

Shows row details of a selected item - form configuration

This is a configuration on an example presented on this page: [Shows row details of a selected item](#)

```
<ac:structured-macro ac:macro-id="aad1cc0c-08da-41b5-bfb0-40a6d79653dd" ac:name="confiorm" ac:schema-version="1">
  <ac:parameter ac:name="formName">formB</ac:parameter>
  <ac:rich-text-body>
    <ac:structured-macro ac:macro-id="7996200c-c6cf-4ab8-89dc-1de664f7757a" ac:name="confiorm-entry-register"
ac:schema-version="1">
      <ac:parameter ac:name="restrictions">sash</ac:parameter>
      <ac:rich-text-body>
        <p>
          <br/>
        </p>
      </ac:rich-text-body>
    </ac:structured-macro>
  <p>
    <ac:structured-macro ac:macro-id="3dfdd337-88cf-4f7c-bf05-06edc6248d4d" ac:name="confiorm-field-
definition" ac:schema-version="1">
      <ac:parameter ac:name="fieldName">t</ac:parameter>
      <ac:parameter ac:name="fieldLabel">t</ac:parameter>
      <ac:parameter ac:name="type">text</ac:parameter>
    </ac:structured-macro>
  </p>
  <p>
    <ac:structured-macro ac:macro-id="73e5ab0f-0466-44b1-88cc-46e5ecaecce6" ac:name="confiorm-field-
definition" ac:schema-version="1">
      <ac:parameter ac:name="fieldName">ta</ac:parameter>
      <ac:parameter ac:name="fieldLabel">ta</ac:parameter>
      <ac:parameter ac:name="type">textarea</ac:parameter>
    </ac:structured-macro>
  </p>
  <p>
    <ac:structured-macro ac:macro-id="993421f6-df23-47fb-8e2d-00c853ac72fe" ac:name="confiorm-field-
definition" ac:schema-version="1">
      <ac:parameter ac:name="fieldName">label</ac:parameter>
      <ac:parameter ac:name="fieldLabel">Label</ac:parameter>
      <ac:parameter ac:name="values">>false[label1=label1|label2=label2|label3=label3|]</ac:parameter>
      <ac:parameter ac:name="type">multi</ac:parameter>
    </ac:structured-macro>
  </p>
</ac:rich-text-body>
</ac:structured-macro>
<ac:structured-macro ac:macro-id="3f158dec-1332-4e74-abe1-d7a1af8debd1" ac:name="confiorm" ac:schema-version="1">
  <ac:parameter ac:name="formName">formA</ac:parameter>
  <ac:rich-text-body>
    <p>
      <ac:structured-macro ac:macro-id="c965d8ca-69a9-45be-98dc-cdbe25763521" ac:name="confiorm-field-
definition" ac:schema-version="1">
        <ac:parameter ac:name="fieldName">option</ac:parameter>
        <ac:parameter ac:name="fieldLabel">Option</ac:parameter>
        <ac:parameter ac:name="values">[111510524|formB|t|true|]</ac:parameter>
        <ac:parameter ac:name="type">smartselect</ac:parameter>
      </ac:structured-macro>
    </p>
    <p>
      <ac:structured-macro ac:macro-id="87d8526a-ae8b-4e08-98b4-207034a23c8c" ac:name="confiorm-field-
definition-rules" ac:schema-version="1">
        <ac:parameter ac:name="condition">!option:[empty]</ac:parameter>
        <ac:parameter ac:name="values">loadFormBDetails(formName, formId)</ac:parameter>
        <ac:parameter ac:name="action">Run custom JavaScript</ac:parameter>
        <ac:parameter ac:name="actionFieldName">option</ac:parameter>
      </ac:structured-macro>
    </p>
    <p>
      <ac:structured-macro ac:macro-id="a77e824a-5ee5-4a6c-b3ca-4c1ceda04f35" ac:name="confiorm-field-
definition-rules" ac:schema-version="1">
        <ac:parameter ac:name="condition">option:[empty]</ac:parameter>
      </ac:structured-macro>
    </p>
  </ac:rich-text-body>
</ac:structured-macro>
```

```

        <ac:parameter ac:name="action">Hide container</ac:parameter>
        <ac:parameter ac:name="actionFieldName">option</ac:parameter>
        <ac:parameter ac:name="values2">#formBDetails</ac:parameter>
        <ac:parameter ac:name="withReverseRule">true</ac:parameter>
    </ac:structured-macro>
</p>
</ac:rich-text-body>
</ac:structured-macro>
<ac:structured-macro ac:macro-id="f7fbb3be-70f5-459f-9bd5-6472c649590c" ac:name="html" ac:schema-version="1">
    <ac:plain-text-body><![CDATA[<script type="text/javascript">

        var apiCall = AJS.params.contextPath + '/rest/confiforms/1.0/search/' + AJS.Meta.get("page-id") + '/formB
        /id:';

        function loadFormBDetails(formName, formId){
            AJS.$.ajax({
                url: apiCall + AJS.$('#' + formId).find('#i_option').val(),
                type: "GET",
                async: false,
                cache: false,
                success: function (data) {
                    var transformTemplate = [{"name": "Created", "prop": "created", "renderer": "renderDate"}, {"name": "
                    TextArea", "prop": "fields.ta"}, {"name": "Label", "prop": "fields.label", "renderer": "renderLabel"}];
                    buildHtmlTable('#formBDetailsTable', transformData(data.list.entry, transformTemplate));
                }
            });
        }
    </script>]]></ac:plain-text-body>
</ac:structured-macro>
<ac:structured-macro ac:macro-id="975cf00d-358e-4c8b-9e0e-bae8150b1f38" ac:name="confiiform-entry-register" ac:
schema-version="1">
    <ac:parameter ac:name="formName">formA</ac:parameter>
    <ac:parameter ac:name="type">Embedded</ac:parameter>
    <ac:rich-text-body>
        <p>
            <ac:structured-macro ac:macro-id="cfb58f6d-6928-4f93-ad4e-9e6747b7a4fc" ac:name="confiiform-field" ac:
            schema-version="1">
                <ac:parameter ac:name="fieldName">option</ac:parameter>
                <ac:parameter ac:name="withLabel">true</ac:parameter>
            </ac:structured-macro>
        </p>
        <p>
            <br/>
        </p>
    </ac:structured-macro ac:macro-id="7aec3ef9-7368-4cfd-ba9d-5c9557e0e5d1" ac:name="html" ac:schema-version="1"
    >
        <ac:plain-text-body><![CDATA[<div id="formBDetails" style="display:none">
        <table id="formBDetailsTable" class="au1">
        </table>
        </div>]]></ac:plain-text-body>
        </ac:structured-macro>
    </ac:rich-text-body>
</ac:structured-macro>
<ac:structured-macro ac:macro-id="2eb2a8d2-3a19-473c-87eb-8eb297cfc508" ac:name="html" ac:schema-version="1">
    <ac:plain-text-body><![CDATA[<script type="text/javascript">
    // based on an answer found at https://stackoverflow.com/questions/5180382/convert-json-data-to-a-html-table

    // Builds the HTML Table out of myList.
    function buildHtmlTable(selector, dataList) {
        AJS.$(selector).html('');
        var columns = addAllColumnHeaders(dataList, selector);

        for (var i = 0; i < dataList.length; i++) {
            var row$ = AJS.$('<tr/>');
            for (var colIndex = 0; colIndex < columns.length; colIndex++) {
                var cellValue = dataList[i][columns[colIndex]];
                if (cellValue == null) cellValue = "";
                row$.append(AJS.$('<td/>').html(cellValue));
            }
            $(selector).append(row$);
        }
    }
    </script>]]></ac:plain-text-body>
    </ac:structured-macro>

```

```

}

// Adds a header row to the table and returns the set of columns.
// Need to do union of keys from all records as some records may not contain
// all records.
function addAllColumnHeaders(dataList, selector) {
    var columnSet = [];
    var headerTr$ = AJS.$('<tr/>');

    for (var i = 0; i < dataList.length; i++) {
        var rowHash = dataList[i];
        for (var key in rowHash) {
            if ($.inArray(key, columnSet) == -1) {
                columnSet.push(key);
                headerTr$.append(AJS.$('<th/>').html(key));
            }
        }
    }
    $(selector).append(headerTr$);

    return columnSet;
}

function findProp(obj, prop, defval){
    if (typeof defval == 'undefined') defval = null;
    prop = prop.split('.');
    for (var i = 0; i < prop.length; i++) {
        if (typeof obj[prop[i]] == 'undefined')
            return defval;
        obj = obj[prop[i]];
    }
    return obj;
}

function transformData(dataArray, transformTemplate) {
    var transformed = [];
    for (var i = 0; i < dataArray.length; i++) {
        var dataRow = dataArray[i];
        var transformedRow = {};
        for (var j = 0; j < transformTemplate.length; j++) {
            var propValue = findProp(dataRow, transformTemplate[j].prop);

            if (propValue != null && 'renderLabel' == transformTemplate[j]['renderer']) {
                var propValues = JSON.parse(propValue);

                var values = '';
                for (var k = 0; k < propValues.length; k++) {
                    var val = findProp(propValues[k], transformTemplate[j]['renderer']);
                    if (val != null) {
                        if (values != '') {
                            values = values + ', ';
                        }
                        values = values + val;
                    }
                }
                transformedRow[transformTemplate[j].name] = values;
            } else if (propValue != null && 'renderDate' == transformTemplate[j]['renderer']) {
                var newDate = new Date();
                newDate.setTime(propValue);
                transformedRow[transformTemplate[j].name] = newDate.toUTCString();
            } else {
                if (propValue != null) {
                    propValue = propValue.replace(/(?:\r\n|\r|\n)/g, '<br>');
                }
                transformedRow[transformTemplate[j].name] = propValue;
            }
        }
        transformed.push(transformedRow);
    }
    return transformed;
}

```

</script>]]></ac:plain-text-body>
</ac:structured-macro>