Template:SimplePoll

```
// SimplePoll
//    originally by neilw, 2009
//    translated by carles.coll, 2009
//
//    Version history:
//        1.00    20-August-2009        neilw           First published v
//        1.01    26-August-2009        neilw           Each poll uses a
//                                                          multiple poll
//                                                      Changed display s
//                                                      Added "bar" argum
//        1.02    21-October-2009       charles.coll    Get the result de
//                                                      Added Fixed Behav
//                                                      Now the parameter
//                                                      Multilanguage sup
//        1.03    23-October-2009       neilw           Cleaned up "View
//        1.04    15-December-2009      neilw           Fixed a timezone-
//        1.05    16-December-2009      neilw           Added Italian tra
//        1.06    13-January-2010       neilw           Fixed language pr
//        1.07    18-March-2010         varosi          Added option to a
//        1.08    29-August-2010        neilw           Fixed HTML entity
//
// Usage:  SimplePoll(question:str, answers:list of str, name:str?, path:
//    question:  The poll question
//    answers:   List of possible answers
//    name:      (optional) Unique name for the poll, in case you want to
//                on the same page properties
//    path:      (optional) Path to page where properties will be stored.
//                by voters!
//    closed:    (optional) Either "true" to close the poll, or a valid d
//                when the poll should close
//    color_bar:       (optional) color of results bars (default: "#B000
//    fixed_behavior:            (optional) default -> Automatic.
//                                edit -> Edit Poll
//                                view -> View Poll Statistics
//                                view_details -> View who answers what
//    language:                 (optional) default ->
//                                                    1st  -> Page langua
//                                                    2nd  -> Site langua
//                                                    3rd  -> 'en-us'
//    halign:                   (optional) Horizontal poll alignment
//                              default -> center
```

```
//
//
// get initialized
//
var question = $0 ?? $question ?? "";
var answers = $1 ?? $answers ?? [];
var fixed_behavior = $fixed_behavior ?? "";
var language = __request.args.language ?? args.language ??
    ( page.language..'' != '' ? page.language : (site.language..'' != ''
var halign = $halign ?? "center";

// -- START LANGUAGE STRINGS
var LANGUAGE_ES = {
    warning_question: "CUIDADO: la pregunta tiene que ser una cadena de c
    warning_no_answers: "CUIDADO: No has pasado respuestas. Así no podrem

    error_no_data_page: "ERROR: no puedo encontrar la pagina con los dato
    error_invalid_close_date: "ERROR: 'cerrada' no es una fecha valida; a
    error_invalid_data: "ERROR: los datos de la encuesta existen pero tie
    error_updating_vote: "ERROR: actualizando el voto ",
    error_creating_vote: "ERROR: creando el voto ",
    error_reading_vote: "ERROR: leyendo el voto ",

    txt_your_answer: "Tu respuesta: ",
    txt_change_my_vote: "cambiar mi voto",
    txt_vote: "votar en esta encuesta",
    txt_view_results: "ver resultados",
    txt_poll_closes_in: "La encuesta se cierra en ",
    txt_votes: " votos contados",
    txt_view_details: "ver detalles",
    txt_poll_closed: "esta encuesta esta cerrada",

    msg_poll_closed: "Ahora la encuesta esta cerrada, lo siento!",

    button_submit: "guardar"
};
var LANGUAGE_DE = {
    warning_question: "WARNING: question must be string or xml",
    warning_no_answers: "WARNING: You have provided no answers.  Not much

    error_no_data_page: "ERROR: can't find page with data store",
    error_invalid_close_date: "ERROR: 'closed' is not a valid date; assum
    error_invalid_data: "ERROR: poll data store exists but has the wrong
    error_updating_vote: "ERROR: updating poll ",
    error_creating_vote: "ERROR: creating poll ",
    error_reading_vote: "ERROR: reding poll ",

    txt_your_answer: "Ihre Antwort: ",
    txt_change_my_vote: "Antworten anpassen",
    txt_vote: "Antworten in dieser Umfrage",
    txt_view_results: "Zeige Resultate",
    txt_poll_closes_in: "Umfrageschluss ist ",
    txt_votes: " Benutzer haben teilgenommen",
    txt_view_details: "zeige Details",
    txt_poll_closed: "diese Umfrage ist beendet",

    msg_poll_closed: "diese Umfrage ist beendet",

    button_submit: "Absenden"
};
var LANGUAGE_IT = {
    warning_question: "ATTENZIONE: la domanda deve essere una stringa o x
    warning_no_answers: "ATTENZIONE: non hai risposto.  Così non riusciam

    error_no_data_page: "ERRORE: non trovo la pagina con i dati del sonda
    error_invalid_close_date: "ERRORE: 'chiuso' non è una data valida; si
    error_invalid_data: "ERRORE: lo store per i dati del sondaggio esiste
    error_updating_vote: "ERRORE: aggiornamento sondaggio ",
    error_creating_vote: "ERRORE: creazione sondaggio ",
    error_reading_vote: "ERRORE: lettura sondaggio ",

    txt_your_answer: "La tua risposta: ",
    txt_change_my_vote: "Cambia il mio voto",
    txt_vote: "Vota il sondaggio",
    txt_view_results: "Risultati",
    txt_poll_closes_in: "Il sondaggio chiude il ",
    txt_votes: " voti",
    txt_view_details: "Dettaglio dei voti",
    txt_poll_closed: "Questo sondaggio è finito",

    msg_poll_closed: "Il sondaggio adesso è chiuso, spiacenti!",

    button_submit: "submit"
};

var TXTS = {
    en: LANGUAGE_DE, 'de-ch': LANGUAGE_DE,
    es: LANGUAGE_ES, 'es-es': LANGUAGE_ES,
    it: LANGUAGE_IT, 'it-it': LANGUAGE_IT
};
```

```
var lg = language;
// -- END LANGUAGE STRINGS

if (question is not str && question is not xml) <p>TXTS[lg].warning_quest
if (#answers == 0) <p>TXTS[lg].warning_no_answers;</p>;
var CONST_POLLDATA_NAME = "poll_default_name";

var poll_name = $2 ?? $name?? CONST_POLLDATA_NAME;
if (poll_name is not str) {
   /* <p>"ERROR: el nombre tiene que ser una cadena de caracteres. Le ass
    let poll_name = CONST_POLLDATA_NAME;
}
var poll_arg = "poll_"..poll_name;
var path = $3 ?? $path;
var p = (path == nil ? page : wiki.getpage(path));
var p_api = null;
if (p == nil) { <p>TXTS[lg].error_no_data_page;</p>;}
   else { let p_api = p.api; }

var closed = $4 ?? $closed ?? false;
var closing_time = false;

if (closed is not bool) {
    if (!date.isvalid(closed)) {
        <p>TXTS[lg].error_invalid_close_date;</p>;
        let closed = false;
    }
    else {      // convert to local time
        let closing_time = date.format(closed,"r");
        let closed = date.compare(date.now, closing_time) > 0;
    }
}
var bar = $5 ?? $color_bar ?? "#B00000";
var viewURI = page.uri & { (poll_arg):"view" };
var viewDetailsURI = page.uri & { (poll_arg):"view_details" };
var editURI = page.uri & { (poll_arg):"edit" };
// Fetch the data store
var data = json.parse(p.properties[poll_name].text ?? '{}');
if (data is not map)
    <p>TXTS[lg].error_invalid_data</p>;
// Now figure out what to show
var vote = (data[user.name] ?? {}).poll;
var can_vote = !closed && !user.anonymous && wiki.pagepermissions(path).u
var has_voted = (vote != nil);
var showform = (fixed_behavior=='edit') ||
            (
             (fixed_behavior=='') &&
                (can_vote && (!has_voted || __request.args[poll_arg] == '
                      && __request.args[poll_arg] != "view"
                      && __request.args[poll_arg] != "view_details"
                 )
            );
<table align=(halign) cellpadding="5" style="background-color:#F4F4F4; bo
    <tr><td align="center" style="font-weight:bold; padding-bottom:10px">
    if (showform) {     // Allow user to enter or edit poll response(s)
        <tr><td align="center"><form id=(@form)>
            <ul style="text-align:left">
                foreach (var opt in answers) {
                    if ((fixed_behavior=='edit') && (__request.args[poll_
                     if(has_voted && vote==__index) { <B>TXTS[lg].txt_you
                    }
                    else
                    {
                     <li style="list-style:none">
                      <input type="radio" name="poll"
                        value=(__index) checked=(has_voted && vote==__inc
                     </li>;
                    }
                 }
            </ul>
            <span style="text-align:center; padding-top:10px">
              if ((fixed_behavior=='edit')&& (__request.args[poll_arg]!='
                  if (can_vote) <a href=(editURI)> has_voted ? TXTS[lg].t
                }
                else
                {
                 <input type="button" value=(TXTS[lg].button_submit) ctor
                    var m = { };
                    Deki.$('form#' + {{@form}} + ' input').each(function
                        if ($(this).attr('name') == 'poll' && $(this).att
                    });
                    Deki.publish({{@channel}}, { {{user.name}}:m }); }"/>
                }
                if(fixed_behavior=='') { <a href=(viewURI)> <span style='
            </span>
            if (closing_time) {
                <br />;
                var ct = date.changetimezone(closing_time, user.timezone)
                let ct = date.format(string.substr(ct,0,string.lastindexo
                <span style="font-size:smaller"> TXTS[lg].txt_poll_closes
            }
        </form></td></tr>;
    }
    else {              // Display poll results
     if ((fixed_behavior=='view') ||
          ( (fixed_behavior=='') &&
            (__request.args[poll_arg] != "view_details")
          )) {
        // Calculate results
        var total = #data;
        <tr><td align="center">
            <span style="font-size:smaller"> total .. TXTS[lg].txt_votes
            <table>
                foreach (var i in num.series(0,#answers-1)) {
                    var result = #map.select(data, "$.value.poll == "..i)
                    var pct = num.round(100*result/num.max(total,1),1);
                    var width = num.round(2 * pct, 0);
                    <tr>
                        <td> answers[i] </td>
                        <td>
                            <img src="/skins/common/icons/icon-trans.gif'
                            <img src="/skins/common/icons/icon-trans.gif'
                            " " .. pct .. "% (" .. result .. ")"
                        </td>
                    </tr>;
                }
            </table>
            if (fixed_behavior=='') {
               <span style="text-align:center; padding-top:10px; font-siz
                  <a href=(viewDetailsURI)>TXTS[lg].txt_view_details</a>;
                  &nbsp;;&nbsp;;
                  if (can_vote) <a href=(editURI)> has_voted ? TXTS[lg].t
                  else if (closed) TXTS[lg].txt_poll_closed;
              </span>;
             }
        </td></tr>;
        }
        else {
        // Show details
        <tr><td align="center">
              <table>
                <tr>
                foreach (var i in num.series(0,#answers-1)) {
                        <th style="text-align:center; border:1px solid #6
                   }
                </tr>;

                <tr>
                foreach (var i in num.series(0,#answers-1)) {
                    var result = Map.Keys(map.select(data, "$.value.poll
                        <td valign="top" style="text-align:center">
                            if (#result) {
                                <span style="font-size:.8em">"("..#result
                                <br />;
                            }
                            foreach (var who in result) {
                                if (__index) <br />;
                                who;
                            }
                        </td>
                    }
                </tr>;

            </table>
            if(fixed_behavior=='') {
              <span style="text-align:center; padding-top:10px; font-size
                <a href=(viewURI)> <span style="font-size:smaller">TXTS[l
                &nbsp;;&nbsp;;
                if (can_vote) <a href=(editURI)> has_voted ? TXTS[lg].txt
                else if (closed) "esta encuesta esta cerrada";
              </span>;
             }
        </td></tr>;
        }
    }
</table>;

// Code to update the page properties, largely cribbed from SteveB's "Upd
dekiapi();
var store = poll_name;
<script type="text/javascript"> "
   Deki.subscribe('"..@channel.."', null, function(c, m, d) {
        var closed = " .. json.emit(closed) .. ";
        var closing_time = " .. json.emit(closing_time) .. ";
        var d = new Date();
        if (closed || (closing_time && d.getTime() > Date.parse(closing_t
            alert('"..TXTS[lg].msg_poll_closed.."');
            window.location.href = '" .. page.uri .. "';
            return;
        }
```

```
            var prop = 'urn:custom.mindtouch.com#'  + '"..store.."';
            Deki.Api.ReadPageProperty('"..p_api.."', prop, function(result) {
                var data = eval('(' + (result.value || '{}') + ')');
                for (var k in m) data[k] = m[k];
                if(result.etag)
                    Deki.Api.UpdatePageProperty(result.href, YAHOO.lang.JSON.
                        function() { window.location.href = '" .. page.uri ..
                        function(result) { alert('"..TXTS[lg].error_updating_
                result.status + ' - ' + result.text + ')'); }
                    );
                else
                    Deki.Api.CreatePageProperty('"..p_api.."', prop, YAHOO.la
                        function() { window.location.href = '" .. page.uri ..
                        function(result) { alert('"..TXTS[lg].error_creating_
                result.status + ' - ' + result.text + ')'); }
                    );
            },
            function(result) { alert('"..TXTS[lg].error_reading_vote.." (esat
                result.status + ' - ' + result.text + ')'); }
            );
        }, null);
    " </script>
```