component

Assembly: com.cobilas.godot.icomponent.dll

Interface for inner class for handling IComponentHub.

```
public interface IInternalComponentHub : IComponentHub, IEnumerable<Node>, IEnumerable
```

**Inherited Members**
IComponentHub.Parent , IComponentHub.ComponentsCount , IComponentHub.ParentComponent , IComponentHub.GetComponent(Type, bool) , IComponentHub.GetComponent(Type) , IComponentHub.GetComponent<TypeComponent>(bool) , IComponentHub.GetComponent<TypeComponent>() , IComponentHub.GetComponents(Type, bool) , IComponentHub.GetComponents(Type) , IComponentHub.GetComponents<TypeComponent>(bool) , IComponentHub.GetComponents<TypeComponent>() , IComponentHub.AddComponent(Type) , IComponentHub.AddComponent<TypeComponent>() , IComponentHub.AddComponents(params Type[]) , IComponentHub.AddNodeComponent(Node) , IComponentHub.AddNodeComponents(params Node[]) , IComponentHub.RemoveComponent(Node) , IComponentHub.RemoveComponents(params Node[]) , IEnumerable<Node>.GetEnumerator()

# Properties

## Entity

The Godot.Node object that is associated.

```
Node? Entity { get; }
```

### Property Value

Node

Returns the associated Godot.Node object.

# Class InternalComponentHub

Namespace: Cobilas.GodotEngine.Component

Assembly: com.cobilas.godot.icomponent.dll

Inner class for handling IComponentHub.

```
[Serializable]
public sealed class InternalComponentHub : IInternalComponentHub, IComponentHub,
IEnumerable<Node>, IEnumerable, IDisposable
```

**Inheritance**

object ← InternalComponentHub

**Implements**

IInternalComponentHub, IComponentHub, IEnumerable<Node>, IEnumerable, IDisposable

**Inherited Members**

object.ToString() , object.Equals(object) , object.Equals(object, object) , object.ReferenceEquals(object, object) , object.GetHashCode() , object.GetType()

## Constructors

### InternalComponentHub(Node)

Inner class for handling IComponentHub.

```
public InternalComponentHub(Node entity)
```

#### Parameters

`entity` Node

## Properties

### ComponentsCount

The number of child objects.

```
public int ComponentsCount { get; }
```

### Property Value

int

Returns the number of child objects.

## Entity

The Godot.Node object that is associated.

```
public Node? Entity { get; }
```

### Property Value

Node

Returns the associated Godot.Node object.

## Parent

The parent object.

```
public Node? Parent { get; }
```

### Property Value

Node

Returns the parent object.

## ParentComponent

The parent object as IComponentHub.

```
public IComponentHub? ParentComponent { get; }
```

### Property Value

IComponentHub

Returns the parent object as IComponentHub.

## Methods

### AddComponent(Type?)

Allows you to add a component by specifying its type.

```
public Node? AddComponent(Type? component)
```

### Parameters

`component` Type

The type to be added.

### Returns

Node

Returns the type that was added.

### Remarks

If the specified type is null or not found in the component list, an object of type NullNode will be returned.

### Exceptions

ArgumentException

Occurs when the specified type does not inherit from Godot.Node.

# AddComponent<TypeComponent>()

Allows you to add a component by specifying its type.

```
public TypeComponent? AddComponent<TypeComponent>() where TypeComponent : Node
```

## Returns

TypeComponent

Returns the type that was added.

## Type Parameters

`TypeComponent`

The type to be added.

## Remarks

If the specified type is null or not found in the component list, an object of type NullNode will be returned.

## Exceptions

ArgumentException

Occurs when the specified type does not inherit from Godot.Node.

# AddComponents(params Type[]?)

Allows you to add multiple components by specifying their type.

```
public void AddComponents(params Type[]? components)
```

## Parameters

`components` Type[]

The types to be added.

### Exceptions

ArgumentException

Occurs when the specified type does not inherit from Godot.Node.

## AddNodeComponent(Node?)

Allows you to add a Godot.Node object to the component list.

```
public void AddNodeComponent(Node? component)
```

### Parameters

`component` Node

The Godot.Node object to add.

## AddNodeComponents(params Node[]?)

Allows you to add multiple Godot.Node objects to the component list.

```
public void AddNodeComponents(params Node[]? components)
```

### Parameters

`components` Node[]

The Godot.Node objects to add.

## AddRequireComponent(Node?)

Static function to add components automatically.

```
public static void AddRequireComponent(Node? mono)
```

### Parameters

`mono` Node

Target Godot.Node object.

### Remarks

The target Godot.Node object must have the RequireComponentAttribute attribute to specify the types to be added.

### Exceptions

ArgumentException

Occurs when the specified type does not inherit from Godot.Node.

## Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

```
public void Dispose()
```

## ~InternalComponentHub()

The destructor is responsible for discarding unmanaged resources.

```
protected ~InternalComponentHub()
```

## GetComponent(Type?)

Gets the component by the specified type.

```
public Node? GetComponent(Type? component)
```

### Parameters

`component` Type

The type to be obtained.

### Returns

Node

Returns the component type as node.

### Remarks

If the specified type is null or not found in the component list, an object of type NullNode will be returned.

### Exceptions

ArgumentException

Occurs when the specified type does not inherit from Godot.Node.

## GetComponent(Type?, bool)

Gets the component by the specified type.

```
public Node? GetComponent(Type? component, bool recursive)
```

### Parameters

`component` Type

The type to be obtained.

`recursive` bool

Allows searching in sub-children.

### Returns

Node

Returns the component type as node.

### Remarks

If the specified type is null or not found in the component list, an object of type NullNode will be returned.

### Exceptions

ArgumentException

Occurs when the specified type does not inherit from Godot.Node.

## GetComponent<TypeComponent>()

Gets the component by the specified type.

```
public TypeComponent? GetComponent<TypeComponent>() where TypeComponent : Node
```

### Returns

TypeComponent

Returns the component type as node.

### Type Parameters

`TypeComponent`

The type to be obtained.

### Remarks

If the specified type is null or not found in the component list, an object of type NullNode will be returned.

### Exceptions

ArgumentException

Occurs when the specified type does not inherit from Godot.Node.

## GetComponent<TypeComponent>(bool)

Gets the component by the specified type.

```
public TypeComponent? GetComponent<TypeComponent>(bool recursive) where TypeComponent : Node
```

## Parameters

`recursive` bool

Allows searching in sub-children.

## Returns

TypeComponent

Returns the component type as node.

## Type Parameters

`TypeComponent`

The type to be obtained.

## Remarks

If the specified type is null or not found in the component list, an object of type NullNode will be returned.

## Exceptions

ArgumentException

Occurs when the specified type does not inherit from Godot.Node.

# GetComponents(Type?)

Gets components by the specified type.

```
public Node[]? GetComponents(Type? component)
```

## Parameters

`component` Type

The type to be obtained.

### Returns

Node[]

Returns the component types as a node list.

### Remarks

If the specified type is null or not found in the component list, an empty list will be returned.

### Exceptions

ArgumentException

Occurs when the specified type does not inherit from Godot.Node.

## GetComponents(Type?, bool)

Gets components by the specified type.

```
public Node[]? GetComponents(Type? component, bool recursive)
```

### Parameters

`component` Type

The type to be obtained.

`recursive` bool

Allows searching in sub-children.

### Returns

Node[]

Returns the component types as a node list.

### Remarks

If the specified type is null or not found in the component list, an empty list will be returned.

### Exceptions

ArgumentException

Occurs when the specified type does not inherit from Godot.Node.

## GetComponents<TypeComponent>()

Gets components by the specified type.

```
public TypeComponent[]? GetComponents<TypeComponent>() where TypeComponent : Node
```

### Returns

TypeComponent[]

Returns the component types as a node list.

### Type Parameters

`TypeComponent`

The type to be obtained.

### Remarks

If the specified type is null or not found in the component list, an empty list will be returned.

### Exceptions

ArgumentException

Occurs when the specified type does not inherit from Godot.Node.

## GetComponents<TypeComponent>(bool)

Gets components by the specified type.

```
public TypeComponent[]? GetComponents<TypeComponent>(bool recursive) where TypeComponent
: Node
```

### Parameters

`recursive` bool

Allows searching in sub-children.

### Returns

TypeComponent[]

Returns the component types as a node list.

### Type Parameters

`TypeComponent`

The type to be obtained.

### Remarks

If the specified type is null or not found in the component list, an empty list will be returned.

### Exceptions

ArgumentException

Occurs when the specified type does not inherit from Godot.Node.

## GetEnumerator()

Returns an enumerator that iterates through the collection.

```
public IEnumerator<Node> GetEnumerator()
```

### Returns

IEnumerator<Node>

An enumerator that can be used to iterate through the collection.

## RemoveComponent(Node?)

Allows you to remove a Godot.Node object from the list of components.

```
public bool RemoveComponent(Node? component)
```

### Parameters

`component` Node

The Godot.Node object to remove.

### Returns

bool

Returns `true` if the operation is successful.

## RemoveComponents(params Node[]?)

Allows you to remove several Godot.Node objects from the list of components.

```
public void RemoveComponents(params Node[]? components)
```

### Parameters

`components` Node[]

The Godot.Node objects to be removed.

# Class NullComponentHub

Namespace: Cobilas.GodotEngine.Component

Assembly: com.cobilas.godot.icomponent.dll

Represents a null ComponentHub.

```
public class NullComponentHub : Node, IDisposable, IComponentHub, IEnumerable<Node>,
IEnumerable, INullObject
```

### Inheritance

object ← Object ← Node ← NullComponentHub

### Implements

IDisposable, IComponentHub, IEnumerable<Node>, IEnumerable, INullObject

### Inherited Members

Node.NotificationEnterTree , Node.NotificationExitTree , Node.NotificationMovedInParent , Node.NotificationReady , Node.NotificationPaused , Node.NotificationUnpaused , Node.NotificationPhysicsProcess , Node.NotificationProcess , Node.NotificationParented , Node.NotificationUnparented , Node.NotificationInstanced , Node.NotificationDragBegin , Node.NotificationDragEnd , Node.NotificationPathChanged , Node.NotificationChildOrderChanged , Node.NotificationInternalProcess , Node.NotificationInternalPhysicsProcess , Node.NotificationPostEnterTree , Node.NotificationResetPhysicsInterpolation , Node.NotificationWmMouseEnter , Node.NotificationWmMouseExit , Node.NotificationWmFocusIn , Node.NotificationWmFocusOut , Node.NotificationWmQuitRequest , Node.NotificationWmGoBackRequest , Node.NotificationWmUnfocusRequest , Node.NotificationOsMemoryWarning , Node.NotificationTranslationChanged , Node.NotificationWmAbout , Node.NotificationCrash , Node.NotificationOsImeUpdate , Node.NotificationAppResumed , Node.NotificationAppPaused , Node.GetNode<T>(NodePath) , Node.GetNodeOrNull<T>(NodePath) , Node.GetChild<T>(int) , Node.GetChildOrNull<T>(int) , Node.GetOwner<T>() , Node.GetOwnerOrNull<T>() , Node.GetParent<T>() , Node.GetParentOrNull<T>() , Node._EnterTree() , Node._ExitTree() , Node._GetConfigurationWarning() , Node._Input(InputEvent) , Node._PhysicsProcess(float) , Node._Process(float) , Node._Ready() , Node._UnhandledInput(InputEvent) , Node._UnhandledKeyInput(InputEventKey) , Node.AddChildBelowNode(Node, Node, bool) , Node.SetName(string) , Node.GetName() , Node.AddChild(Node, bool) , Node.RemoveChild(Node) , Node.GetChildCount() , Node.GetChildren() , Node.GetChild(int) , Node.HasNode(NodePath) , Node.GetNode(NodePath) , Node.GetNodeOrNull(NodePath) , Node.GetParent() , Node.FindNode(string, bool, bool) , Node.FindParent(string) , Node.HasNodeAndResource(NodePath) ,

Node.GetNodeAndResource(NodePath) , Node.IsInsideTree() , Node.IsAParentOf(Node) , Node.IsGreaterThan(Node) , Node.GetPath() , Node.GetPathTo(Node) , Node.AddToGroup(string, bool) , Node.RemoveFromGroup(string) , Node.IsInGroup(string) , Node.MoveChild(Node, int) , Node.GetGroups() , Node.Raise() , Node.SetOwner(Node) , Node.GetOwner() , Node.RemoveAndSkip() , Node.GetIndex() , Node.PrintTree() , Node.PrintTreePretty() , Node.SetFilename(string) , Node.GetFilename() , Node.PropagateNotification(int) , Node.PropagateCall(string, Array, bool) , Node.SetPhysicsProcess(bool) , Node.GetPhysicsProcessDeltaTime() , Node.IsPhysicsProcessing() , Node.GetProcessDeltaTime() , Node.SetProcess(bool) , Node.SetProcessPriority(int) , Node.GetProcessPriority() , Node.IsProcessing() , Node.SetProcessInput(bool) , Node.IsProcessingInput() , Node.SetProcessUnhandledInput(bool) , Node.IsProcessingUnhandledInput() , Node.SetProcessUnhandledKeyInput(bool) , Node.IsProcessingUnhandledKeyInput() , Node.SetPauseMode(Node.PauseModeEnum) , Node.GetPauseMode() , Node.CanProcess() , Node.PrintStrayNodes() , Node.GetPositionInParent() , Node.SetDisplayFolded(bool) , Node.IsDisplayedFolded() , Node.SetProcessInternal(bool) , Node.IsProcessingInternal() , Node.SetPhysicsProcessInternal(bool) , Node.IsPhysicsProcessingInternal() , Node.SetPhysicsInterpolationMode(Node.PhysicsInterpolationModeEnum) , Node.GetPhysicsInterpolationMode() , Node.IsPhysicsInterpolated() , Node.IsPhysicsInterpolatedAndEnabled() , Node.ResetPhysicsInterpolation() , Node.GetTree() , Node.CreateTween() , Node.Duplicate(int) , Node.ReplaceBy(Node, bool) , Node.SetSceneInstanceLoadPlaceholder(bool) , Node.GetSceneInstanceLoadPlaceholder() , Node.SetEditableInstance(Node, bool) , Node.IsEditableInstance(Node) , Node.GetViewport() , Node.QueueFree() , Node.RequestReady() , Node.IsNodeReady() , Node.SetNetworkMaster(int, bool) , Node.GetNetworkMaster() , Node.IsNetworkMaster() , Node.GetMultiplayer() , Node.GetCustomMultiplayer() , Node.SetCustomMultiplayer(MultiplayerAPI) , Node.RpcConfig(string, MultiplayerAPI.RPCMode) , Node.RsetConfig(string, MultiplayerAPI.RPCMode) , Node.SetUniqueNameInOwner(bool) , Node.IsUniqueNameInOwner() , Node.Rpc(string, params object[]) , Node.RpcUnreliable(string, params object[]) , Node.RpcId(int, string, params object[]) , Node.RpcUnreliableId(int, string, params object[]) , Node.Rset(string, object) , Node.RsetId(int, string, object) , Node.RsetUnreliable(string, object) , Node.RsetUnreliableId(int, string, object) , Node.UpdateConfigurationWarning() , Node.EditorDescription , Node._ImportPath , Node.PauseMode , Node.PhysicsInterpolationMode , Node.Name , Node.UniqueNameInOwner , Node.Filename , Node.Owner , Node.Multiplayer , Node.CustomMultiplayer , Node.ProcessPriority , Object.NotificationPostinitialize , Object.NotificationPredelete , Object.IsInstanceValid(Object) , Object.WeakRef(Object) , Object.Dispose() , Object.Dispose(bool) , Object.ToString() , Object.ToSignal(Object, string) , Object._Get(string) , Object._GetPropertyList() , Object._Notification(int) , Object._Set(string, object) , Object.Free() , Object.GetClass() , Object.IsClass(string) , Object.Set(string, object) , Object.Get(string) , Object.SetIndexed(NodePath, object) , Object.GetIndexed(NodePath) , Object.GetPropertyList() ,

Object.GetMethodList() , Object.Notification(int, bool) , Object.GetInstanceId() , Object.SetScript(Reference) , Object.GetScript() , Object.SetMeta(string, object) , Object.RemoveMeta(string) , Object.GetMeta(string, object) , Object.HasMeta(string) , Object.GetMetaList() , Object.AddUserSignal(string, Array) , Object.HasUserSignal(string) , Object.EmitSignal(string, params object[]) , Object.Call(string, params object[]) , Object.CallDeferred(string, params object[]) , Object.SetDeferred(string, object) , Object.Callv(string, Array) , Object.HasMethod(string) , Object.HasSignal(string) , Object.GetSignalList() , Object.GetSignalConnectionList(string) , Object.GetIncomingConnections() , Object.Connect(string, Object, string, Array, uint) , Object.Disconnect(string, Object, string) , Object.IsConnected(string, Object, string) , Object.SetBlockSignals(bool) , Object.IsBlockingSignals() , Object.PropertyListChangedNotify() , Object.SetMessageTranslation(bool) , Object.CanTranslateMessages() , Object.Tr(string) , Object.IsQueuedForDeletion() , Object.NativeInstance , Object.DynamicObject , object.Equals(object) , object.Equals(object, object) , object.ReferenceEquals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone()

# Properties

## ComponentsCount

The number of child objects.

```
public int ComponentsCount { get; }
```

### Property Value

int

Returns the number of child objects.

## Null

A representation of a null ComponentHub.

```
public static NullComponentHub Null { get; }
```

### Property Value

NullComponentHub

Returns a representation of a null ComponentHub.

## Parent

The parent object.

```
public Node? Parent { get; }
```

### Property Value

Node

Returns the parent object.

## ParentComponent

The parent object as IComponentHub.

```
public IComponentHub? ParentComponent { get; }
```

### Property Value

IComponentHub

Returns the parent object as IComponentHub.

# Methods

## AddComponent(Type?)

Allows you to add a component by specifying its type.

```
public Node? AddComponent(Type? component)
```

### Parameters

`component` Type

The type to be added.

### Returns

Node

Returns the type that was added.

## AddComponent<T>()

Allows you to add a component by specifying its type.

```
public T? AddComponent<T>() where T : Node
```

### Returns

T

Returns the type that was added.

### Type Parameters

`T`

## AddComponents(params Type[]?)

Allows you to add multiple components by specifying their type.

```
public void AddComponents(params Type[]? components)
```

### Parameters

`components` Type[]

The types to be added.

# AddNodeComponent(Node?)

Allows you to add a Godot.Node object to the component list.

```
public void AddNodeComponent(Node? component)
```

## Parameters

`component` Node

The Godot.Node object to add.

# AddNodeComponents(params Node[]?)

Allows you to add multiple Godot.Node objects to the component list.

```
public void AddNodeComponents(params Node[]? components)
```

## Parameters

`components` Node[]

The Godot.Node objects to add.

# GetComponent(Type?)

Gets the component by the specified type.

```
public Node? GetComponent(Type? component)
```

## Parameters

`component` Type

The type to be obtained.

## Returns

Node

Returns the component type as node.

## GetComponent(Type?, bool)

Gets the component by the specified type.

```
public Node? GetComponent(Type? component, bool recursive)
```

### Parameters

`component` Type

The type to be obtained.

`recursive` bool

Allows searching in sub-children.

### Returns

Node

Returns the component type as node.

## GetComponent<T>()

Gets the component by the specified type.

```
public T? GetComponent<T>() where T : Node
```

### Returns

T

Returns the component type as node.

### Type Parameters

T

## GetComponent<T>(bool)

Gets the component by the specified type.

```
public T? GetComponent<T>(bool recursive) where T : Node
```

### Parameters

`recursive` bool

Allows searching in sub-children.

### Returns

T

Returns the component type as node.

### Type Parameters

T

## GetComponents(Type?)

Gets components by the specified type.

```
public Node[]? GetComponents(Type? component)
```

### Parameters

`component` Type

The type to be obtained.

### Returns

Node[]

Returns the component types as a node list.

## GetComponents(Type?, bool)

Gets components by the specified type.

```
public Node[]? GetComponents(Type? component, bool recursive)
```

### Parameters

`component` Type

The type to be obtained.

`recursive` bool

Allows searching in sub-children.

### Returns

Node[]

Returns the component types as a node list.

## GetComponents<T>()

Gets components by the specified type.

```
public T[]? GetComponents<T>() where T : Node
```

### Returns

T[]

Returns the component types as a node list.

### Type Parameters

`T`

## GetComponents<T>(bool)

Gets components by the specified type.

```
public T[]? GetComponents<T>(bool recursive) where T : Node
```

### Parameters

`recursive` bool

Allows searching in sub-children.

### Returns

T[]

Returns the component types as a node list.

### Type Parameters

`T`

## GetEnumerator()

Returns an enumerator that iterates through the collection.

```
public IEnumerator<Node> GetEnumerator()
```

### Returns

IEnumerator<Node>

An enumerator that can be used to iterate through the collection.

## RemoveComponent(Node?)

Allows you to remove a Godot.Node object from the list of components.

```
public bool RemoveComponent(Node? component)
```

### Parameters

`component` Node

The Godot.Node object to remove.

### Returns

bool

Returns `true` if the operation is successful.

## RemoveComponents(params Node[]?)

Allows you to remove several Godot.Node objects from the list of components.

```
public void RemoveComponents(params Node[]? components)
```

### Parameters

`components` Node[]

The Godot.Node objects to be removed.

# Class RequireComponentAttribute

Namespace: Cobilas.GodotEngine.Component

Assembly: com.cobilas.godot.icomponent.dll

Signals to the AddRequireComponent(Node?) method which components to add to the Godot.Node object.

```
[AttributeUsage(AttributeTargets.Class, Inherited = true, AllowMultiple = false)]
public sealed class RequireComponentAttribute : Attribute, _Attribute
```

### Inheritance

object ← Attribute ← RequireComponentAttribute

### Implements

_Attribute

### Inherited Members

Attribute.GetCustomAttributes(MemberInfo, Type) ,
Attribute.GetCustomAttributes(MemberInfo, Type, bool) ,
Attribute.GetCustomAttributes(MemberInfo) , Attribute.GetCustomAttributes(MemberInfo, bool) ,
Attribute.IsDefined(MemberInfo, Type) , Attribute.IsDefined(MemberInfo, Type, bool) ,
Attribute.GetCustomAttribute(MemberInfo, Type) ,
Attribute.GetCustomAttribute(MemberInfo, Type, bool) ,
Attribute.GetCustomAttributes(ParameterInfo) , Attribute.GetCustomAttributes(ParameterInfo, Type) ,
Attribute.GetCustomAttributes(ParameterInfo, Type, bool) ,
Attribute.GetCustomAttributes(ParameterInfo, bool) , Attribute.IsDefined(ParameterInfo, Type) ,
Attribute.IsDefined(ParameterInfo, Type, bool) , Attribute.GetCustomAttribute(ParameterInfo, Type) ,
Attribute.GetCustomAttribute(ParameterInfo, Type, bool) ,
Attribute.GetCustomAttributes(Module, Type) , Attribute.GetCustomAttributes(Module) ,
Attribute.GetCustomAttributes(Module, bool) , Attribute.GetCustomAttributes(Module, Type, bool) ,
Attribute.IsDefined(Module, Type) , Attribute.IsDefined(Module, Type, bool) ,
Attribute.GetCustomAttribute(Module, Type) , Attribute.GetCustomAttribute(Module, Type, bool) ,
Attribute.GetCustomAttributes(Assembly, Type) ,
Attribute.GetCustomAttributes(Assembly, Type, bool) , Attribute.GetCustomAttributes(Assembly) ,
Attribute.GetCustomAttributes(Assembly, bool) , Attribute.IsDefined(Assembly, Type) ,
Attribute.IsDefined(Assembly, Type, bool) , Attribute.GetCustomAttribute(Assembly, Type) ,
Attribute.GetCustomAttribute(Assembly, Type, bool) , Attribute.Equals(object) ,
Attribute.GetHashCode() , Attribute.Match(object) , Attribute.IsDefaultAttribute() ,

Attribute.TypeId , object.ToString() , object.Equals(object, object) , object.ReferenceEquals(object, object) , object.GetType()

# Constructors

## RequireComponentAttribute(Type)

Creates a new instance of this object.

```
public RequireComponentAttribute(Type component)
```

### Parameters

`component` Type

## RequireComponentAttribute(Type, Type)

Creates a new instance of this object.

```
public RequireComponentAttribute(Type component1, Type component2)
```

### Parameters

`component1` Type

`component2` Type

## RequireComponentAttribute(Type, Type, Type)

Creates a new instance of this object.

```
public RequireComponentAttribute(Type component1, Type component2, Type component3)
```

### Parameters

`component1` Type

`component2` Type

`component3` Type

## RequireComponentAttribute(params Type[])

Signals to the AddRequireComponent(Node?) method which components to add to the Godot.Node object.

```
public RequireComponentAttribute(params Type[] components)
```

### Parameters

`components` Type[]

# Properties

## Components

The types of components to be added.

```
public Type[] Components { get; }
```

### Property Value

Type[]

Returns the types of components to be added.