

```

// ****
md.directive('comboTree', function () {
    return {
        restrict: 'EA',
        templateUrl: '/partials/common/comboTree.html',
        transclude: true,
        replace: true,
        scope: {
            treeData: '=', // ****
            treeModel: '=', // ****w****Value
            treeName: '@', // ****name
            treeDisabled: '=', // ****y****
            displayTextNull: '@displayWhenNull', // ****w****'****l****
            displayText: '=treeDisplay', // ****z'****
            treeWidth: '@', // ****-****
            treeHeight: '@', // ****-****
            treeMargin: '@', // ****\****
            expandFirst: '=' // ****I****h****
        },
        require: '?uiTree',
        link: function (scope, element, attributes) {
            // ****
            var dataList = []; // ****δ****n****(****
            var dataListCopy = []; // ****j****s****.
            var count = 0;
            var initParentID = '00000000-0000-0000-0000-000000000000';
            function LoadTreeData(treeNodesData, parentId, flag) {
                var data = [];
                if (treeNodesData)
                    angular.forEach(treeNodesData, function (item) {
                        if (item.ParentID == parentId) {
                            var t = {
                                'Id': item.Id,
                                'Name': item.Name,
                                'ParentID': item.ParentID,
                                'isExpander': flag ? true : false, // ****z****
                                'persons': item.persons, // ****U
                                'nodes': LoadTreeData(treeNodesData, item.Id,
flag)
                            };
                            dataList.push(t);
                            this.push(t);
                        }
                    }, data);
                return data;
            };
            scope.$watch('treeData', function (v) {
                if (v) {
                    dataList = [];
                    // ****j****parentid****Ç****ğ****code****
                    if (scope.treeData && scope.treeData.length > 0) {
                        initParentID = scope.treeData[0].ParentID;
                    }
                }
            });
        }
    };
});

```

```

    }
    scope.source = LoadTreeData(scope.treeData, initParentID);
    if (count == 0 && scope.treeData.length > 0) {
        // 00000000' 000 000000004
        dataListCopy = JSON.parse(JSON.stringify(dataList));
        count++;
    }
}
}, true)

/**
 * 00wStart
 */

// 000000J0K00J0
function GetParentNode(node) {
    var parent = null;
    var currentNode = null;
    var flag = false;
    for (var v in dataList) {
        for (var j in dataList[v].persons) {
            if (dataList[v].persons[j].Id == node.Id) {
                currentNode = dataList[v]; //w0000
                flag = true;
                break;
            }
        }
    }
    if (flag) {
        break;
    }
}
if (currentNode != null) {
    for (var k in dataList) {
        if (dataList[k].Id == currentNode.ParentID) {
            parent = dataList[v];
            break;
        }
    }
}

return parent;
}

#00 J0000000J0
function GetParentNodeByNode(node) {
    var parent = null;
    for (var k in dataList) {
        if (dataList[k].Id == node.ParentID) {
            parent = dataList[k];
            break;
        }
    }
    return parent;
}

```

```

    }
    // 22222222
    function ExpendParent(node, type) {
        var parent = null;
        if (type == "person") {
            parent = GetParentNode(node, scope.source);
        } else {
            parent = GetParentNodeByNode(node, scope.source);
        }

        if (parent != null) {
            parent.isExpander = true;
            ExpendParent(parent);
        }
    }

    // 22' 22222222w 2222w 2222
    function setSingleSelectedNode(selectedValue) {
        // 22 22 22222222
        var flag = false;
        for (var d in dataList) {
            var node = dataList[d];
            for (var j in node.persons) {
                if (node.persons[j].Id == selectedValue) {
                    scope.selectedNode = node.persons[j]; //w 2222
                    flag = true;
                    break;
                }
            }
        }
        // 22 22 22222222
        if (flag) {
            break;
        }
    }

    // 22222222 22 21
    scope.$watch('selectedNode', function (newValue) {
        if (newValue) {
            scope.treeModel = newValue.Id;
            scope.displayText = newValue.Name;
        } else {
            scope.displayText = "";
        }
    }, true)
    /**
     * 22wEnd
     */

    // 22222222222222 22 2222
    scope.$watch('displayText', function (newValue) {
        if (newValue) {

```

```

var currentData = [];
var flag = false;
//????.?
dataList = JSON.parse(JSON.stringify(dataListCopy));
for (var d in dataList) {

    flag = false;
    var node = dataList[d];
    var currentPersons = [];
    //???
    for (var j in node.persons) {
        if (node.persons[j].Name.indexOf(newValue) != -1) {
            currentPersons.push(node.persons[j]);
            flag = true;
        }
    }
    if (flag) {
        node.persons = currentPersons;
        currentData.push(node);
    }
}
//?X
for (var i in currentData) {
    var parent = GetParentNodeByNode(currentData[i]);
    while (parent != null) {
        var isContain = false;
        for (var m in currentData) {
            if (currentData[m].Id == parent.Id) {
                isContain = true;
                break;
            }
        }
        if (!isContain) {
            parent.persons = [];
            currentData.push(parent);
        }
        parent = GetParentNodeByNode(parent);
    }
}

dataList = [];
scope.source = LoadTreeData(currentData, initParentID,
true);

} else {
    scope.treeModel = null;
    scope.selectedNode = null;
    dataList = [];
    scope.source = LoadTreeData(scope.treeData, initParentID);
}

})

//?n?δ??-??â??w?ж?Ö
function initSelectedNode() {

```

```

        if (scope.treeModel) {
            scope.selectedNode = null;
            setSingleSelectedNode(scope.treeModel);
            // 初始化
            dataList = [];
            scope.source = LoadTreeData(scope.treeData, initParentID);
            if (scope.selectedNode) {
                for (var v in dataList) {
                    dataList[v].isExpander = false;
                }
                ExpendParent(scope.selectedNode, "person");
            } else
                scope.selectedNode = null;
        }
    }
    scope.$watch('source', function (v) {
        if (v) {
            // 初始化
            if (scope.expandFirst == true && scope.source) {
                angular.forEach(scope.source, function (v, k) {
                    v.isExpander = true;
                });
            }
        }
    });
})

// 初始化treeModel
scope.$watch('treeModel', function (newValue, oldValue) {
    if (!(oldValue == undefined && newValue == undefined)) {
        scope.selectedNode = null;
        setSingleSelectedNode(scope.treeModel);
        if (scope.selectedNode) {
            ExpendParent(scope.selectedNode, "person");
        } else
            scope.selectedNode = null;
    }
})

// 初始化
scope.selectNode = function (item, node) {
    scope.selectedNode = item;
    scope.showTree = false;
}

// 初始化
scope.expandTree = function () {
    if (!scope.showTree) {
        initSelectedNode();
    }
    scope.showTree = !scope.showTree;
};

```

```

// 000+00^n 00w00
scope.expandItem = function (item, $event) {
    $event.stopPropagation(); // 000¼0000
    item.isExpander = !item.isExpander;
}

// 00000000000-0000' 01 0000000000 0^n 0 000-00
$(document).mousedown(function (e) {
    var event = window.event || e;
    var target = event.target || event.srcElement;
    var comboTrees = document.getElementsByName('combo_tree');
    var flag = true;
    do {
        if (target == document) {
            flag = false;
            break;
        }
        for (var i = 0; i < comboTrees.length; i++) {
            if (target == comboTrees[i]) {
                flag = false;
                break;
            }
        }
        if (flag)
            target = target.parentNode;
    } while (flag && target != null);

    if (target == document) {
        scope.$apply(function () {
            scope.showTree = false;
        })
    }
})
}
});

```

```

<combo-tree tree-name="build_Region4" nametext="00w0000000" tree-
data="points" tree-model="patrolPlan.patrol_pointId" display-when-null=""
tree-display="patrolPlanName1" tree-width="483" tree-
height="400" expand-first="true"></combo-tree>

```