

ECrudDesignPattern-FormProcessing

Form processing

<http://www.bluexml.org/static/images/form-processing-dp.gif>

Controller

A controller manages and detects the mode the user is waiting for. At least 3 cases may be distinguished :

- Create mode :
 - init with default values
 - call display service until validation is ok (thanks to the use of continuation)
 - call create service
- Update mode :
 - init with existing values
 - call display service, after filtering fields, until validation is ok
 - call update service
- Search mode :
 - init with default values for search mode
 - call search service, after filtering fields, until validation is ok
 - call query service

This behavior may be coded with the following javascript in Cocoon 2.1 :

```

function dispatch() {
    // Initialization from default files
    // -----
    if (('create' == action)||('search' == action)) {
        documentURI = tmpPath + "/tmp/" + docType + "-new.xml";
    } else {
        documentURI = cocoon.parameters["documentURI"];
    }

    var bindingURI = cocoon.parameters["bindingURI"];
    var formToShowPrefix = cocoon.parameters["formToShow"];
    var formToShow;

    if (action.indexOf('review',0) != -1) {
        formToShow = formToShowPrefix + "-review";
    } else if ('read' == action) {
        formToShow = formToShowPrefix + "-read";
    } else if ('create' == action) {
        formToShow = formToShowPrefix + "-create";
    } else if ('search' == action) {
        formToShow = formToShowPrefix + "-search";
    }

    if (docType) {
        // Initialization from code. It should be better located in form definition
        // -----
        var factory = new FormFactory();
        var aObject = factory.createForm(this, docType);

        // Call display service
        // -----
        var result = display(action, aObject, form, documentURI, bindingURI, formToShow, tmpPath);

        // Call service
        // -----
        if (100 == result) {
            // init problem
        } else if ( 3 == result) {
            var target = "process-errorsuffix";
            cocoon.sendPage(target);
        } else {
            if ('create' == action) {
                var target = "process-create-" + aObject.getData();
                cocoon.sendPage(target);
            } else if ('search' == action) {
                var target = "process-qbe-tmp-" + aObject.getData();
                cocoon.sendPage(target);
            } else {
                // Review action
                var target = "process-update-" + docid + "-" + aObject.getData();
                cocoon.sendPage(target);
            }
        }
    } else {
        var target = "read/process-list-" + doctype + "-" + category;
        cocoon.sendPage(target);
    }
}

```