Template:AjaxVoter

```
<input type="hidden" id=('TheProperty_' ..@id) value=($0 ?? $name ?? 'vot
<input type="hidden" id=('pageApi_' ..@id) value=($1 ?? $pageid ?? wiki.g
<input type="hidden" id=('CurUserID_' ..@id) value=(user.id) />
<input type="hidden" id=('UniqueId_' ..@id) value=($2 ?? $voterid ?? '')

var scoretype = ($3 ?? $scoring ?? 'sum');
if(scoretype != 'sum' && scoretype != 'percent') {let scoretype = 'sum'}
<input type="hidden" id=('Scoring_' ..@id) value=(scoretype) />

if(($3 ?? $voterid ?? '') == '') {
    <p>
        'Template:AjaxVoter (ERROR):  The $voterid Parameter MUST have a
    </p>
} else {

<table>
<tr>
<td style="padding:0px; border:1px solid black; min-width:75px; width:75p
<div id=('voter-container_' ..@id) style="padding-bottom:5px">
    <div id=('vote-results-container_' ..@id)>
        <div id=('vote-results-percent_' ..@id) style="font-size:.8em;for
        <div id=('vote-results_' ..@id) style="font-size:1.5em;font-weigh
        <div id=('vote-loading_' ..@id) style="margin:auto;width:16px;hei
    </div>
    <div id=('results-footer_' ..@id) style="font-weight:bold;font-size:.
    <div id=('icon-wrap_' ..@id) style="text-align:center;">
        if(!user.anonymous) {
            <a href="#" id=('down_val_' ..@id)><img src="/@api/deki/files
            <a href="#" id=('up_val_' ..@id)><img src="/@api/deki/files/2
        }
        else 'Login to Vote';
    </div>
</div>
</td>
</tr>
</table>
}
<html><head>

//    <link rel="stylesheet" type="text/css" href="" />
//    <script type="text/javascript" src="" />
```

```
    <script type="text/javascript">"

        $(document).ready(function() {
var scoring = $('#Scoring_"..@id.."').val();
if (scoring == 'sum') $('#vote-results-container_"..@id.."').hover(ShowPe
$('#vote-results-percent_"..@id.."').hide();

$('#up_val_"..@id.."').click(function() {
    var thePageApi = $('#pageApi_"..@id.."').val();
    var thePropertyName = $('#TheProperty_"..@id.."').val();

    ReadProperty( thePageApi, thePropertyName, IncreaseVote_"..@id..");
    return false;
});
$('#down_val_"..@id.."').click(function() {
    var thePageApi = $('#pageApi_"..@id.."').val();
    var thePropertyName = $('#TheProperty_"..@id.."').val();

    ReadProperty( thePageApi, thePropertyName, DecreaseVote_"..@id..");
    return false;
});

GetVoteCount_"..@id.."();

        });

function ShowPercent"..@id.."() {
    $('#vote-results_"..@id.."').hide();
    $('#vote-results-percent_"..@id.."').show();
}
function ShowScore"..@id.."() {
    $('#vote-results-percent_"..@id.."').hide();
    $('#vote-results_"..@id.."').show();
}

var SaveTotals_"..@id.." = function(curVal, status) {

    var voteCounts = '';
    var totalsPos = 0;
    var totalsNeg = 0;
    var thePageApi = $('#pageApi_"..@id.."').val();
    var thePropertyName = 'Totals@' + $('#UniqueId_"..@id.."').val();

    if(typeOf(curVal) == 'array') {
```

```
        for(var i = 0; i < curVal.length; i++) {
            if(curVal[i].vote == 1) {
                totalsPos = totalsPos + 1;
            } else {
                totalsNeg = totalsNeg + -1;
            }
        }
        voteCounts = {positive: totalsPos, negative: Math.abs(totalsNeg),
    } else {
        voteCounts = {positive: 0, negative: 0, score: 0};
    }

    AddUpdateProperty(thePageApi, thePropertyName, voteCounts, '', '');
}


var IncreaseVote_"..@id.." = function(curVal, status) {
    $('#icon-wrap_"..@id.."').hide();
    $('#vote-loading_"..@id.."').show();
    $('#vote-results_"..@id.."').hide();

    var voteCounts = new Array;
    var newUser = true;
    var thePageApi = $('#pageApi_"..@id.."').val();
    var thePropertyName = $('#TheProperty_"..@id.."').val();
    var CurUserID = $('#CurUserID_"..@id.."').val();

    if(typeOf(curVal) == 'array') {
        for(var i = 0; i < curVal.length; i++) {
            if(curVal[i].userid == CurUserID) {
                curVal[i].vote = 1;
                newUser = false;
                break;
            }
        }
    } else {
        curVal = new Array;
    }

    if(newUser == true) {
        curVal[curVal.length] = {userid: CurUserID, vote: 1};
    }

    AddUpdateProperty(thePageApi, thePropertyName, curVal, GetVoteCount_'
}
```

```
var DecreaseVote_"..@id.." = function(curVal, status) {
    $('#icon-wrap_"..@id.."').hide();
    $('#vote-loading_"..@id.."').show();
    $('#vote-results_"..@id.."').hide();

    var voteCounts = new Array;
    var newUser = true;
    var thePageApi = $('#pageApi_"..@id.."').val();
    var thePropertyName = $('#TheProperty_"..@id.."').val();
    var CurUserID = $('#CurUserID_"..@id.."').val();

    if(typeOf(curVal) == 'array') {
        for(var i = 0; i < curVal.length; i++) {
            if(curVal[i].userid == CurUserID) {
                curVal[i].vote = -1;
                newUser = false;
                break;
            }
        }
    } else {
        curVal = new Array;
    }

    if(newUser == true) {
        curVal[curVal.length] = {userid: CurUserID, vote: -1};
    }

    AddUpdateProperty(thePageApi, thePropertyName, curVal, GetVoteCount_'
}

var GetVoteCount_"..@id.." = function(parm, status) {
    $('#vote-loading_"..@id.."').show();
    $('#vote-results_"..@id.."').hide();

    var thePageApi = $('#pageApi_"..@id.."').val();
    var thePropertyName = $('#TheProperty_"..@id.."').val();
    var CurUserID = $('#CurUserID_"..@id.."').val();

    ReadProperty( thePageApi, thePropertyName, ShowVoteCount_"..@id..");
}

var ShowVoteCount_"..@id.." = function(curVal, status) {
    $('#icon-wrap_"..@id.."').show();
    $('#vote-results_"..@id.."').show();
```

```
    $('#vote-loading_"..@id.."').hide();

// calculate vote totals
    var totalsPos = 0;
    var runTotal = 0;
    var totalVotes = 0;
    var bgcolor;
    var green = new Array(80,225,80);
    var red = new Array(245,110,100);
    var white = new Array(255,255,255);
    var scoring = $('#Scoring_"..@id.."').val();
    if(typeOf(curVal) == 'array') {
        for(var i = 0; i < curVal.length; i++) {
            runTotal = runTotal + curVal[i].vote
            if(curVal[i].vote > 0) totalsPos ++;
        }
        totalVotes = curVal.length;
    }
// calculate and output final score and color
    if (scoring == 'sum') {
        if (runTotal == 0)
            bgcolor = 'rgb(' + white.join(',') + ')';
        else if (runTotal > 0) {
            runTotal = '+' + runTotal;
            bgcolor = 'rgb(' + green.join(',') + ')';
        }
        else if (runTotal < 0)
            bgcolor = 'rgb(' + red.join(',') + ')';
        // update percentage overlay
        var percentPos = Math.round((totalsPos / Math.max(totalVotes,1))
        $('#vote-results-percent_"..@id.."').html('Pos: ' + percentPos +
    }
    else {
        var outColor = new Array(192,192,192);

        if (totalVotes == 0)
            runTotal = 'NA';
        else {
            var degree = 2*totalsPos/totalVotes - 1;
            var i;
            for (i = 0; i < 3; i++)
                outColor[i] = Math.round(white[i] + degree*
                    (degree > 0 ? green[i] - white[i] : white[i] - red[i]
            if (scoring == 'percent') runTotal = Math.round(100*totalsPos
            else                      runTotal = (degree > 0 ? '+' : '')
```

```
        }

        bgcolor = 'rgb(' + outColor.join(',') + ')';
    }
    $('#vote-results_"..@id.."').html(runTotal).css('background-color', b
    $('#results-footer_"..@id.."').html(totalVotes + ' votes');
}

function typeOf(obj) {
  if ( typeof(obj) == 'object' ) {
    if (obj.length) {
      return 'array';
    } else {
      return 'object';
    }
  } else {
      return typeof(obj);
  }
}

// This function is simply used to aid in debugging.
var AlertMe = function(text, status) {

    alert(status + ':' + text);

}

function AddUpdateProperty(pageApi, propName, propValue, callback, totals

    // property name to update
    var propertyname = 'urn:custom.mindtouch.com#' + propName;
    propValueJson = YAHOO.lang.JSON.stringify(propValue);

    // send AJAX request to find the property
    $.ajax({

        // set the uri for the properties API of the current page
        // add parameter to only list properties that match the requested
        // add parameters to convert response to JSON and
        url: pageApi + '/properties?dream.out.format=json&names=' + Deki.

        // set the request HTTP verb
        type: 'GET',
        cache: false,
```

```
// check outcome of the request
complete: function(xhr) {

    // check response status code
    if(xhr.status == 200) {

        // evaluate respone JSON data
        var data = eval('(' + xhr.responseText + ')');

        // read property href and ETag
        var href = data.property && data.property.contents['@href
        var etag = data.property && data.property['@etag'];

        // check we the value was found
        if(href && etag) {

            // send AJAX request to update the property
            $.ajax({

                // set the request HTTP verb
                url: href + '?dream.in.verb=PUT',
                type: 'POST',

                // set the value of the updated property
                data: propValueJson,
                contentType: 'text/plain',
                processData: false,

                // add the 'ETag' header which checks if the prop
                beforeSend: function(xhr) {

                    // NOTE (see Update 1 below): remove content-
                    etag = etag.replace('-gzip', '').replace('-bz

                    xhr.setRequestHeader('ETag', etag);
                    return true;
                },

                // check response status code
                complete: function(xhr) {

                    // check the response status code
                    if(xhr.status == 200) {
                        // alert('Property was successfully updat
                        if(typeof callback == 'function') callbac
```

```
                    if(typeof totalsCallBack == 'function') t
                } else {
                    // alert('Unable to update the property.
                    if(typeof callback == 'function') callbac
                }
            }
        });
    } else {

        // CREATE PAGE PROPERTY
        // send AJAX request using jQuery
        $.ajax({

            // set the uri for the properties API of the curr
            url: pageApi + '/properties',

            // set the request HTTP verb
            type: 'POST',

            // set the value of the new property
            data: propValueJson,
            contentType: 'text/plain',
            processData: false,

            // add the 'Slug' header which sets the property
            beforeSend: function(xhr) {
                xhr.setRequestHeader('Slug', 'urn:custom.minc
                return true;
            },

            // check outcome of the request
            complete: function(xhr) {

                // check the response status code
                if(xhr.status == 200) {
                    // alert('Property was successfully creat
                    if(typeof callback == 'function') callbac
                    if(typeof totalsCallBack == 'function') t
                } else {
                    // alert('Unable to create the property.
                    if(typeof callback == 'function') callbac
                }
            }
        });  // End Ajax (Create Property)
```

```
                }
            } else {
                // alert('Unable to query the property. You don\\'t have
                if(typeof callback == 'function') callback(false, xhr.sta
            }
        }
    });  // End AJAX (Update Property)

}  // End Function AddUpdateProperty


function DeleteProperty(pageApi, propName, callback) {

    // property name to update
    var propertyname = 'urn:custom.mindtouch.com#' + propName;

    // send AJAX request to find the property
    $.ajax({

        // set the uri for the properties API of the current page
        // add parameter to only list properties that match the requested
        // add parameters to convert response to JSON and
        url: pageApi + '/properties?dream.out.format=json&names=' + Deki.

        // set the request HTTP verb
        type: 'GET',
        cache: false,

        // check outcome of the request
        complete: function(xhr) {

            // check response status code
            if(xhr.status == 200) {

                // evaluate respone JSON data
                var data = eval('(' + xhr.responseText + ')');

                // read property href
                var href = data.property && data.property.contents['@href

                // check we the value was found
                if(href) {

                    // send AJAX request to delete the property
                    $.ajax({
```

```
                        // set the request HTTP verb
                        url: href + '?dream.in.verb=DELETE',
                        type: 'POST',

                        // check response status code
                        complete: function(xhr) {

                            // check the response status code
                            if(xhr.status == 200) {
                                // alert('Property was successfully delet
                                if(typeof callback == 'function') callbac
                            } else {
                                // Who cares we were going to trash it an
                                // alert('Unable to delete the property.
                                if(typeof callback == 'function') callbac
                            }
                        }
                    });
                } else {
                    // Who cares we were going to trash it anyhow
                    // alert('Unable to find the property. It may not exi
                    if(typeof callback=='function') callback(true, xhr.st
                }
            } else {
                alert('Unable to query the property. You don\\'t have per
                if(typeof callback=='function') callback(false, xhr.statu
            }
        }
    });

} // End Function DeleteProperty


function ReadProperty(pageApi, propName, callback) {

    // property name to read
    var propertyname = 'urn:custom.mindtouch.com#' + propName;

    // send AJAX request using jQuery
    $.ajax({

        // set the uri for the properties API of the current page
        // add parameter to only list properties that match the requested
        // add parameters to convert response to JSON and
```

```
        url: pageApi + '/properties?dream.out.format=json&names=' + Deki.

        // set the request HTTP verb
        type: 'GET',
        cache: false,
        async: false,

        // check outcome of the request
        complete: function(xhr) {

            // check response status code
            if(xhr.status == 200) {

                // evaluate response JSON data
                var data = eval('(' + xhr.responseText + ')');

                // read property value
                var value = data.property && data.property.contents['#tex

                // check we the value was found
                if(value) {

                    value = YAHOO.lang.JSON.parse(value);
                     // alert('Property was successfully read.\\nIts valu
                    if(typeof callback == 'function') callback(value, xhr
                } else {
                    // alert('Unable to find the property. It may not exi
                    if(typeof callback == 'function') callback(value, xhr
                }
            } else {
                // alert('Unable to query the property. You don\\'t have
                if(typeof callback == 'function') callback(value, xhr.sta
            }
        }
    });

}

    "</script>

    <style type="text/css">"

    "</style>
```

```
</head></html>
```