```
/***
    USAGE: ToDoList()
        Embed a TODO-list into a page.  The TODO list allows users to
create new items, assign them to other users, and set deadlines.
        Items that are due are highlighted in yellow, items passed due
are shown in red, and completed items are struck out.  The TODO-
        list can be added to any page.  The information of the TODO-list
is stored as a page property.

    NOTES:
        * only 1 TODO-list can be embedded per page
        * this template requires the DekiAPI script extension
        * this template requires the jQuery script extension

    CHANGE HISTORY:
        1.0
            * initial version
        1.1
            * show error message if dekiapi or jquery extensions are
missing
            * verify if user is logged in or not; if not, disable add &
update controls

        1.2
            * allow custom user list to be passed in
            * publish changes to a channel
***/

var userlist = $userlist;
var publish = $publish ?? 'default';
var message = $message ?? { };
var allowed;

if(userlist) {

    // convert list of names to list of user objects
    let userlist = [
        (u is map) ? u : wiki.getuser(u)
        foreach var u in userlist
    ];
    let allowed = [ string.tolower(u.name) foreach var u in userlist ];
} else if(site.usercount <= 500) {

    // get list of users from site
    let userlist = site.users;
}


// check if required extensions are installed
```

```
if(__env.dekiapi is nil) {
    <span style="color: red; font-weight: bold">"ERROR: Template:TODO
requires DekiApi extension to be installed."</span>
} else if(__env.jquery is nil) {
    <span style="color: red; font-weight: bold">"ERROR: Template:TODO
requires jQuery extension to be installed."</span>
} else {

    // includes
    dekiapi();
    jquery.ui('smoothness');

    // input form
    var disabled = (

        // anonymous users can't edit
        user.anonymous ||

        // user cannot update page
        (wiki.pagepermissions().update is nil) ||

        // user is not listed in custom user list
        (allowed && !list.contains(allowed, string.tolower(user.name))
    ) ? "disabled" : nil);
    <div id="deki-todo">
        <form>
            <input id="todoid" type="hidden" value="" />
            <div class="todo-assign">" Task: ";</div>
            <div class="todo-who">" Who: ";

                // check if the userlist is provided, otherwise use a
simple textbox instead of a pre-filled listbox
                if(userlist) {
                    <select id="todowho"disabled=(disabled) >
                        foreach(var u in userlist where !u.anonymous) {
                            <option value=(u.name) selected=((u.id ==
user.id) ? 'selected' : nil)>
                                    u.name
                            </option>
                        }
                    </select>
                } else {
                    <input id="todowho" type="text" value=(userlist)
disabled=(disabled) />
                }
            </div>
            <div class="todo-what">"What: "; <input id="todowhat"
type="text" disabled=(disabled) /></div>
            <div class="todo-when">" When: "; <input id="todowhen"
type="text" value=(date.format(date.now, 'MM/dd/yyyy'))
```

```
disabled=(disabled) ctor="$this.datepicker();" /></div>
            <div class="todo-submit">
            <input id="todoadd" type="submit" value="Add"
disabled=(disabled) ctor="
                when($this.click) {
                    @todoupdate({
                        index: #todoid.val(),
                        entry: {
                            what: #todowhat.val(),
                            who: #todowho.val(),
                            when: #todowhen.val(),
                            author: {{user.name}},
                            done: false
                        }
                    });
                    @todoreset();
                    return false;
                }
                when(#tododelete.click) {
                    @todoupdate({ index: #todoid.val(), entry: null });
                    @todoreset();
                }
                when(@todoreset) {
                    #todoid.val('');
                    #todowhat.val('');
                    #todoadd.val('Add');
                    #tododelete.hide().removeAttr('disabled');
                    #todoadd.blur();
                }
                when(@todoedit) {
                    MindTouch.Deki.ReadPageProperty(null,
'urn:custom.mindtouch.com#todo', function(result) {
                        var data = YAHOO.lang.JSON.parse(result.value ||
'[]');
                        #todoid.val(@todoedit.index);
                        #todowhat.val(data[@todoedit.index].what);
                        #todowhen.val(data[@todoedit.index].when);
                        #todowho.val(data[@todoedit.index].who);
                        #todoadd.val('Update');
                        #tododelete.show();
                    }, function(result) {
                        alert('An error occurred trying to read the TODO
list (status: ' + result.status + ' - ' + result.text + ')');
                    });
                }
                when(@todoupdate) {
                    MindTouch.Deki.ReadPageProperty(null,
'urn:custom.mindtouch.com#todo', function(result) {
                        var op;
                        var data = eval('(' + (result.value || '[]') +
```

```
')');
                            if(@todoupdate.index !== '') {
                                if(@todoupdate.entry === null) {
                                    op = 'remove';
                                    data.splice(@todoupdate.index, 1);
                                } else {
                                    op = 'update';
                                    $.extend(data[@todoupdate.index],
@todoupdate.entry);
                                }
                            } else {
                                op = 'add';
                                @todoupdate.index = data.length;
                                data.push(@todoupdate.entry);
                            }
                            var completed = function() {
                                var msg = {{ message }};
                                msg.op = op;
                                msg.index = @todoupdate.index;
                                msg.entry = @todoupdate.entry;
                                msg.list = data;
                                Deki.publish({{ publish }}, msg);
                                MindTouch.Deki.Reload(#todolist, null,
function() { ToDoWireControls(); });
                            };
                            if(result.etag) {

MindTouch.Deki.UpdatePageProperty(result.href,
YAHOO.lang.JSON.stringify(data), result.etag, completed, function(result)
{
                                    alert('An error occurred trying to update
the TODO list (status: ' + result.status + ' - ' + result.text + ')');
                                });
                            } else {
                                MindTouch.Deki.CreatePageProperty(null,
'urn:custom.mindtouch.com#todo', YAHOO.lang.JSON.stringify(data),
completed, function(result) {
                                    alert('An error occurred trying to create
the TODO list (status: ' + result.status + ' - ' + result.text + ')');
                                });
                            }
                    }, function(result) {
                        alert('An error occurred trying to read the TODO
list (status: ' + result.status + ' - ' + result.text + ')');
                    });
                }
            " />
            <input id="tododelete" type="button" style="display: none;"
value="Delete" />
            </div>
```

```
        </form>

        // add index to each todo item
        var todo = json.parse(page.properties["todo"].text ?? "") ?? [ ];
        let todo = [ item .. { index: __index } foreach var item in todo
];
        let invaliddates = [ item foreach var item in todo where
!date.isvalid(item.when) ];
        let todo = [ item foreach var item in todo where
date.isvalid(item.when) ];
        let todo = list.sort(todo, _, _, "date.compare($left.when,
$right.when)") .. invaliddates;
        <table id="todolist" cellspacing="0" cellpadding="0">
            <colgroup>
                <col width="20" />
                <col width="20%" />
                <col />
                <col width="15%" />
                <col width="75" />
            </colgroup>
            <tr>
                <th>string.nbsp</th>
                <th>"When"</th>
                <th>"What"</th>
                <th>"Who"</th>
                <th>string.nbsp</th>
            </tr>
            foreach(var entry in todo) {
                var class = "";
                var now = date.now;
                if(entry.done) let class..= "completed ";
                else if(date.isvalid(entry.when)) {
                    if(date.issameday(now, entry.when)) let class ..=
"due-today ";
                    else if(date.isafter(now, entry.when)) let class ..=
"due-past ";
                }
                var title = "Created by " .. entry.author;
                <tr class=(class)>
                    <td><input type="checkbox" name="done"
id=(entry.index) class="todochange" value=(entry.index)
checked=(entry.done ? 'checked' : nil) disabled=(disabled) /></td>
                    <td title=(title)><span>entry.when</span></td>
                    <td title=(title)><span>entry.what</span></td>
                    <td title=(title)><span>entry.who</span></td>
                    <td class="todo-edit">
                        if(disabled) {
                            string.nbps;
                        } else {
                            <a href="#" rel=(entry.index)>"edit"</a>
```

```
                        }
                    </td>
                </tr>
            }
        </table>
    </div>

    // global script code
    <script type="text/jem">"
    function ToDoWireControls() {
        $('#todolist a').bind('click', function() {
            @todoedit({ index: $(this).attr('rel') });
            #todowhat.select();
            return false;
        });
        $('.todochange').bind('click', function() {
            var $this = $(this);
            @todoupdate({ index: $this.val(), entry: { done: this.checked
} });
        });
    }
    ToDoWireControls();
    "</script>

    // styles
    <html>
        <head>
            <style type="text/css">"
    //--- Start TODO styles ---
    #deki-todo {
        margin: 0;
        padding: 4px;
        width: 98%;
    }
    #deki-todo table {
        width: 100%;
        margin: 0;
    }
    #deki-todo table td {
        border-bottom: 1px solid #000;
        padding: 4px 2px;
    }
    #deki-todo tr td.todo-edit {
        text-align: right;
        padding-right: 8px;
    }
    #deki-todo tr.due-today td {
        background-color: #ff0;
    }
    #deki-todo tr.completed td {
```

```css
        background-color: #fff;
    }
    #deki-todo tr.completed td span {
        text-decoration: line-through;
    }
    #deki-todo tr.due-past td,
    #deki-todo tr.due-past td.todo-edit a {
        color: #f00;
    }
    #deki-todo table th {
        font-size: 14px;
        font-weight: bold;
        text-align: left;
        border-bottom: 2px solid #000;
    }
    #deki-todo form {
        padding: 4px;
        background-color: #000;
        color: #fff;
        overflow: hidden;
        margin-bottom: 8px;
    }
    #deki-todo div.todo-assign {
        font-weight: bold;
    }
    #deki-todo form,
    #deki-todo form input,
    #deki-todo form select {
        font-size: 11px;
    }
    #deki-todo form div {
        padding-right: 8px;
        float: left;
    }
    #deki-todo div.todo-submit {
        float: right;
        padding-right: 0px;
    }
    #deki-todo div.todo-submit input {
        margin-right: 4px;
    }
    #deki-todo table tr td.todo-edit a {
        background: url(/skins/common/icons/silk/page_white_edit.png)
no-repeat center left;
        color: #000;
        text-decoration: none;
        padding: 3px 3px 3px 21px;
        font-size: 11px;
    }
    #deki-todo table tr td.todo-edit a:hover {
```

```
            text-decoration: underline;
        }
        //--- End TODO styles ---
        "
                </style>
            </head>
        </html>
}
```