

# User Manual for MasterCRM Kanban View for Creatio

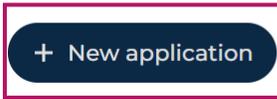
**MASTERCRM.**

# Table of Contents

Installation.....	3
Add Kanban view for object list page.....	4

## Installation

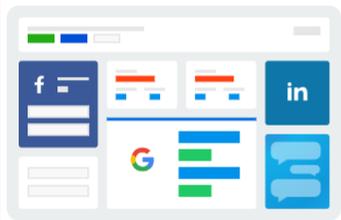
Install MasterCRM Kanban View for Creatio through Application hub



🔍 Search

, either directly from the marketplace (<https://academy.creatio.com/docs/8.x/no-code-customization/customization-tools/app-management/apps-management#title-2304-1>)

### SELECT THE APP



**MARKETPLACE POWERED**  
Install a ready-to-go app from our Marketplace



**CUSTOMER 360**  
Build a 360-degree view of your customers and personalize their engagement journeys

or from downloaded installation archive (<https://academy.creatio.com/docs/8.x/no-code-customization/customization-tools/app-management/apps-management#title-2304-2>)

### SELECT THE APP

#### EVENT MANAGEMENT

Plan, promote, and manage your marketing events to maximize its success

#### KNOWLEDGE MANAGEMENT

Collect knowledge about all services and known errors in a unified knowledge base



#### ORDER AND CONTRACT MANAGEMENT

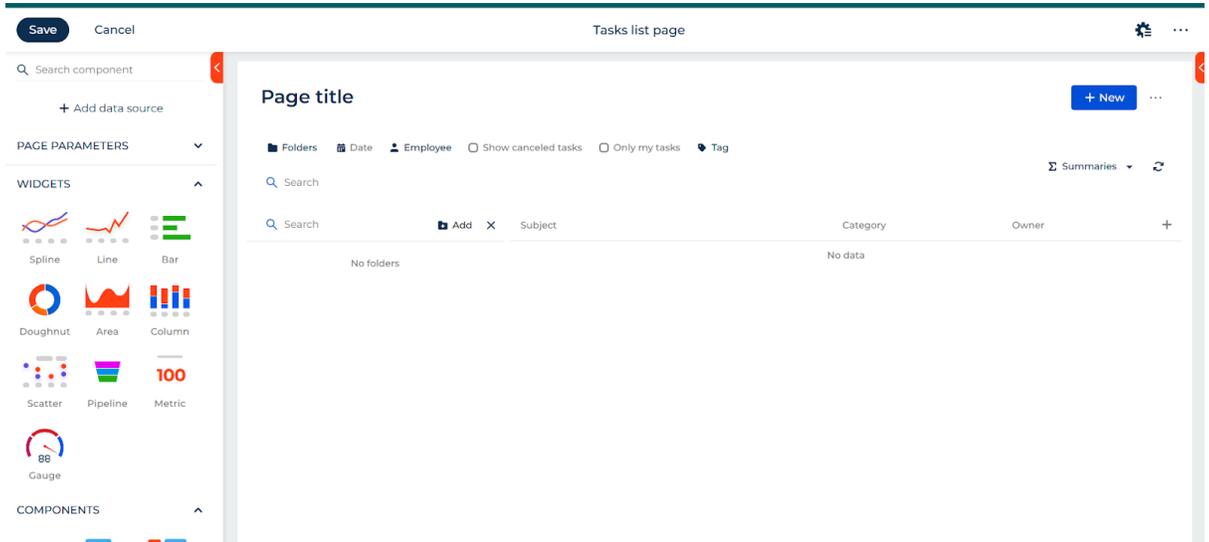
Create, manage and monitor your orders and contracts. Ensure timely fulfillment of all your



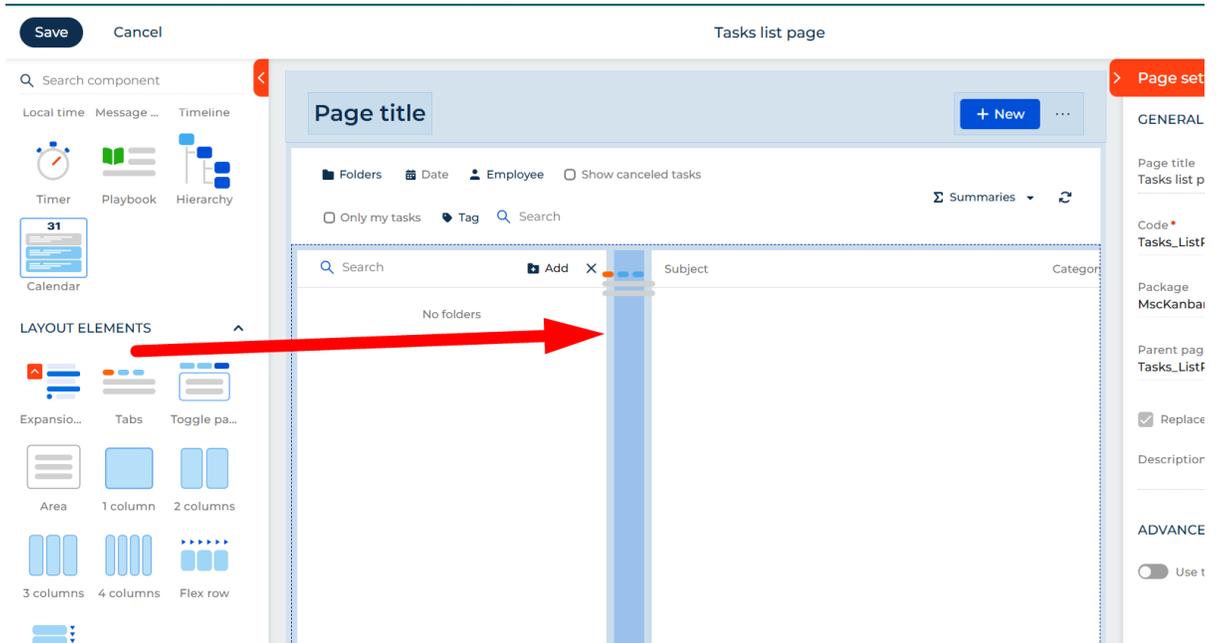
**INSTALL FROM FILE**  
Install a ready-to-go app from a file

## Add Kanban view for object list page

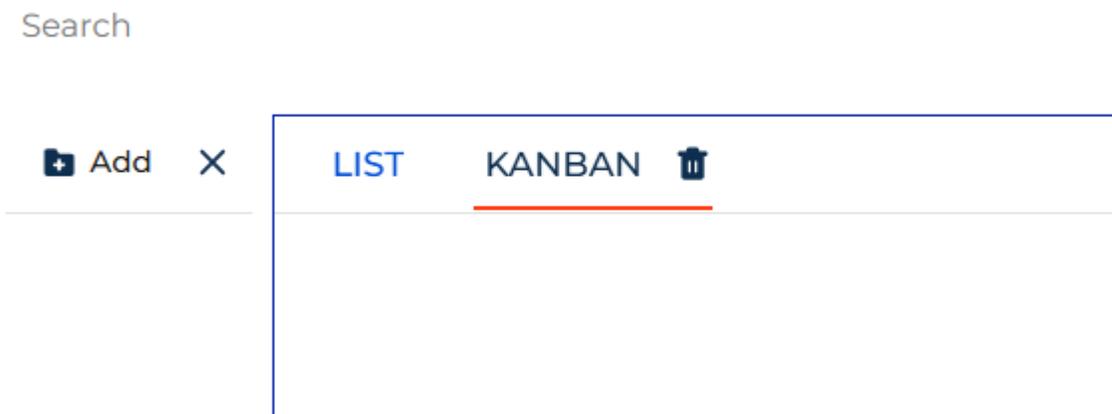
1. Open selected list page designer



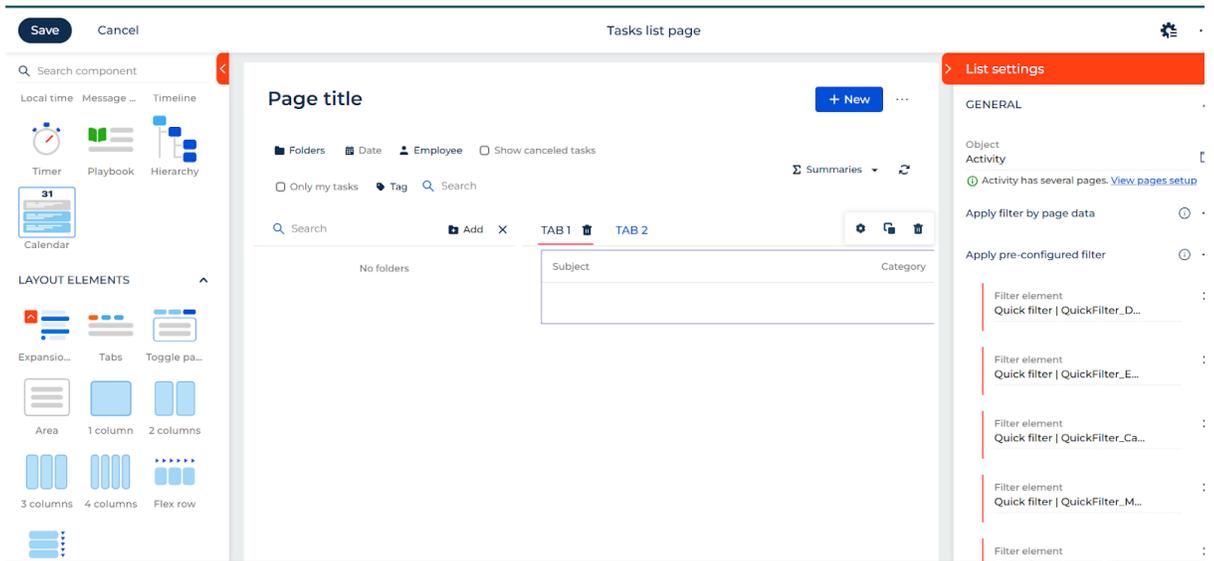
2. Add Tabs component onto the page



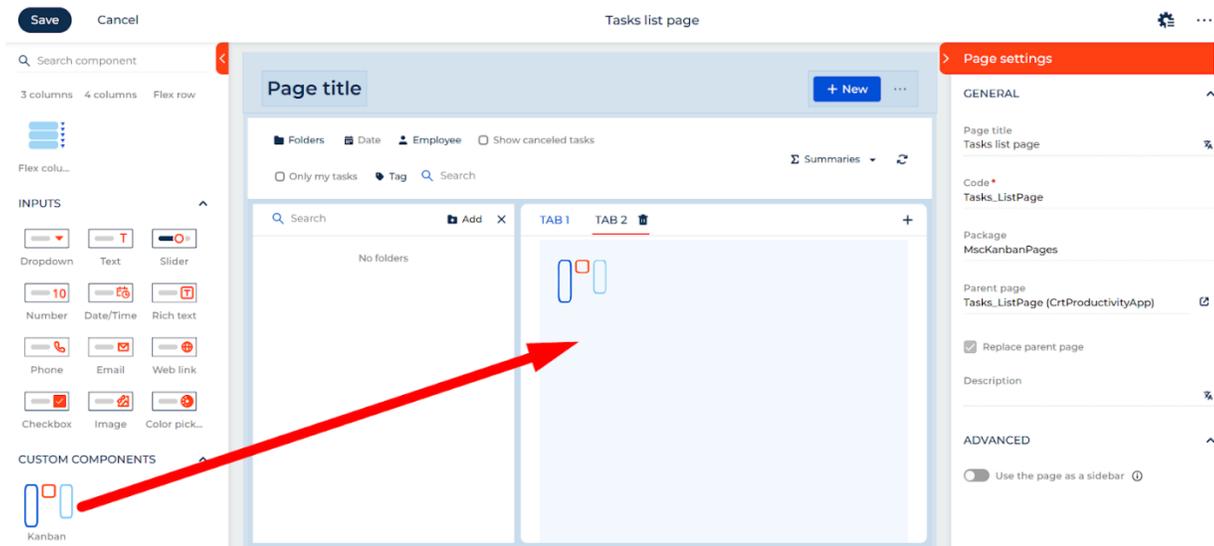
3. Create, for example, 2 tab: *List* and *Kanban*.



4. Move record list to the *List* tab.



5. Insert Kanban View to the *Kanban* tab



6. Save the page. Open page source code for edit

7. Find inserted msc.Kanban element  
Change element settings

```
"values": {
  "type": "msc.Kanban",
  "listInp": "$Items",
  "idField": "PDS_Id",
  "statusField": "PDS_QualifyStatus",
  "nameField": "PDS_LeadName",
  "additionalFields": [
    {
      "label": "Phone",
      "code": "PDS_MobilePhone"
    }
  ],
  "allStatuses": "$MscAllStatuses",
  "triggerItemsChange": "$triggerItemsChange",
  "loadOnChange": true,
  "statusChanged": {
    "request": "msc.SaveRecordRequest"
  },
  "mscOpenPage": {
    "request": "msc.OpenPageRequest"
  },
  "layoutConfig": {}
},
```

where:

listInp – name of the *Item* attribute for your list page (you can find it in the page attributes, by default on the list pages – *\$Items*)  
statusField – field for the Kanban stages (Kanban columns will be set by this field)  
nameField – object field for use as a tile title  
additionalFields – array of the additional fields to view on the every tile

All specified fields must be available in the page viewModelConfigDiff (viewModelConfig)

```
558     viewModelConfig: /**SCHEMA_VIEW_MODEL_CONFIG*/{
559       "attributes": {
560         "Items": {
561           "viewModelConfig": {
562             "attributes": {
563               "PDS_Id": {
564                 "modelConfig": {
565                   "path": "PDS.Id"
566                 }
567               },
568               "PDS_LeadName": {
569                 "modelConfig": {
570                   "path": "PDS.LeadName"
571                 }
572               },
573               "PDS_CreatedOn": {
574                 "modelConfig": {
575                   "path": "PDS.CreatedOn"
576                 }
577               },
578               "PDS_CreatedBy": {
579                 "modelConfig": {
580                   "path": "PDS.CreatedBy"
581                 }
582               },
583               "PDS_MobilePhone": {
584                 "modelConfig": {
585                   "path": "PDS.MobilePhone"
586                 }
587               },
588               "PDS_QualifyStatus": {
589                 "modelConfig": {
590                   "path": "PDS.QualifyStatus"
591                 }
592               }
593             }
594           }
595         }
596       }
597     },
598     ],
599     /**SCHEMA_VIEW_CONFIG_DIFF*/,
600     viewModelConfigDiff: /**SCHEMA_VIEW_MODEL_CONFIG_DIFF*/[
601       {
602         "operation": "remove",
603         "path": [
604           "attributes",
605           "Items",
606           "viewModelConfig",
607           "attributes"
608         ],
609         "properties": [
610           "PDS_ShowInScheduler"
611         ]
612       },
613       {
614         "operation": "merge",
615         "path": [
616           "attributes",
617           "Items",
618           "viewModelConfig",
619           "attributes"
620         ],
621         "values": {
622           "PDS_Status": {
623             "modelConfig": {
624               "path": "PDS.Status"
625             }
626           }
627         }
628       }
629     ],
630     }
631 ]
```

, as well as (in `viewModelConfigDiff`) required attributes `triggerItemsChange`, `MscAllStatuses` (empty), `MscIsShowMessage`

```
{
  "operation": "merge",
  "path": [
    "attributes"
  ],
  "values": {
    "triggerItemsChange": {
      "value": 0
    },
    "MscAllStatuses": {},
    "MscIsShowMessage": {
      "value": false
    }
  }
}
```

You can add there sorting preferences and rows quantity for initial loading also

```
558 ▼   viewModelConfig: /**SCHEMA_VIEW_MODEL_CONFIG*/{
559 ▼     "attributes": {
560 ▼       "Items": {
561 ▶         "viewModelConfig": {↔},
595 ▼         "modelConfig": {
596           "path": "PDS",
597 ▼         "pagingConfig": {
598           "rowCount": 500
599         },
600 ▼         "sortingConfig": {
601           "attributeName": "ItemsSorting",
602 ▼         "default": [
603 ▼           {
604             "direction": "asc",
605             "columnName": "LeadName"
606           }
607         ]
608       },
609 ▶     "filterAttributes": [↔]
619   }
620 },
```

```

"modelConfig": {
  "path": "PDS",
  "pagingConfig": {
    "rowCount": 100
  },
  "sortingConfig": {
    "attributeName": "ItemsSorting",
    "default": [
      {
        "direction": "asc",
        "columnName": "LeadName"
      }
    ]
  }
}

```

To display lookup field in the tile, add attribute `displayValue` in the `additionalFields` config

```

"additionalFields": [
  {
    "label": "Start",
    "code": "PDS_StartDate"
  },
  {
    "label": "Category",
    "code": "PDS_ActivityCategory.displayValue"
  }
],
"allstatuses": "#McAllStatuses"

```

8. Add `"@creatio-devkit/common"` into schema dependencies:

```

define("Leads_ListPage", /**SCHEMA_DEPS*/["@creatio-devkit/common"]/**SCHEMA_DEPS*/, function/**SCHEMA_ARGS*/(sdk)/**SCHEMA_ARGS*/ {
  return {

```

9. Add required handlers

a.

```
{
  request: "msc.OpenPageRequest",
  handler: async (request, next) => {
    var id = await request.$initialEvent.detail.taskId;
    const handlerChain = sdk.HandlerChainService.instance;
    await handlerChain.process({
      type: "crt.UpdateRecordRequest",
      entityName: "Lead",
      $context: request.$context,
      recordId: id
    });

    await next?.handle(request);
  }
},
```

Fill *entityName* with the list page entity schema name (Lead for example)

b.

```
{
  request: "crt.HandleViewModelInitRequest",
  handler: async (request, next) => {
    await next?.handle(request);
    var payload = {
      entityName: "Lead",
      statusField: "QualifyStatusId",
      statusOrderField: "StageNumber",
      colourColumn: "MscColour"
    };
    const httpClientService = new sdk.HttpClientService();
    const response = await
    httpClientService.post("rest/MscKanbanService/GetStatusesForObjectByField",
    payload);
    if (response && response?.body?.GetStatusesForObjectByFieldResult) {
      request.$context.MscAllStatuses =
      response?.body?.GetStatusesForObjectByFieldResult;
    }
  }
},
```

*entityName* – list page entity schema name (Lead for example)

*statusField* – lookup (!) field of the entity to build Kanban board

*statusOrderField* – (optional) numeric field name of the status (*statusField*) lookup determining column order on the board

*colourColumn* – (optional) numeric field name of the status (*statusField*) lookup determining column color

c.

```
{
  request: "crt.HandleViewModelAttributeChangeRequest",
  handler: async (request, next) => {
    if (request.attributeName === "Items") {
      const elements = document.querySelectorAll("msc-kanban");
      elements.forEach(el => {
        el.style.minHeight = '100%';
      });
      let items = await request.$context.Items;
      //request.$context.MsclsShowMessage = items.length >= 500; this
      parameter required only when PDS.rowCount was set
      request.$context.triggerItemsChange = await
      request.$context.triggerItemsChange + 1;
    }
    await next?.handle(request);
  }
},
```

d.

```
{
  request: "msc.SaveRecordRequest",
  handler: async (request, next) => {
    if (request.$initialEvent?.detail?.taskId &&
    request.$initialEvent?.detail?.newStatusId) {
      const mask = new sdk.MaskService();
      // show the mask
      mask.showBodyMask();
      var payload = {
        entityName: "Lead",
        statusField: "QualifyStatusId",
        recordId: request.$initialEvent.detail.taskId,
        newStatus: request.$initialEvent.detail.newStatusId
      };
      const httpClientService = new sdk.HttpClientService();
      const response = await
      httpClientService.post("rest/MscKanbanService/SaveRecordById", payload);
      // hide the mask
      mask.hideBodyMask();
      if (response && response?.body?.SaveRecordByIdResult) {
        request.$context.executeRequest({
          type: "crt.NotificationRequest",
          message: response.body.SaveRecordByIdResult
        });
      }
    }
    await next?.handle(request);
  }
}
```

*entityName* – назва об'єкта

*statusField* – поле статусу по якому побудовано канбан