UIRecycleTree

class in UIRecycleTreeNamespace

Description

component UIRecycleTree

Properties

NodeCollection **nodes** {get;}	Get NodeCollection of tree root node
Node **rootNode** {get;}	Root node of UIRecycleTree
int **nodesCount** {get;}	Get Nodes count in tree (recursive)
string **separator** {get; set;}	The symbol separating nodes when executing the fullPath property in Node
NodeStyle[] **nodeStyles** {get;}	Get array of NodeStyles
Node **selectedNode** {get;}	Get currently selected of selected Node or return null
bool **hasSelected** {get;}	Does the node currently have a selected Node?
bool **isRecursiveChecked** {get; set;}	When you click the Node checkBox. Change the isChecked property of all child nodes of the clicked node.(recursive)

Public Methods

void **ExpandAll** ()	Expand all child nodes of root node
void **CollapseAll** ()	Collapse all child nodes of root node
void **ExpandAllWithoutNotify** ()	Expand all child nodes without tree Rebuild
void **Clear**()	Remove All Nodes in tree
Node **FindNodeByIdRecursive**(int id)	
Node[] **FindNodesByNameRecursive**(string searchName)	Returns an array of nodes with the name specified in the parameter (Recursive)

Node FindFirstNodeByDataRecursive (object searchedData)	Returns first Node with the data specified in the parameter (Recursive)
void FocusOn(Node **node**)	Focus on node in tree view
void FocusOnSelected()	Focus on node with selected state in tree view
Node FindNodeByNameRecursive (string searchedName)	Search for a node by name among its child nodes

Node

Class in UIRecycleTreeNamespace

Description

Describe the behavior of a node in a tree

Properties

bool hasChildren {get;}	Does the current node have children
int childCount {get;}	
int depth {get;}	Depth in tree hierarchy
bool isExpanded {get; set;}	
bool isSelected {get; set;}	
bool isChecked {get; set;}	
bool isFaded {get; set;}	
string name {get; set;}	Node name
int styleIndex	Index in NodeStylesArray in UIRecycleTree
object data {get; set;}	You can assign any data to node
NodeCollection nodes {get;}	Collection of node children
Int nodeId {get;}	ID unique for each node
UIRecycleTree tree {get; set;}	Node parent tree

Node parentNode {get;}	
string fullPath {get;}	

Constructors

Node()	Creates a new Node
Node(string name**)**	Creates a new Node with name
Node(UIRecycleTree treeView)	Creates a new Node with assigned tree
Node(UIRecycleTree treeView, **Node**[] children)	Creates a new Node with assigned tree, and node children array

Public Methods

void ExpandAll ()	Expand all child nodes
void CollapseAll ()	Collapse all child nodes
void ExpandAllWithoutNotify ()	Expand all child nodes without tree Rebuild
void CollapseAllWithoutNotify ()	Collapse all child nodes without tree Rebuild
Node[] GetAllChildrenRecursive ()	Get all child node array include sender
bool TryCastData<T> (**out T** castedData)	An attempt to cast a node's data field into the specified type
void RemoveYourself ()	
Node FindNodeByIdRecursive (**int** id, **Node** item)	Search for a node by ID among its child nodes
Node FindNodeByDataRecursive (**object** searchedData)	Search for a node by data among its child nodes
Node FindNodeByNameRecursive (**string** searchedName)	Search for a node by name among its child nodes

void FindAllChildrenWithIsCheckedStateRecursive (**List<Node>** foundedItems)	It is necessary to pass a new List<Node> as a parameter
int GetAllChildrenCountRecursive()	
void ChangeIsCheckedStateForAllChildren(**bool** isCheck)	
void SetExpandedStateWithoutNotify()	Set isExpanded state without tree Rebuild
void SetCheckedWithoutNotify(**bool** nodeIsChecked)	Set isChecked state without tree Rebuild
Node[] GetAllParentsRecursive()	Get all node parents

NodeCollection

Class in UIRecycleTreeNamespace

Description

Describe the storage of adding and deleting child elements in a node

Properties

bool count {get;}	Get NodeCollection child count

Constructors

NodeCollection(Node ownerNode**)**	Creates a new NodeCollection. You must specify owner **Node** in param

Public Methods

void Add(Node node**)**	Add node to collection
Node AddFluent(Node node**)**	Add node to collection and return added node
Node AddFluent(string name**)**	Create new Node with the name specified in the parameter Add node to collection And return added node

void AddRange(Node[] nodeArray**)**	
bool Remove(Node node**)**	Remove node from NodeCollection
void RemoveAt(int index**)**	Remove node from NodeCollection by index
int IndexOf(Node node**)**	Return node index from collection