Delivery Knowhow

This is RunMyProcess application delivery knowhow wiki page. This website is used by RunMyProcess and RunMyProcess global partners (see Members), in order to share and leverage experiences. You can search by keyword or label to look for a problem solution, best practice, or use the internal libraries in your project. If you want to contribute, refer to Howto - Contribute a Page for details.

Information on this site is disclosed to the granted members only. Information share (copy document, sample code, figure, etc), to out of Fujitsu entities is not allowed. Click here to contact RunMyProcess if you have any special request.

Read here to start

- Howto - User this wiki
- Howto - Contribute a page
- Howto - Review and publish a page
- Confluence User Guide

Administrators

- Mani
- David Courtaigne
- Richard Manga

Create a Page

Create from template

Browse by topic

A-B

about
account
administration
api
array
authentication
backoffice
barcode
basket
browser
button

C

capi
code
coding_rule
collection
confluence
connector
css
csv
custom-list
custom-widget

D-E

dashboard
dashboards
data
delegation
delivery
design_advice
dialog
documentation
editor
email_notification
engineering-rules
event

F-I

file-list
files
freemarker
gateway
google
gsuite
history
homepage
howto
html
id
integration

Recently updated articles

Page statistics
yesterday at 4:00 PM • updated by Mani • view change

Diff tool
Aug 01, 2018 • updated by Sei FUJITA • view change

Best Practices
Jul 16, 2018 • updated by David Courtaigne • view change

Howto - Use Google APIs with personal accounts (not a service account) [under construction]
Jul 12, 2018 • updated by Enrique Castilla Contreras • view change

Howto - Design a web interface - Process
Jun 22, 2018 • updated by Franck Martin • view change

Howto - Organize JavaScript code
Jun 13, 2018 • updated by David Courtaigne • view change

Howto - Generate PDF from WI along with History widget
Jun 12, 2018 • updated by Mani • view change

Howto - Custom Home Page / Enterprise Process Store
Jun 12, 2018 • updated by Mani • view change

Howto - Test SEC + JDBC adapter + MySQL
Jun 12, 2018 • updated by Mani • view change

Howto - How to free a port already in use
Jun 12, 2018 • updated by Mani • view change

J-O

javascript
language
launch
library
list
main
manual-task
menu
mobile
modal
notification-task
organization

P-Q

paginate
pdf
performance
platform
process
project
project_copy
query

R

reg_exp
report
resources
role
runmyapp

S

sample_code
script
scripts
sec
section
security
sla
stylesheet
subprocess
support
synchronization

T-Z

tabs
task
text
thesaurus
translation
trigger
user
version
web_interface
web-interface
widget
workaround

Engineering Rules

Thesaurus

> *« Mal nommer les choses, c'est ajouter au malheur de ce monde. »*
>
> *– Albert Camus*

What is an application ?

An application is a piece of **software** designed to carry out a **particular task**.

At RunMyProcess, the terms **application** and **web interface** are synonyms.

What is a project ?

A project is an organized effort motivated to achieve **one or several applications** over a **fi xed period** and within **certain cost**.

At RunMyProcess, the term **project** is used for two radically different concepts:

- **Client project**: See definition above
- **RunMyProcess IDE project**: A set of interrelated resources

Project

Howto - Configure a project

Keep in mind that a **project** (aka RunMyProcess IDE projects) and a **client project** are two separate things. Most of the time, you'll need to create at least three projects (Main, Administration and Resources) in order to implement a client project.

Procedure - Name the project

Name - template

<Client project name> *- <Application name>* ***- <Suffix>***

Name - Rules

You shall define a project's name in accordance with the above **template**.
The client project's name shall be **capitalized**.
The client project's name shall be followed by a **white space**, an **hyphen** and a **white space**.
The application's name shall be **capitalized**.
The application's name shall be followed by a **white space**, an **hyphen** and a **white space**.
The project's name shall be written in **plain english**.
The project's name shall be **descriptive**. Avoid abbreviations and cryptic names.

Note: If needed, you may create a project named "Global - Resources" in order to share assets between several client's projects.

Note: You shall not include an "application name" in the project's name if the suffix is "Resources"

Name - Suffix

Suffix	Description
Main	The project contains the application main **processes** and **web interfaces**.
Administration	The project contains the administration **back offices**.
Resources	The project contains shared **resources**.

Name - Examples

- Human resources - Incorporate employee - Main
- Human resources - Administration
- Human resources - Resources

Procedure - Describe the project

The project shall have a **description**.
The project description shall be written in **plain english**.

Howto - Version a project

Procedure - Version a project

Version name

> The version name shall be a number formatted in **Semantic Versioning** (semver).
> The version name shall **remain the same** when switching from acceptance to live.

Version	Example	Description
MAJOR	1.3.1 2.0.0	When you make incompatible API changes
MINOR	1.3.1 1.4.0	When you add functionality in a backwards-compatible manner
PATCH	1.3.1 1.3.2	When you make backwards-compatible bug fixes

Version Description

> The version shall have a **description**.
> The version description shall be written in **plain english**.
> The version description shall contain **main reasons** for the new version.
> The version description shall contain any other **significant information**.

Howto - Configure the Resources project

Procedure - Define a collection

> You shall create **any trivial collection** into the **Resources project**.
> You shall create **any confidential collection** requiring **restricted access right** into another project.

See How to - Define a collection

Procedure - Define a connector

> You shall create **any connector** into the **Resources project**.
> You shall **never** create a connector into another project.

Procedure - Define a CAPI

> You shall create **any CAPI** into the **Resources project**.
> You shall **never** create a CAPI into another project.

Procedure - Import a stylesheet

You shall import **<u>any</u> stylesheet** as a versioned file into the **Resources project**.
You shall **<u>never</u>** import a stylesheet into another project.

See How to - Use the Delivery stylesheet

Procedure - Import a JavaScript library

You shall import **<u>any</u> JavaScript library** as a versioned file into the **Resources project**.
You shall **<u>never</u>** import a JavaScript library into another project.

Procedure - Create a project's JavaScript library (optional)

*You may create **one** versioned file containing the **project's JavaScript library** into the **Resources project**.*

See Howto - Organize JavaScript files

Procedure - Create a project's FreeMarker library (optional)

*You may create **one** versioned file containing the **project's FreeMarker library** into the **Resources project**.*

See Howto - Organize FreeMarker files

Howto - Organize FreeMarker files

Procedure - Store a FreeMarker file

You shall create **one** FreeMarker versioned file **per process** inside the project.
*You may create **one** FreeMarker versioned file containing shared functions inside the **Resources project**.*
You shall **<u>not</u>** create any other FreeMarker versioned file.

Procedure - Name a FreeMarker file (Process)

Name - Template

***<action-verb>-<business-object>*.ftl**

Name - Rules

You shall define a FreeMarker versioned file's name in accordance with the above **template**.
The **action verb** shall be the **same as** the one used for the related **process name**.
The **business object** shall be the **same as** the one used for the related **process name**.
The versioned file's name shall be in **small case kebab-case**.
The versioned file's **extension** shall be **.ftl**

Name - Examples

- build-tractor.ftl

- incorporate-employee.ftl

Procedure - Name a FreeMarker file (Resources)

Name - Template

project-library.ftl

Name - Rules

You shall define a FreeMarker versioned file's name in accordance with the above **templat e**.
The versioned file's name shall be in **small case kebab-case**.
The versioned file's **extension** shall be **.ftl**

See Howto - Configure the Resources project

Procedure - Describe a FreeMarker file

You shall include **comments** atop the FreeMarker versioned file.
The comments shall list the tasks in which the functions are used.

Howto - Organize JavaScript files

Procedure - Store a JavaScript file

Files - Rules

You shall create **all** the JavaScript versioned files listed below for **each web interface**.
*You may create **one** JavaScript versioned file containing shared functions inside the **Reso urces project**.*
You shall **not** create any other JavaScript versioned file.

Files - Summary

File	Suffix	Description
Initialize	initialize.js	Contains functions executed when a **screen is initialized**.
Action	action.js	Contains functions executed when an **action occurs** on the web interface.
Listen	listen.js	Contains functions executed when a **variable changes**.

See Howto - Organize JavaScript code

Procedure - Name a JavaScript file (Web interface)

Name - Template

***<action-verb>-<business-object>-<suffix>*.js**

Name - Rules

You shall define a JavaScript versioned file's name in accordance with the above **template**.
The **action verb** shall be the **same as** the one used for the related **web interface's name**.
The **business object** shall be the **same as** the one used for the related **web interface's name**.
The versioned file's **suffix** shall be one of those listed if the **array above**.
The versioned file's name shall be in **small case kebab-case**.
The versioned file's **extension** shall be **.js**

Name - Examples

- incorporate-employee-initialize.js
- incorporate-employee-action.js
- incorporate-employee-listen.js

Procedure - Name a JavaScript file (Resources)

Name - Template

project-library.js

Name - Rules

You shall define a JavaScript versioned file's name in accordance with the above **template**.
The versioned file's name shall be in **small case kebab-case**.
The versioned file's **extension** shall be **.js**

See Howto - Configure the Resources project

Web interface -

Howto - Design a web interface - Process

Procedure - Name the web interface

Name - Template

<Action verb> <business object>

Name - Rules

You shall define a web interface's name in accordance with the above **template**.
The action verb shall be **capitalized** and followed by a **white space**.
The business object shall be in **small case**.
The web interface's name shall be written in **plain english**.
The web interface's name shall be **descriptive**. Avoid abbreviations and cryptic names.

Name - Examples

- Solve problem
- Hire employee

Name - Remarks

The web interface’s name is used as the **application name**:

- Displayed on the **homepage** of the RunMyProcess portal. If the name doesn't suit the client's needs, you shall use a customized home page.
- Used as the **meta description** html tag displayed in the web browser's tab (on the launch screen only, once this screen has been submitted the web interface dynamic’s name will be used as the tab’s title in the browser).

Procedure - Describe the web interface

Description - Rules

The web interface shall have a **description**.
The web interface's description shall be written in **plain english**.

Procedure - Configure the web interface's header

There are no rules regarding the web interface's header configuration.

Procedure - Configure the web interface's dynamic name

Dynamic name - Template

${process.instance.id}

Dynamic name - Rules

You shall define a web interface's dynamic name in accordance with the above **template**.

See How to - Configure the launch event

Dynamic name - Remarks

The **process.instance.id variable** is defined as an output variable of the **launch event** (See How to - Configure the launch event)
On the **launch screen**, the process.instance.id variable is defined as the **default value** of the related **text widget** (see procedure below).

Dynamic name - Examples

- PB000078
- HRE000420

Procedure - Design the web interface

Add a text widget - Manual task ID

Rules

You shall create a text widget **at the bottom** of the web interface.
The text widget's **properties and rules** shall be configured in accordance with **configuration array** below.
The text widget's configuration shall remain the **same on every screen**.
The text widget's **default value** shall be defined as specified below.

Configuration

Properties			Rules			
Label	**Variable**	**Identifier**	**Available**	**Active**	**Visible**	**Required**
Manual Task ID	process.manual_task.id	id_process_manual_task_id	true	false	false	true

See How to - Configure a text widget

Default value

You shall initialize the **process.manual_task.id** variable by setting the **default value** of the related **text widget**.
The default value will be the **identifier** of the **start event** which is considered as a manual task.
The default value shall fulfill the requirements defined in the **Input variable - process.manual_task.id** chapter of Howto - Design a manual task.

Procedure - Name a screen

Name - Template

<Action verb> <business object>

Name - Rules

You shall define a screen's name in accordance with the above **template**.
The screen's name shall be the **same** as the name of the **related manual task**.
*If the screen is used for **several manual task**, you may **omit** the optional "**additional information**" in order to generalize the name.*
The action verb shall be **capitalized** and followed by a **white space**.
The business object shall be in **small case**.
The screen's name shall be written in **plain english**.
The screen's name shall be **descriptive**. Avoid abbreviations and cryptic names.

See Howto - Design a manual task

Name - Examples

- Review contract
- Specify hiring date

Howto - Design a web interface - Report

Overview

We call "**Report**" a **web interface** (defined in this page) containing several **report widgets**, each of them being based of a **web interface report**.

Procedure - Name the web interface

Name - Template

Report - <*Action verb*> <*business object*>

Name - Rules

You shall define a web interface's name in accordance with the above **template**.
The web interface's name shall have the prefix "**Report**"
The prefix shall be followed by a **white space**, an **hyphen** and a **white space**.
The action verb shall be **capitalized** and followed by a **white space**.
The business object shall be in **small case**.
The web interface's name shall be written in **plain english**.
The web interface's name shall be **descriptive**. Avoid abbreviations and cryptic names.

Name - Examples

- Report - Solve problem
- Report - Hire employee

Procedure - Describe the web interface

Description - Rules

The web interface shall have a **description**.
The web interface's description shall be written in **plain english**.

Procedure - Configure the web interface's header

There are no rules regarding the web interface's header configuration.

Procedure - Design the web interface

Add a tabs widget

You shall integrate a **tabs widget** in order **to organize** the **report widgets** within the web interface.

See How to - Configure a tabs widget

Add the report widgets

You shall integrate **one** and only one **report widget** in each **tab** of the tabs widget.

See Howto - Configure a report widget - Web interface

Howto - Design a menu

Procedure - Store the menu

The menu shall be stored in the **Resources project.**
The menu shall be created as a **Custom widget**.

Procedure - Name the menu

Name - Template

Menu - *<Project name>*

Name - Rules

You shall define a menu's name in accordance with the above **template**.
The menu's name shall have the prefix "**Menu**"
The prefix shall be followed by a **white space**, an **hyphen** and a **white space**.
The project name shall be **capitalized**.
The menu's name shall be written in **plain english**.
The menu's name shall be **descriptive**. Avoid abbreviations and cryptic names.

Name - Examples

- Menu - Human resources
- Menu - Human resources - Administration

Procedure - Describe the menu

The menu shall have a **description**.
The menu's description shall be written in **plain english**.

Procedure - Configure the menu

Configuration - JSON array

The menu shall structured as a **well-formed JSON array**.
The JSON array shall be validated using **JSONLint**.
The JSON array shall be **indented using tabs**.

Configuration - Access rights

Access rights shall be defined using the **has_right()** freemarker function.
Access rights shall be described in a **"_comment_" attribute** above the "visible" attribute of the menu item.
Access rights description shall include **roles names and identifiers**.

Howto - Configure a widget's rules

Procedure - Define a widget's "Active" condition

Active condition - Rules

You shall **never** use the **"Edit active condition"** modal window to conditionally define the "Active" parameter.
You shall code the conditional logic in the **appropriate JavaScript versioned file**.

See Howto - Organize JavaScript files

Active condition - Code sample

```
if (my_variable === 'my_condition') {
 window['id_of_my_widget'].setActive(true)
}
```

Documentation: Javascript widgets API reference for RMP_TextInput

Procedure - Define a widget's "Visible" condition

Visible condition - Rules

You shall **never** use the **"Edit visible condition"** modal window to conditionally define the "Visible" parameter.
You shall code the conditional logic in the **appropriate JavaScript versioned file**.

See Howto - Organize JavaScript files

Visible condition - Code sample

```
if (my_variable === 'my_condition') {
 window['id_of_my_widget'].setVisible(true)
}
```

Documentation: Javascript widgets API reference for RMP_TextInput

Procedure - Define a widget's "Required" condition

Required condition - Rules

You shall **never** use the **"Edit required condition"** modal window to conditionally define the "Required" parameter.
You shall code the conditional logic in the **appropriate JavaScript versioned file**.

See Howto - Organize JavaScript files

Required condition - Code sample

```
if (my_variable === 'my_condition') {
 window['id_of_my_widget'].setRequired(true)
}
```

Documentation: Javascript widgets API reference for RMP_TextInput

Howto - Configure a tabs widget

Procedure - Define a tabs widget's identifier (ID)

Identifier - Template

id_tabs_<*container_name*>

Identifier - Rules

You shall define a tabs widget's identifier in accordance with the above **template**.
The tabs widget's identifier shall be written in **english**.
The tabs widget's identifier shall be written in **lower case snake_case**.
The tabs widget's identifier shall be **descriptive**. Avoid abbreviations and cryptic names.

Identifier - Examples

- id_tabs_problem_definition
- id_tabs_create_tractor_report

Procedure - Define a tab "Active" condition

Active condition - Rules

You shall **never use the "Script edition"** modal window (accessible from the tabs array) to conditionally define the "Active" parameter of a tab.
You shall code the conditional logic in the **appropriate JavaScript versioned file**.

See Howto - Organize JavaScript files

Active condition - Code sample

```
if (my_variable === 'my_condition') {
 window['id_of_my_tabs_widget'].setTabActive('0
', true)
}
```

Documentation: Javascript widgets API reference for RMP_TabPanel

Procedure - Define a tab "Visible" condition

Visible condition - Rules

You shall **never** **use the "Script edition"** modal window (accessible from the tabs array) to conditionally define the "Visible" parameter of a tab.
You shall code the conditional logic in the **appropriate JavaScript versioned file**.

See Howto - Organize JavaScript files

Visible condition - Code sample

```
if (my_variable === 'my_condition') {
 window['id_of_my_tabs_widget'].setTabVisible('
0', true)
}
```

Documentation: Javascript widgets API reference for RMP_TabPanel

Procedure - Trigger a script when a tab is clicked

Script - Rules

You **shall** **never** **write JavaScript code** directly in the Script edition modal window.
You shall **call a JavaScript function** stored in the appropriate **action.js versioned file**.

See Howto - Organize JavaScript files

Script - Code sample

```
PROJECT.tabs.onClick('id_tabs_container_name',
'id_tab')
```

Note: This is just an example. You have to write your own function in the appropriate versioned file.

Howto - Configure a section widget

Procedure - Define a section widget's identifier (ID)

Identifier - Template

id_section_<*container_name*>

Identifier - Rules

You shall define a section widget's identifier in accordance with the above **template**.
The section widget's identifier shall be written in **english**.
The section widget's identifier shall be written in **lower case snake_case**.
The section widget's identifier shall be **descriptive**. Avoid abbreviations and cryptic names.

Identifier - Examples

- id_section_problem_definition
- id_section_create_tractor_report

Procedure - Trigger a script when a section is opened or closed

Script - Rules

You **shall never write JavaScript code** directly in the Script edition modal window.
You shall **call a JavaScript function** stored in the appropriate **action.js versioned file**.

See Howto - Organize JavaScript files

Script - Code sample

```
PROJECT.section.onClick('id_section_container_name')
```

Note: This is just an example. You have to write your own function in the appropriate versioned file.

Howto - Configure a text widget

Procedure - Define a text widget's variable

Variable - Philosophy

Variables are defined using the **object notation**. By doing so we avoid scattered variables and ensure they are properly organized, easily exportable as a JSON. Variables are the **attributes** of the **business object**.

Variable - Template

<business_object>.<object_attribute>*.<object_attribute>*

Variable - Rules

You shall define a text widget's variable in accordance with the above **template**.
The text widget's variable shall use the **object notation**: attributes are separated by **dots**.
The object's attributes shall **not** belong to the **reserved keywords** listed below.
The business object and its attributes shall be written in **english**.
The business object and its attributes shall be written in **lower case snake_case**.
The business object and its attributes shall be **descriptive**. Avoid abbreviations and cryptic names.

Variable - Reserved keywords

The object's attributes shall not belong to the reserved keywords listed below:

> action, status, value, label, history, section, tabs

Variable - Examples

- recruitment.candidate.first_name
- recruitment.hiring_date
- problem.description

Procedure - Define a text widget identifier (ID)

Identifier - Template

> **id_*<business_object>*_*<object_attribute>***_*<object_attribute>*

Identifier - Rules

> You shall define a text widget's identifier in accordance with the above **template**.
> The text widget's identifier shall strictly **map the text widget's variable**.
> The text widget's identifier shall be written in **lower case snake_case**, dots shall be replaced by **underscores**.

Identifier - Examples

- id_recruitment_candidate_first_name
- id_recruitment_hiring_date
- id_problem_desciption

Howto - Configure a button widget - Process update

Procedure - Define a button widget's variable

Variable - Template

> ***<business_object>*.status**

Variable - Rules

> You shall define a button widget's variable in accordance with the above **template**.
> The button widget's variable shall use the **object notation**: attributes are separated by **dot s**.
> The business object's attribute shall always be "**status**".
> The business object shall be written in **english**.
> The business object shall be written in **lower case snake_case**.
> The business object shall be **descriptive**. Avoid abbreviations and cryptic names.

Variable - Examples

- problem.status
- recruitment.status

sample

Procedure - Define a button widget's value

Value - Template

<action_verb_preterit>

Value - Rules

You shall define a button widget's value in accordance with the above **template**.
The button widget's value shall be the **preterit tense** of an **action verb**.
The action verb shall be written in **english**.
The action verb shall be written in **lower case snake_case**.
The action verb shall be **descriptive**. Avoid abbreviations and cryptic names.

Value - Examples

- escalated
- validated
- canceled

Procedure - Define a button widget's identifier (ID)

Identifier - Template

id_*<business_object>*_status_*<action_verb_preterit>*

Identifier - Rules

You shall define a button widget's identifier in accordance with the above **template**.
The button widget's identifier shall strictly **map the button widget's variable and value**.
The button widget's identifier shall be written in **lower case snake_case**, dots shall be replaced by **underscores**.

Identifier - Examples

- id_recruitment_status_canceled
- id_problem_status_escalated

Procedure - Execute a pre-launch script

Script - Rules

You **shall <u>never</u> write JavaScript code** directly in the Script edition modal window.
You shall **call a JavaScript function** stored in the web interface's **action.js versioned file**.

See Howto - Organize JavaScript files

Script - Code sample

```
PROJECT.button.onClick('id_button')
```

Note: This is just an example. You have to write your own function in the appropriate versioned file.

Howto - Configure a button widget - JavaScript

Procedure - Define a button widget's identifier (ID)

Identifier - Template

id_*<action_verb>*_*<object>*

Identifier - Rules

You shall define a button widget's identifier in accordance with the above **template**.
The button widget's identifier shall be written in **english**.
The button widget's identifier shall be written in **lower case snake_case**.
The button widget's identifier shall be **descriptive**. Avoid abbreviations and cryptic names.

Identifier - Examples

- id_open_modal_window
- id_display_barcode

Procedure - Execute a script

Script - Rules

You **shall <u>never</u> write JavaScript code** directly in the Script edition modal window.
You shall **call a JavaScript function** stored in the web interface's **action.js versioned file**.

See Howto - Organize JavaScript files

Script - Code sample

```
PROJECT.button.onClick('id_button')
```

Note: This is just an example. You have to write your own function in the appropriate versioned file.

Howto - Configure a HTML widget

Procedure - Define the HTML widget's identifier (ID)

Identifier - Template

id_html_*<container_name>*

Identifier - Rules

You shall define a HTML widget's identifier in accordance with the above **template**.
The HTML widget's identifier shall be written in **english**.
The HTML widget's identifier shall be written in **lower case snake_case**.
The HTML widget's identifier shall be **descriptive**. Avoid abbreviations and cryptic names.

Identifier - Examples

- id_html_page_header
- id_html_modal_window_action_buttons

Procedure - Describe the HTML widget

Description - Rules

You shall include **HTML comments** atop the HTML widget's content.
The comments **first line** shall remind the **identifier** of the HTML widget.
The comments **second line** shall clearly describe the **purpose** of the HTML widget.
The comments second line shall be written in **plain english**.

Description - Code sample

```
<!-- id_html_container_name -->
<!-- Detailed explanations about the purpose of
my HTML widget -->
```

Procedure - Develop the HTML widget's content

Content - Style

You shall **never use the <style> tag** within an HTML widget.
You shall use a **stylesheet versioned file** in order to style an HTML widget's content.

See Howto - Create a custom stylesheet

Content - Logic

You shall **never code logic** directly within an HTML widget.
You shall use a **JavaScript versioned file** in order to implement application logic and display the result in the HTML widget.

See Howto - Organize JavaScript files

Howto - Configure a report widget - Web interface

Overview

We call "**Report**" a **web interface** containing several **report widgets** (defined in this page), each of them being based of a **web interface report**.

Procedure - Define a report widget's identifier (ID)

Identifier - Template

id_report_<action_verb>_<business_object>_<status>_<subset>

Identifier - Rules

You shall define a report widget's identifier in accordance with the above **template**.
The report widget's identifier shall start with the prefix **id_report**.
The report widget's identifier shall **match the name** of the associated **web interface report** (action verb, business object, status, subset)
The report widget's identifier shall be written in **english**.
The report widget's identifier shall be written in **lower case snake_case**.
The report widget's identifier shall be **descriptive**. Avoid abbreviations and cryptic names.

See Howto - Configure a report - Web interface

Identifier - Status & subset

See Howto - Configure a report - Web interface

Identifier - Examples

- id_report_solve_problem_solved
- id_report_solve_problem_solved_self
- id_report_incorporate_employee_in_progress
- id_report_release_employee_all_headquarters

Procedure - Associate a web interface report

You shall associate a **web interface report** which **name matches** the report **widget identifier**.

See Howto - Configure a report - Web interface

Procedure - Prohibit a report widget edition

You shall set the **Active checkbox to false** in order to prevent report editing by the users.

Procedure - Execute a post loaded script

Script - Rules

You **shall <u>not</u> write JavaScript code** directly in the Script edition modal window.
You shall **call a JavaScript function** stored in the web interface's **action.js versioned file**.

See Howto - Organize JavaScript files

Script - Code sample

```
PROJECT.report.onLoad('id_report')
```

Note: This is just an example. You have to write your own function in the appropriate versioned file.

Howto - Configure a report - Web interface

Overview

We call "**Report**" a **web interface** containing several **report widgets**, each of them being based of a **web interface report** (defined in this page).

Procedure - Define a web interface report's name

Name - Template

<Action verb> <business object> - <Status> - <Subset>

Name - Rules

You shall define a report widget's identifier in accordance with the above **template**.
The **action verb** and **business object** shall be the **same** as in the related **web interface**'s name.
The **action verb** shall be **capitalized** and followed by a **white space**.
The **business object** shall be in **small case** and followed by a **white space**, an **hyphen** and a **white space**.
The **status** shall be **capitalized**.
The **subset** is **optional** and shall be **capitalized** and preceded by a **white space**, an **hyphen** and a **white space**.
The report widget's identifier shall be written in **plain english**.
The report widget's identifier shall be **descriptive**. Avoid abbreviations and cryptic names.

Name - Status

The report is filtered according to the value of the **<business_object>.status** variable.

Status	Filter	Description
all	none	All status.
in_progress	status != "completed"	All status except "completed".
<status>	status = *<status>*	A possible value of the **<business_object>.status** variable.

See Howto - Configure a button widget - Process update

Name - Subset

The report is filtered according to the value of one or several process measures.

Subset	Description
self	Only the instances started by the current user ("My Tasks" report).
<custom>	An arbitrary subset.

Name - Examples

- Solve problem - Solved
- Solve problem - Solved - Self
- Incorporate employee - In progress
- Release employee - All - Headquarters

Procedure - Set the report mode

You shall set the web interface report's **mode** to **Automatic**.

Howto - Configure an history widget

Procedure - Define an history widget's identifier (ID)

Identifier - Template

id_history

Identifier - Rules

You shall define an history widget's identifier in accordance with the above **template**.

Procedure - Add an historized variable

Rules

You **shall <u>never</u> write JavaScript code** directly in the Script edition modal window.
You shall **call a JavaScript function** stored in the appropriate **action.js versioned file**.

Code samples

Display an historized variable as is

```
[[historized_variable_name]]
```

Use a JavaScript function

```
PROJECT.history.displayVariable('historized_variable_name', [[historized_variable_name]])
```

Note: This is just an example. You have to write your own function in the appropriate versioned file.

Howto - Avoid the JavaScript widget

Procedure - Avoid the JavaScript widget

You shall **never** use the **JavaScript widget**.
You shall store all your JavaScript code in **versioned files**.

See Howto - Organize JavaScript files
See Howto - Organize JavaScript code

Note: This rule may be bypassed in very specific cases. You shall first ask the Delivery Manager for approval.

Master data

Howto - Define a collection

Procedure - Store the collection

You shall store any **trivial collection** into the **Resources project**.
You shall store any **confidential collection** requiring **restricted access right** into another project.

See Howto - Configure the Resources project

Procedure - Name the collection

Name - Template

col_<*business_object*>

Name - Rules

You shall define a collection's name in accordance with the above **template**.
The collection's name shall be written in the **singular**.
The business object shall **not** belong to the **reserved keywords** listed below.
The collection's name shall be written in **english**.
The collection's name shall be written in **lower case snake_case**.
The collection's name shall be **descriptive**. Avoid abbreviations and cryptic names.

Name - Reserved keywords

The business object shall not belong to the reserved keywords listed below:

manual_task, notification_task, task, notification, process, status, action, alert, list, action

Name - Examples

- col_contract_type
- col_checksheet

Howto - Define a custom list

Procedure - Store the custom list

The custom list shall be stored in the **Resources project**.

See Howto - Configure the Resources project

Procedure - Name the custom list

Name - Template

list_<business object>

Name - Rules

You shall define a custom list's name in accordance with the above **template**.
The custom list's name shall be written in the **singular**.
The custom list's name shall be written in **english**.
The custom list's name shall be written in **lower case snake_case**.
The custom list's name shall be **descriptive**. Avoid abbreviations and cryptic names.

Name - Examples

- list_task
- list_contract_type
- list_checksheet

Procedure - Define custom list items

The item's value shall be written in **english**.
The item's value shall be written in **lower case snake_case**.
The item's value name shall be **descriptive**. Avoid abbreviations and cryptic names.

There's no rule regarding item's label.

Process

Howto - Design a process

Procedure - Name the process

Name - Template

<process_number> - <Action verb> <business object>

Name - Rules

You shall define a process name in accordance with the above **template**.
The process number shall be followed by a **white space** and a **hyphen**.
The action verb shall be **capitalized** and followed by a **white space**.
The business object shall be in **small case**.
The process name shall be written in **plain english**.
The process name shall **describe the main purpose** of the process, the work that has to be performed. Avoid abbreviations and cryptic names.

Name - Process number

The process number shall be an **integer**.
The **main** process shall have the **number 1**.

Name - Examples

- 1 - Build tractor
- 2 - Solve problem
- 1 - Incorporate employee

Procedure - Describe the process

The process shall have a **description**.
The process description shall be written in **plain english**.

Procedure - Set the process dynamic label

Dynamic label - Template

${process.instance.id}

Dynamic label - Rules

You shall define a process dynamic label in accordance with the above **template**.

Dynamic label - Remarks

The **process.instance.id variable** is defined as an output variable of the **launch event** (See How to - Configure the launch event).
On the **launch screen**, the process.instance.id variable is defined as the **default value** of the related **text widget** (see Howto - Design a web interface - Process).

Dynamic label - Examples

- PB000078
- HRE000425

Procedure - Polish the process layout

Polish - Rules

The process flow shall be oriented from **left to right**. and from **top to bottom**.
The connectors (lines) shall be **short** and shall **not overlap**.
The tasks shall be horizontally and vertically **aligned**.
The tasks shall be **evenly spaced**.

Polish - Explanations

- All of these layout rules don't change the content of the process but they make it **much easier to read and understand**.
- A lack of consistent direction of flow (slalom) makes it very difficult to differentiate the **main and alternative scenarios**.
- It also indicates that the designer probably captured all the details at once, while a best practice is to **address the primary ‘happy day’ scenario first** and address alternative scenarios only afterwards.

Procedure - Sunder large processes

Sunder process - Rules

The process shall contain **less than 10 activities**.
You shall use several **layers of detail** (subprocesses).

See How to - Design a subprocess

Sunder process - Explanations

- The “everything in one page” style impairs overview and makes it difficult to **understand the process**.
- Large processes make it very difficult to analyze them and **spot the issues** – the value of visualization is lost.
- You can decompose a very large (and typically inconsistent) business process into a simple and consistent process structured in several levels of detail.
- A business process with three levels of detail following the rule of up to 10 activities in one level may contain 10^3 = 1000 tasks.

Procedure - Sunder activities

An activity shall perform one and only **one action**.

Sunder activity - Example

In the first case, if the second or third task fails, you can reboot the process without risking to add the item to collection A a second time (there creating a duplicate in the collection). Designing the same process using a single task is technically possible but it does not allow a safe reboot of the process.

Sunder activity - Explanation

When designing a process, **keep in mind that process instances might have to be manually relaunched (or rebooted)** and therefore separate the tasks in a way that will allow the process to be relaunched without causing further problems.

Keep also in mind that **when a process is rebooted on a task which status has been set on “Waiting” only the output variables will be recalculated**.

For example, if you have several actions to perform on a collection (or several collections), it is better to divide them in several tasks (boxes). This way if one of the action performed fails, you can reboot the process without risking to perform the same action twice.

Howto - Design a subprocess

Procedure - Name the subprocess

Name - Template

<subprocess_number> - <Action verb> <business object>

Name - Rules

You shall define a subprocess name in accordance with the above **template**.
The subprocess number shall be followed by a **white space** and a **hyphen**.
The action verb shall be **capitalized** and followed by a **white space**.
The business object shall be in **small case**.
The subprocess name shall be written in **plain english**.
The subprocess name shall **describe the task** that has to be performed. Avoid abbreviations and cryptic names.

Name - Subprocess number

Rules

The subprocess number shall be a **hierarchical number**.
The subprocess number shall be composed of **integers** separated by **dots**.
There shall not be more than **three levels** of numbering.

Examples

Number	Level	Description
1	1	Main process
2	1	Another process
1.1	2	A subprocess of the main process
1.4.2	3	A subprocess of a subprocess of the main process

Procedure - Describe the subprocess

The sub process shall have a **description**.
The sub process description shall be written in **plain english**.

Procedure - Include the subprocess

The **label of the collapsed subprocess** in the parent process shall be the **same** as the **na**

me of the subprocess.

This rule ensures a visible relation between the parent process and the subprocess.

Howto - Configure the launch event

Procedure - Define launch event's output variables

Output variables - Rules

You shall create the launch event's **output variables** as specified in the **summary array** b elow.

Output variables - Summary

Rank	Variable	Value	Description
1	**freemarker**	*<file_id>*	Identifier of the **versione d file** containing the freemarker functions.
2	**process.instance.id**	See procedure	Identifier of the **current instance** of the process.

Procedure - Define the "freemarker" output variable

Rules

The **main process** launch event shall define an output variable called "**freemarker**".
This variable contains the **ID of the versioned file** storing the freemarker functions used for this process.
This variable shall also contain the **name of the versioned file** as a freemarker **comment**.

Example

```
<#-- name_of_my_freemarker_file.ftl -->
18277e40-26ce-11e8-9c27-066d75fba2ef
```

Purpose

Then when you want to call a freemarker function, just type:

```
<#include freemarker>${run_my_function()}
```

Procedure - Define the "process.instance.id" output variable

Process instance - Rules

The main process **launch event** shall define an output variable called "**process.instance.id**".
This variable will contain the **computed unique identifier** of the current process instance.
This **variable's value** is set as a **template of the identifier** which shall fulfill the **client's requirements**.

Process instance - Template example 1

Template

<*prefix*>${next_value("<*sequence_variable*>")?number?string("000000")}

Output

- HRE000240
- PB000078

Process instance - Template example 2

Template

***<prefix>-${now("yyyy")}*-${next_value("<*sequence_variable*>"+now("yyyy"))?number?string("0000")}**

Output

- SYS-2018-0538
- CHOMP-2017-0045

Process instance - Prefix

The prefix can be freely defined in accordance with the client's wishes.

Process instance - Sequence variable

Sequence variable - Template

***<action_verb>*_*<business_object>*_sequence**

Sequence variable - Rules

You shall define a sequence variable in accordance with the above **template**.
The **action verb** shall be the **same as** the one used for the related **process name**.
The **business object** shall be the **same as** the one used for the related **process name**.
The sequence variable shall have the suffix "**sequence**".
The sequence variable shall be written in **english**.
The sequence variable shall be written in **lower case snake_case**.

Sequence variable - Examples

- build_tractor_sequence
- review_problem_sequence

Howto - Design a manual task

Procedure - Name the manual task

Name - Template

<Action verb> <business object> *(<additional information>)*

Name - Rules

You shall define a manual task's name in accordance with the above **template**.
The action verb shall be **capitalized** and followed by a **white space**.
The business object shall be in **small case**.
The additional information is not mandatory.
The additional information shall be written **in parentheses** and preceded by a **white space**.
The manual task's name shall be written in **plain english**.
The manual task's name shall be **descriptive**. Avoid abbreviations and cryptic names.

Name - Examples

- Review problem
- Express requirements (manager)
- Update check-sheet
- Specify hiring date

Procedure - Describe the manual task

Description - Rules

The manual task shall have a **description**.
The manual task's description shall be written in **plain english**.

Procedure - Define the manual task's input variables

Input variables - Rules

You shall create the manual task's **input variables** as specified in the **summary array** below.

Input variables - Summary

Rank	Variable	Value	Description
1	**process.manual_task.id**	*See procedure below*	Unique identifier of the manual task.

Input variable - process.manual_task.id

Template

> ***<process_name>_<action_verb>_<business_object>***

Rules

> The process.manual_task.id input variable contains the **identifier of the manual task**.
> You shall define the **process.manual_task.id value** in accordance with the above **templat e**.
> The process.manual_task.id value shall be **unique** across application's processes and subprocesses.
> The process.manual_task.id value shall be written in **english**.
> The process.manual_task.id value shall be written in **lower case snake_case**.
> The process.manual_task.id value shall be **descriptive**. Avoid abbreviations and cryptic names.

Examples

- incorporate_employee_confirm_hiring_date
- solve_problem_review_problem

Howto - Design a notification task

Procedure - Name the notification task

Name - Template

> **Notify *<user group>***

Name - Rules

> You shall define a notification task's name in accordance with the above **template**.
> The action verb shall always be "**Notify**".
> The action verb shall be **capitalized** and followed by a **white space**.
> The user group shall be in **small case**.
> The notification task's name shall be written in **plain english**.
> The notification task's name shall be **descriptive**. Avoid abbreviations and cryptic names.

Name - User group

> If the user group is a **generic term** (*e.g. stakeholders, production team*) it shall be defined it in the notification task's **description**.
> You may use the "**or**" (**|**) or "**and**" (**&**) **logical operators** to define the user group.

Name - Examples

- Notify stakeholders
- Notify manager & quality controller
- Notify network or headquarters

Procedure - Describe the notification task

Description - Rules

> The notification task shall have a **description**.
> The notification task's description shall be written in **plain english**.
> The notification task's description shall fully describe the **user group** if it is a generic term.

Procedure - Define the notification task's input variables

Input variables - Rules

> You shall create the notification task's **input variables** as specified in the **summary array** below.

Input variables - Summary

Rank	Variable	Value	Description
1	**process.notification_task.id**	*See procedure below*	Unique identifier of the notification task.

Input variable - process.notification_task.id

Template

> ***<process_name>_notify_<user_group>***

Rules

> The process.notification_task.id input variable contains the **identifier of the manual task**.
> You shall define the **process.notification_task.id value** in accordance with the above **template**.
> The process.notification_task.id value shall be **unique** across application's processes and subprocesses.
> The process.notification_task.id value shall be written in **english**.
> The process.notification_task.id value shall be written in **lower case snake_case**.
> The process.notification_task.id value shall be **descriptive**. Avoid abbreviations and cryptic names.

Examples

- incorporate_employee_notify_network_or_headquarters
- solve_problem_notify_quality_controller

Howto - Design an exclusive gateway

Procedure - Name the gateway

Name - Template

> ***<Question asked>?***

Name - Rules

You shall define an exclusive gateway's name in accordance with the above **template**.
The exclusive gateway's name shall be a **question**.
The exclusive gateway's name shall end with a **question mark**.
The exclusive gateway's name shall be written in **plain english**.
The exclusive gateway's name shall be **descriptive**. Avoid abbreviations and cryptic names.

Name - Examples

- Problem status?
- Company branch?
- Requirements approved?

Procedure - Describe the gateway

The gateway shall have a **description**.
The gateway description shall be written in **plain english**.

Procedure - Name the outgoing sequence flows

You shall define a **name** for **each** outgoing **sequence flow**.
The **name** shall be a possible **answer** to the **question** asked at the gateway.
The name shall be written in **english**.
The name shall be written in **lower case snake_case**.
The name shall be **descriptive**. Avoid abbreviations and cryptic names.

Procedure - Define the gateway conditions

The gateway shall always have a **default condition**.

Script

Howto - Organize JavaScript code

Overview

In RunMyProcess, JavaScript can be used in many different locations on client side. It is therefore very important to keep the code as clean, efficient and centralized as possible in order to ensure its sustainability.

Procedure - Store JavaScript code

JavaScript widget

You shall **never** use the **JavaScript widget**.

See Howto - Avoid the JavaScript widget

Other widgets

> You **shall <u>never</u> write JavaScript code** directly in the Script edition modal window.
> You shall **call a JavaScript function** stored in the appropriate **versioned file**.

Widget	JavaScript Usage
All	Visible and active rules
All input	Validation and required rules
Report	Post loading script and calculated columns
Section	Open and close sections events
Tab	Each tab
List	Post loading script
Checkbox	Post loading script
Radio buttons	Post loading script
Array	Add or delete rows events
Custom	Conditional loading script
Tree	Data and nodes events
Progress bar	Custom configuration

API listener callback

> You **shall <u>never</u> write JavaScript code** directly in the Script edition modal window.
> You shall **call a JavaScript function** stored in the appropriate **action.js versioned file**.

JavaScript files

> You shall store all the JavaScript code in **versioned files**.

See Howto - Organize JavaScript files

Procedure - Define the "Initialize" versioned file

Initialize - Code sample

Javascript versioned file - Initialize

```
(function (PROJECT, undefined) {

 "use strict"

 PROJECT.screen = {}

 // Public method
 // Executed only once on every screen (API
```

```
listener callback)
 PROJECT.screen.initialize = function () {

  // Get the identifier of the current task
  var task_id =
RMPApplication.getVariable('process.manual_task
.id')

  // Execute a function depending of the
current task
  switch (task_id) {

   case 'my_first_task':
    // Listen to variables changes in the web
interface

RMPApplication.addListener(PROJECT.listener.thi
sScreen)
    // Execute some functions
    runMyFunction()
   break

   case 'my_second_task':
   case 'my_third_task':

RMPApplication.addListener(PROJECT.listener.ano
therScreen)
    runAnotherFunction()
   break
  }
 }

 // Private method
 function runMyFunction () {
  /* Do something... */
 }

 // Private method
 function runAnotherFunction () {
  /* Do something... */
```

```
 }

})( window.PROJECT = window.PROJECT || {} );
```

See Howto - Design a manual task and Howto - Design a web interface - Process for information about the **process.manual_task.id** variable.

Procedure - Define the "Action" versioned file

Action - Code sample

Javascript versioned file - Action

```
(function (PROJECT, undefined) {

 "use strict"

 PROJECT.button = {}
 PROJECT.history = {}
 PROJECT.report = {}
 PROJECT.section = {}
 PROJECT.tabs = {}

 // Public method
 // Execute a function depending of the clicked
button's identifier
 PROJECT.button.onClick = function (id_button)
{
  switch (id_button) {
   case 'id_my_first_button':
    runMyFunction()
   break
   case 'id_my_second_button':
   case 'id_my_third_button':
    runAnotherFunction()
   break
  }
 }

 // Public method
 // Execute a function depending of the loaded
report's identifier
 PROJECT.report.onLoad = function (id_report) {
  switch (id_report) {
   case 'id_my_first_report':
    runMyFunction()
   break
   case 'id_my_second_report':
```

```
    case 'id_my_third_report':
     runAnotherFunction()
    break
   }
  }

  // Private method
  function runMyFunction () {
   /* Do something... */
  }

  // Private method
  function runAnotherFunction () {
   /* Do something... */
```

```
 }

})( window.PROJECT = window.PROJECT || {} );
```

Procedure - Define the "Listen" versioned file

Listen - Code sample

Javascript versioned file - Listen

```
(function (PROJECT, undefined) {

 "use strict"

 PROJECT.listener = {}

    // declare a public method to be called at
each variable change. This method is called in
the initialize.js file.
 PROJECT.listener.thisScreen =
function(variable, value, index, widget) {
        switch (variable) {
   case 'my_first_variable':
    runMyFunction()
   break
   case 'my_second_variable':
   case 'my_third_variable':
    runAnotherFunction()
   break
  }
 }

 // Execute a public method to be called at
each variable change. This method is called in
the initialize.js file.
 PROJECT.listener.anotherScreen =
function(variable, value, index, widget) {
        switch (variable) {
   case 'my_first_variable':
    runMyFunction()
   break
   case 'my_third_variable':
    runMyFunction()
    runAnotherFunction()
   break
  }
 }

  // Private method
 function runMyFunction () {
  /* Do something... */
 }

 // Private method
 function runAnotherFunction () {
  /* Do something... */
 }

})( window.PROJECT = window.PROJECT || {} );
```

Documentation: Javascript widgets API reference - RMPApplication

Best Practices

Best Practices / Workarounds

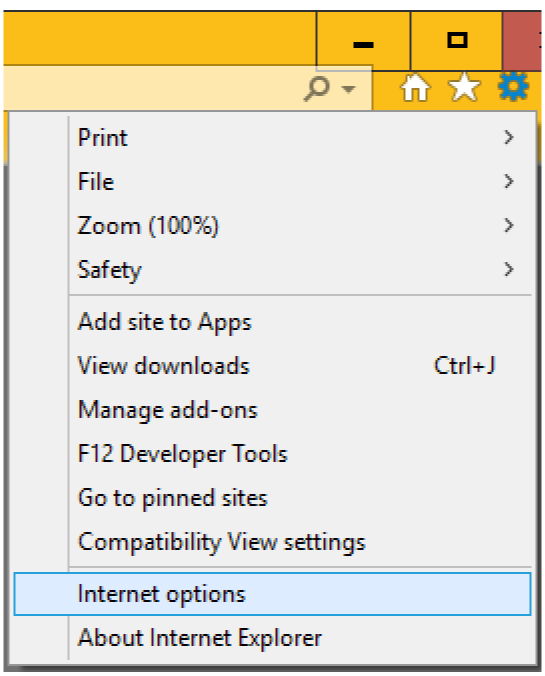
Client / Support

Howto - Change IE browser setting to open link in new tab

Publish date	26 Feb 2018
Contributor	Hugo David E
Reviewer	Reviewer Meeting
Status	PUBLISHE

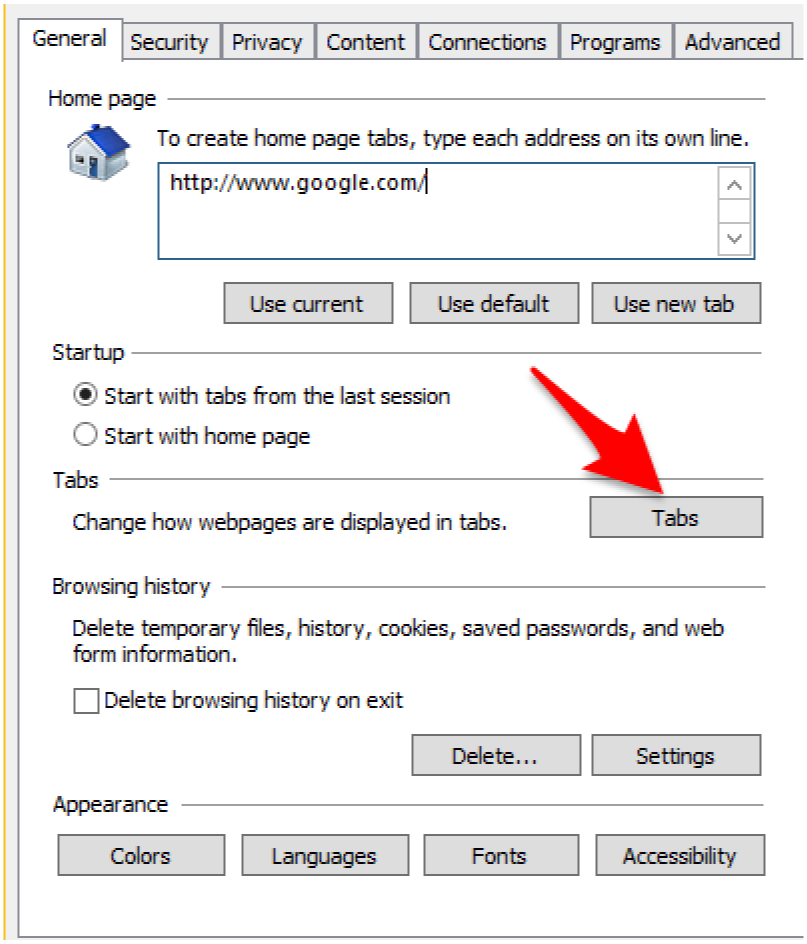
How to open a link in a new tab in Internet Explorer?

1. Go to your browser's settings -> Internet options

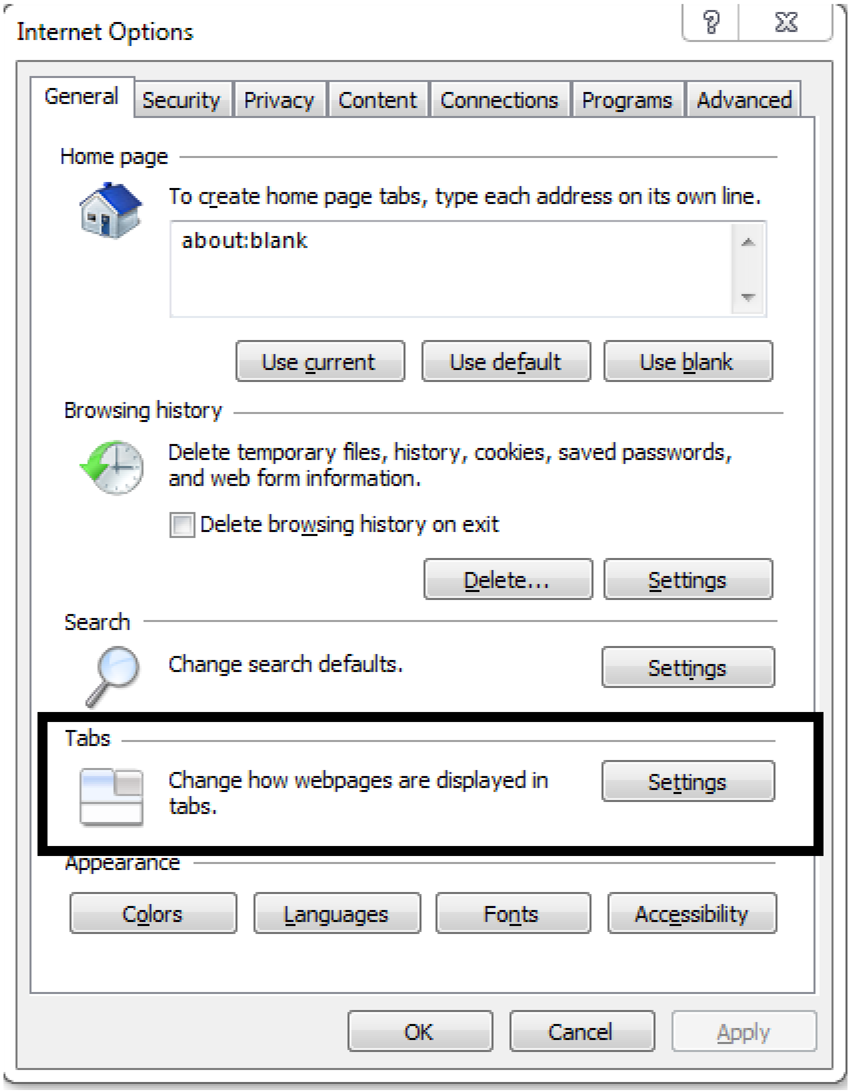


2. Depending on the version of Internet Explorer, you'll want click on the button under the Tabs section

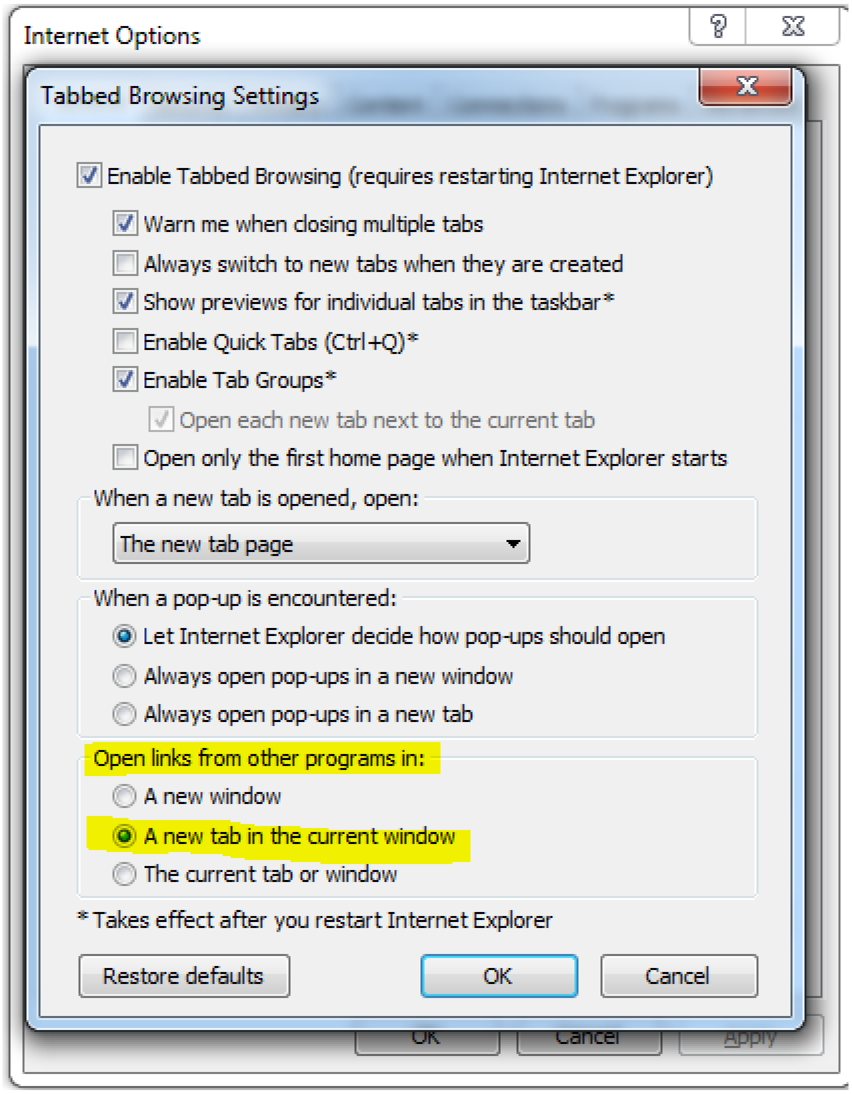
In Internet Explorer 11, click on Tabs >> Tabs as shown below:



In Internet Explorer up to version 10, click on Tabs >> Settings as shown below:



3. As highlighted below, select “A new tab in the current window”



OPEN LINK IN IE

File	Modified
How to open link in new tab in IE.docx	Jan 18, 2018 by Hugo David Buriel-Vasquez

Howto - Request a project copy

RunMyProcess has an internal copy tool named : **SmartCopy**, which can copy **a project** from one platform to another, or one account to another. The tool is operated by RunMyProcess support team. You can create a support ticket (on salseforce) to send your copy request

Publish date	31 May 2018
Contributor	Mani
Reviewer	Richard Mang

Required information

1. Copy from
    - Platform : EMEA, US, JP, Australia
    - Account ID
    - Administrator account: Email / password

- Project name / ID
- Version to copy / version ID

2. Copy to
    - Platform : EMEA, US, JP, Australia
    - Account ID
    - Administrator account: Email / password

Status	PUBLISHE

Things should be avoided for project copy

- Do not include many sub-projects. If project A includes project B, and project B includes C .... it would be complicate to copy all projects at one time. In such case, it is recommended to remove all projects inclusion first, copy all projects indivisually, and re-setup the project inclusion after copy
- Do not include big files in project "Files". If you have test data on "Files", then remove the files before copy

Items that is not copied by the tool

- Runtime user group
- Users inside of a user group
- Provider / Connector login credential

Items changed after copy

To make your application running, you should manually update the information after copy.

1. Account ID
2. Object ID
    - Versioned File ID
    - File ID
3. Resource ID
    - Process ID
    - CAPI ID
    - etc

Note that web interface ID is not changed after copy

Check list after copy

- Menu configuration: Visibility user group settings
- Homepage icons: uploaded image file OID
- Connectors configured to associate between processes, CAPIs
- Providers / Connectors
- User groups
- Reference to a FTL files (versioned file OID) inside of a variable
- Other hard coding inside of your JS, freemarker files

Collection

- Howto - Collection - Access Rights and Enhanced Backoffice User Experience
- Howto - Get data from a collection bigger than 1000 objects
- Howto - Import big size CSV file to a collection
- Howto - Import CSV to Collection

Howto - Collection - Access Rights and Enhanced Backoffice User Experience

Publish date	13 Feb 2018
Contributor	Hugo David E
Reviewer	reviewer meeting

Introduction

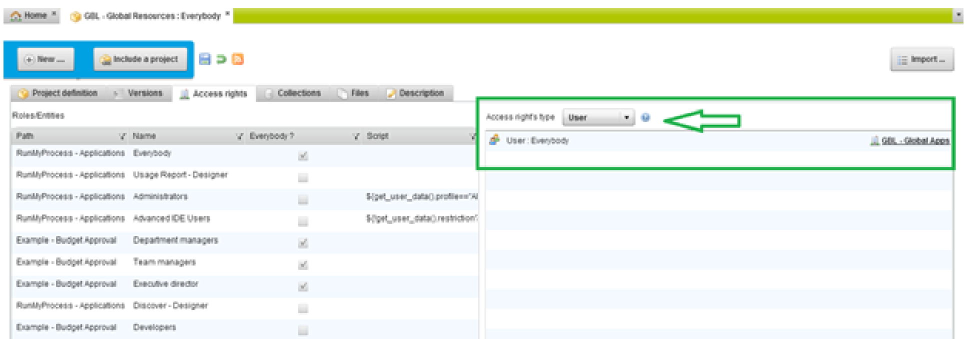
Collection can be used for mainly two reasons – storing application data (such as dropdown fields)

and storing transactional data (instance specific data).
This tutorial is focusing on best practices for managing collection that holds application data, its access rights and improving user experience in backoffice using Jquery dialog box.

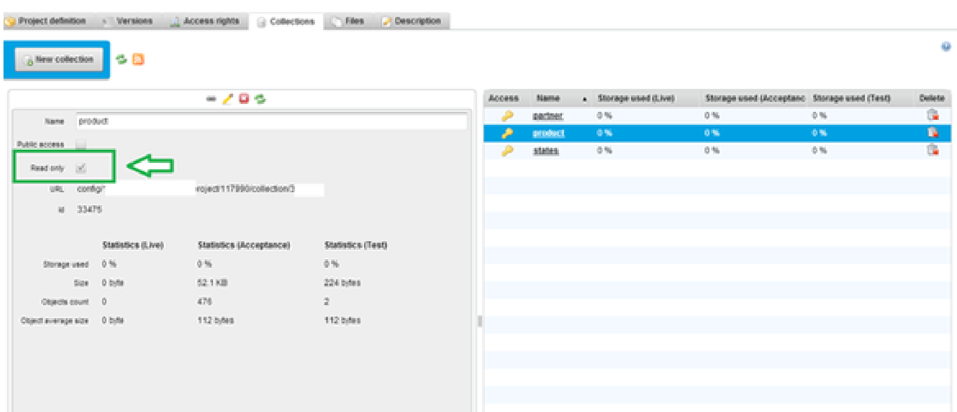
Status	PUBLISHE

Project Architecture

1) Collection should be created inside the MAIN project (that has WI + process + reports) with the USER access rights given to all intended users of the application/collection. Changing a collection name is a bad practice as it is referred in all freemarker or JS that reads or writes into collection. So an appropriate name with all lower cases and if required underscores to concate is recommended. If this collection can be used in other projects in future, then it is good practice to have a dedicated global shared resources project with USER access allowed to everybody that can be included in all projects. Apart from collections, other shared resources like mail provider, custom connectors, css, JS files can be shared in this project.

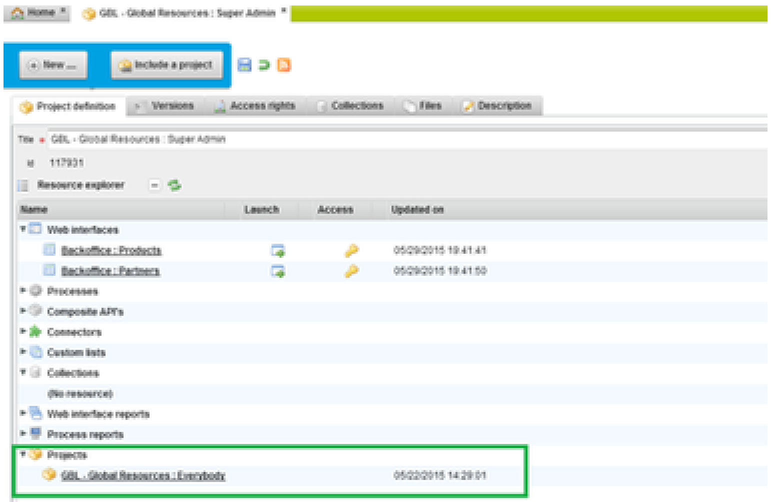


2) It is recommended to set collection as read-only for storing application data (non-transactional) so that the data is not violated by any user. (Most cases, a role with access allowed to everybody has USER access rights of the project, this allows all users to modify(=write) collection, so we need to keep it read-only)

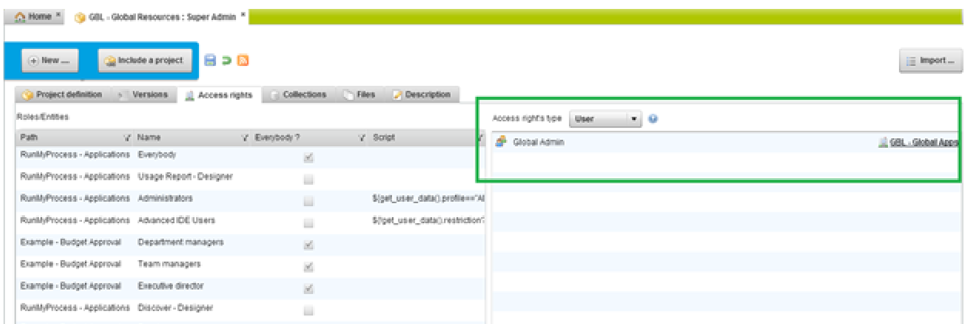


3) There should be a dedicated SUPER ADMIN project that has backoffice web interfaces to allow a database admin type of role to write into collection : add, edit, delete records (=objects).

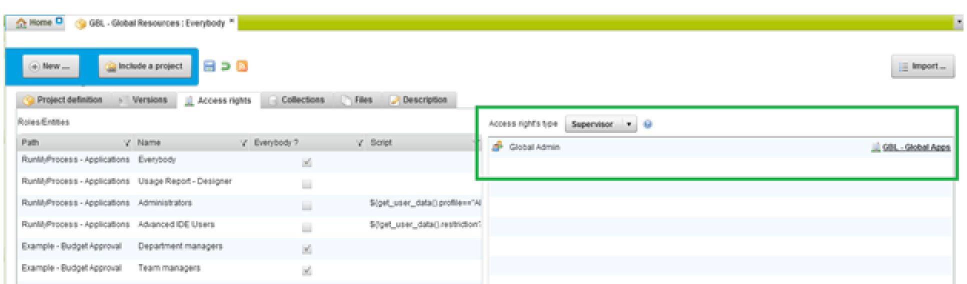
This SUPER ADMIN project should include the MAIN project that contains the collection.



4) USER access rights of this SUPER ADMIN project should be given to a small set of users such as database admin like role.



5) Since the collection in MAIN project was set to read only, we need to give SUPERVISOR access rights of the MAIN project to this database admin role. SUPERVISOR access rights role can write into a read-only collection.

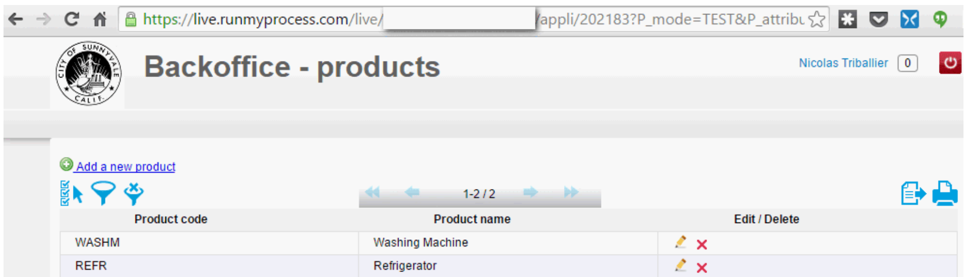


Now, you can create backoffice web interfaces in the SUPER ADMIN project that will allow the database admin role to write into collection. To summarize, this role(Global Admin) has USER access rights on the SUPER ADMIN project that contains backoffices WI and SUPERVISOR access rights on the MAIN project that contains collection. All other users will get Access forbidden error when they open backoffice or via API calls.

Next steps, we discuss how we can improve user experience for database admin in backoffice write.

Configure the backoffice

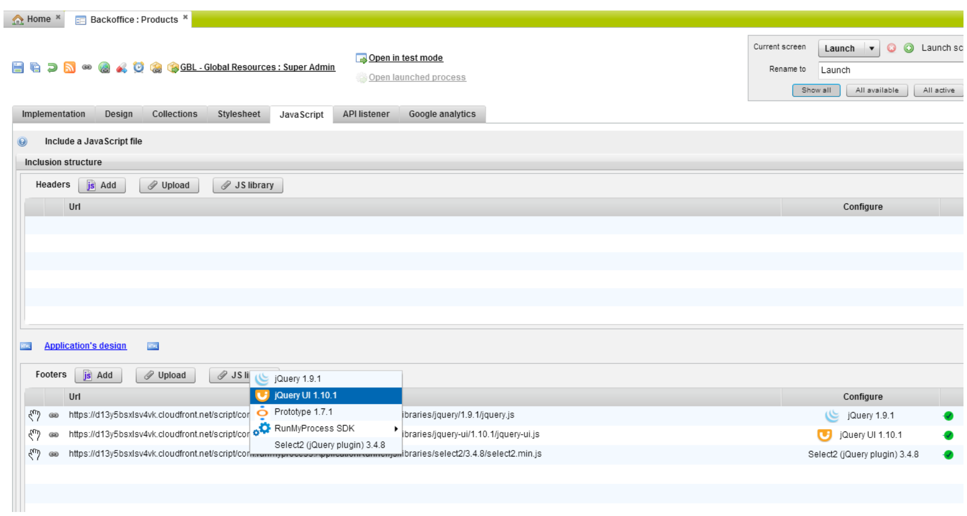
Using JQuery dialog (see its API documentation), user experience is enhanced, by having a dialog box that allows to edit selected record or add a new record (VS using input widgets in custom widget in the older tutorials).



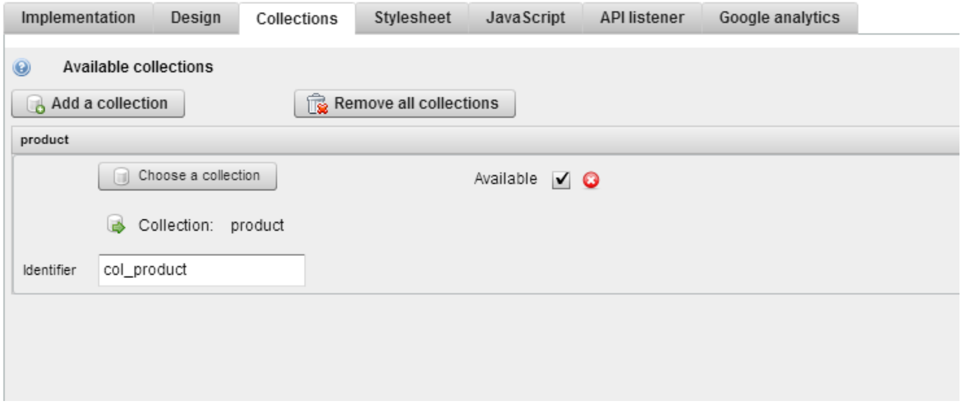
Assuming we have a collection called "product" in MAIN project, with two keys/columns "product_code" and "product_name" where product_code is unique and we need a backoffice for write.

Below are the steps to accomplish this :

Create a new Web Interface for Backoffice in SUPER ADMIN project and include JS libraries for jQuery and jQuery UI from JavaScript tab footers



Attach the collection to the web interface from "Collections" tab and name its identifier (col_product in this case) :



Let's then switch to design tab:

1) Add a HTML widget with code below:

```
<a href="#" onClick="dialog_add_item();return
false;"> <img
src="https://rmp-public.s3.amazonaws.com/public
/icons/n_add.png"/> Add a new product</a>
```

This widget contains message bars which will display after each user action, and a link to add new items in the collection.

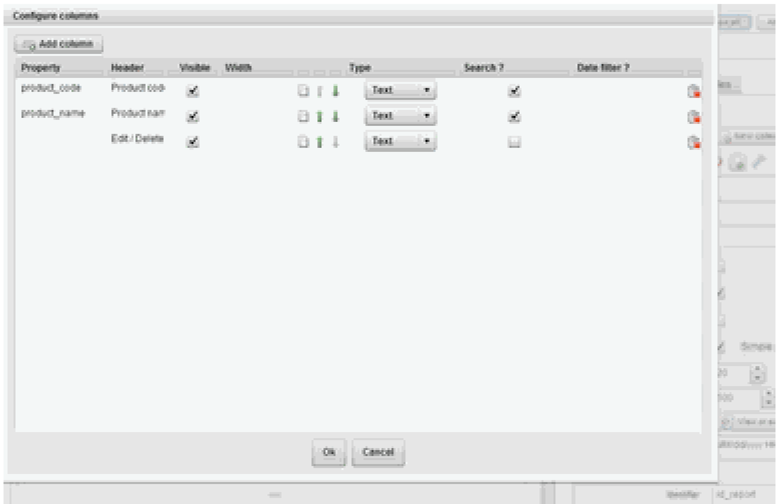
2) Add a report widget for collection, select the collection from the included MAIN project.

Set identifier as "id_report".

Configure the collection column properties and headers

Choose the column that will contain unique IDs : product_code, to use it in Edit/Delete JS.

We'll add control to prevent the user to use the same ID twice.



*Note: **product_code** may have been replaced by **item_id** in the following JS scripts*

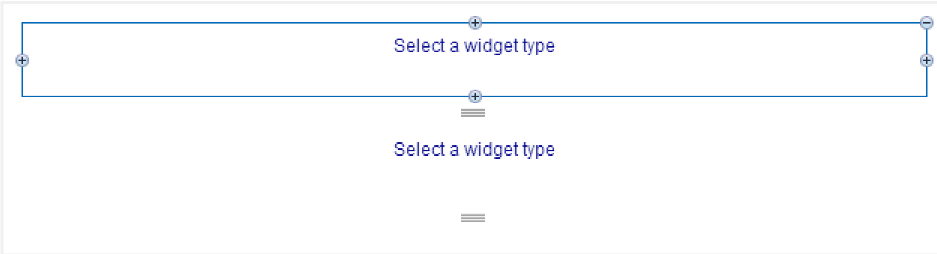
Click on the script icon of the column Edit / Delete:

And add the following code:

```
"<a href=\"#\" onClick=\"javascript:
dialog_load_item(\'" + "[[item_id]]"
+"','edit');return false;\"><img
src=\"https://rmp-public.s3.amazonaws.com/publi
c/icons/n_edit.png\"></a>    <a href=\"#\"
onClick=\"javascript:delete_item(\'" +
"[[item_id]]" +"','delete');return
false;\"><img
src=\"https://rmp-public.s3.amazonaws.com/publi
c/icons/n_delete.png\"></a>";
```

This code will display 2 icons to edit or delete an item.

3) Create a grid widget (click on the "=" bottom right of an empty widget) with identifier "id_dialog":



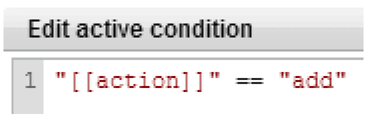
make this grid widget NOT VISIBLE. We'll make it visible using JS.

4) Inside the grid widget, add a text input Product Code:

Label: **Product code**

Variable name: **product_code**

Active condition:



This is meant to prevent the user to modify the ID of the item after its creation

5) Add a text input Product name:

Label: **Product name**
Variable name: **product_name**

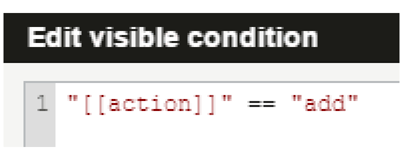
Note: At this stage, there's no required widget. You could add some, and verify they have a value before trying to upsert an item.

6) Add a button 'Add'

Label: **Add**
Type: **Execute script**
Script:

```
add_item();
```

Visible condition:



7) Next to Add, add a button 'Update'

Label: **Update**
Type: **Execute script**
Script:

```
update_item();
```

Edit visible condition

```
"[[action]]" == "edit"
```

8) Next to Update, add a button 'Delete'

Label: **Delete**
Type: **Execute script**
Script:

```
delete_item(RMPApplication.get("item_id"));
```

Visible condition:

Edit visible condition

```
"[[action]]" == "edit"
```

9) Outside of the previous grid widget, create a new HIDDEN grid widget.

10) Within the previous grid widget, add a JS widget:

Label: **Init**

Script:

```
init_dialog();
```

11) plug that .js to your WI

BACKOFFICE

File	Modified
bo.js	Jan 24, 2018 by Hugo David Buriel-Vasquez

Howto - Get data from a collection bigger than 1000 objects

The limitation of list a collection in freemarker and JS are 1000. If you have more than 1000 in your collection (RMP Doc), you can use the following solution.

Publish date	05 Jan 2018
Contributor	Rafal Urbans Mani
Reviewer	Mani Richard Mang

This document partially quotes from codegeek provided by Rafal Urbanski

Freemarker example

Status	PUBLISHED

- Freemarker example
- Javascript example

```
<#function read_all_data pattern collection_name first_index>
    <#assign list = list_objects(pattern,collection_name,first_index*1000,1000)>
    <#if (list?size == 1000)>
        <#assign list = list+ read_all_data(pattern,collection_name,first_index+1)>
        <#return list>
    <#else>
        <#return list>
    </#if>
</#function>
```

Javascript example

> Note that it is not recommended to retrieve a big number of data on a web interface at one time, because it may impact on the performance. The following code is only an example.

```
var P_first_index = 0;
var return_data = [];

function list_all_records() {
    get_all_data(0);
}

function get_all_data(my_index) {
    function list_all_ok(return_data) {
        return function(result){

            return_data.push(result);

            if(result.length == 1000){
                P_first_index  = P_first_index
+ 1000;
                return
get_all_data(P_first_index);
            }

            var all_data =
Array.prototype.concat.apply([], return_data);
            //all_data is the complete data of
mycollection
            console.log("all_data = " +
all_data.length);
        };
    }

    P_first_index = my_index;

col_mycollection.listCallback({},{first:my_inde
x, nb:1000},
list_all_ok(return_data),list_all_ko);
}
```

Howto - Import big size CSV file to a collection

This document describes how to import a big CSV file to a collection (E.g. file size is bigger than the platform limitation, etc). For the normal use case, refer to Howto - Import CSV to Collection

This document partially quotes from codegeek provided by Rafal Urbanski

Publish date	13 Feb 2018
Contributor	Rafal Urbans Mani
Reviewer	reviewer meeting

Status	PUBLISHE

Background

This is a best practice of migration big CSV data to a RunMyProcess collection. It is a useful use case when replacing the existing IT system to RunMyProcess .

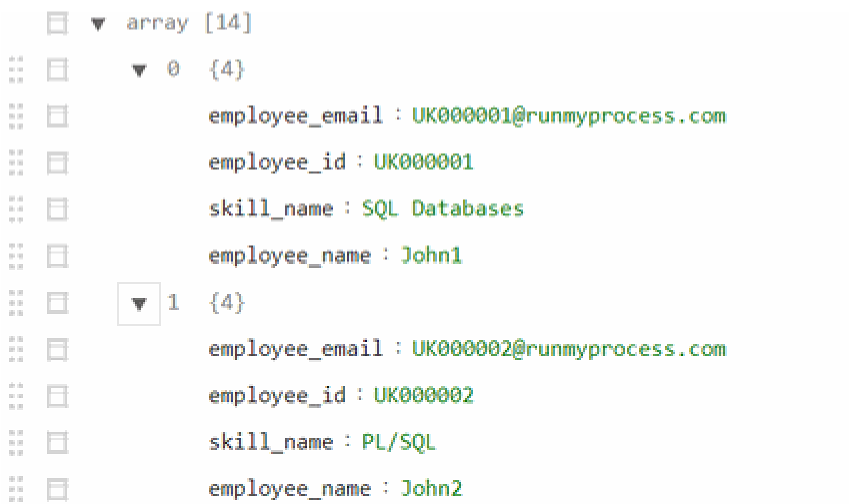
Why a best practice is required:

- RunMyProcess platform has upload file limitation on the platform side (Refer to RMP doc)
- When the data is very big, optimization may be necessary

Input : CSV sample

```
employee_id,employee_email,employee_name,skill_name
UK000001,UK000001@runmyprocess.com,John1,SQL Databases
UK000002,UK000002@runmyprocess.com,John2,PL/SQL
...
```

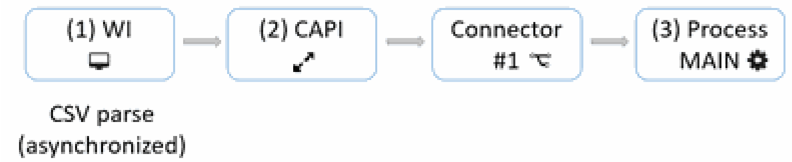
Output : Collection (JSON)



Procedure Overview

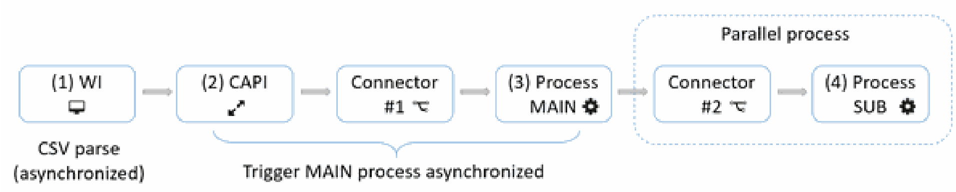
General use case

1. Create a simple web interface to upload and parse the CSV file to JSON
2. Import the data to collection using process



Optimization solution

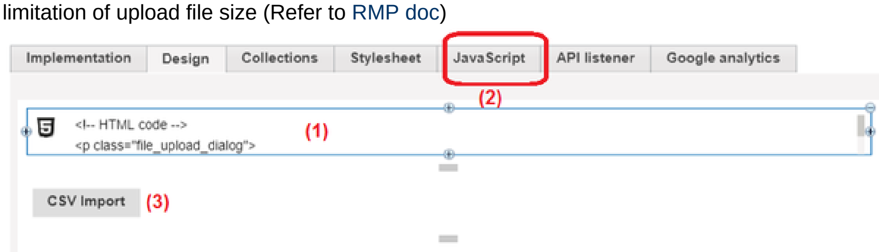
1. Create a simple web interface to upload and parse the CSV file to JSON
2. Split the one big JSON to multiple small JSONs
3. Import the data to collection using asynchronized parallel process



Steps (General User Case)

1. Prepare the CSV parse library
   For parsing a big CSV file with high performance and error handling, it is recommended to use the following external javascript library. Download file "papaparse.min.js" from http://papaparse.com/, and upload it to your project.

2. Create a simple web interface for the CSV upload.
   The benefit of creating a web interface instead of uploading file to the project directly ("Files" tab on the project), is that the CSV file locates on the client site in this way, and it has no limitation of upload file size (Refer to RMP doc)



   (1) A html widget. ID = id_file_upload
   HTML Code:

```
<!-- HTML code -->
<p class="file_upload_dialog">
Select a csv file to upload: <input
type="file" id="my_csv_file"
onchange="validate_file_ext(this);">
</p>
```

(2) Add papaparse.min.js to the footer of the web interface

(3) Javascript code of upload CSV file

```

```

3. Create a simple process to save the data to the collection. Since the data is already parsed to JSON, you just need to loop and save it to your collection


```
<#list csvdata as a_data >
${save_object(a_data,
"csv_import_collection")}
</#list>
```

4. Create a CAPI to connect between the web interface and the process
Refer Howto - Connect two processes / CAPI for detail steps. Note that the input data property should match between the javascript and the process.

The reason of using CAPI connector Process, is that the CSV parsing operation is executed asynchronized when the "CSV Import" button is clicked. If the button click starts a process directly, the CSV parsing is not done yet.

Steps (Optimization)

- In (2)-3, add the following code to divide a big CSV file to multiple array groups

```

```

- Instead of one process, create a father process and child process to process the save_objects parallelly.
Refer Howto - Synchronize multiple asynchronized processes for detailed steps.

Note that parallel process and the data size processed by one process, should be decided based on your actual data size.
In above example, suppose you have 100K data, and 1 group has 10K data, it means that the data is processed in 10 child processes, parallelly.

> Please notify RunMyProcess team **in advance** if you need any advice on performance tuning or testing a big data.

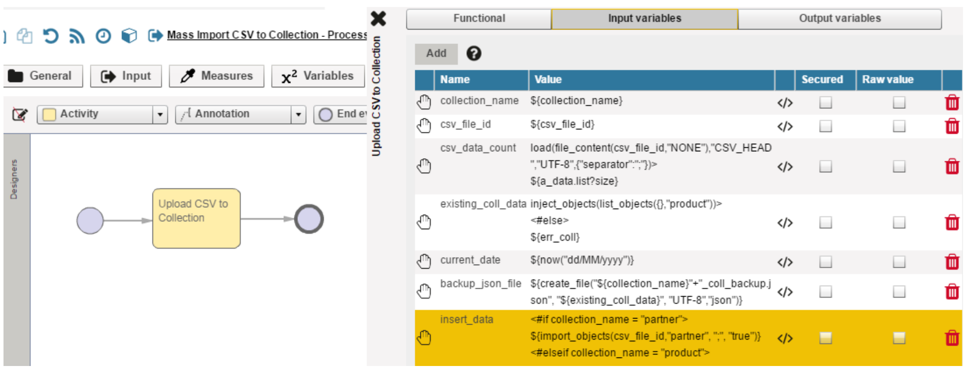
Howto - Import CSV to Collection

This is the general use case to import a CSV file to RMP collection. For the big data size CSV, refer to Howto - Import big size CSV file to a collection

Publish date	13 Feb 2018
Contributor	Hugo David E
Reviewer	reviewer meeting
Status	PUBLISHE

To populate a collection you can mass import a CSV into collection.
Steps to be followed:

1. First you need to upload the CSV file into your project and then use ${import_objects(file_id, collection_name [, separator [, drop]])} function in the process to parse the CSV and load into collection.



2. Create a process with 1 activity.
3. In the input parameters, feed info like collection name and uploaded csv id.
4. If there is some existing data in collection and if want to have a backup JSON file,
   a. you can check if the csv has data
      csv_data_count:
      <#assign a_data =
load(file_content(csv_file_id,"NONE"),"CSV_HEAD","UTF-8",{"separator":";"})>
      ${a_data.list?size}
   b. if data exists, store the collection data in a variable called "existing_coll_data".
      existing_coll_data:
      <#if csv_data_count?number!= 0>
        <#if collection_name = "partner">
          <#assign coll_data = inject_objects(list_objects({},"partner"))>
        <#elseif collection_name = "product">
          <#assign coll_data = inject_objects(list_objects({},"product"))>
        <#else>
       ${err_coll}
        </#if>
      </#if>
      ${coll_data}
   c. create a backup JSON file using the existing collection data
      backup_json_file:
      ${create_file("${collection_name}"+"_coll_backup.json", "${existing_coll_data}",
"UTF-8","json")}

4. Now insert the new csv data into collection
   insert_data:
   <#if collection_name = "partner">
     ${import_objects(csv_file_id,"partner", ";", "true")}
   <#elseif collection_name = "product">
     ${import_objects(csv_file_id,"product", ";", "true")}
   <#else>
   ${err_coll}
   </#if>

Launch the process in TEST/ACCEPTANCE/LIVE depending on which collection mode you want to fill.

Done !!

Homepage

Howto - Add dynamic translation

This document partially quotes from codegeek provided by Rafal Urbanski

Publish date	05 Jan 2018
Contributor	Rafal Urbans Mani
Reviewer	Farva Hussai
Status	PUBLISHE

RMP provides a translation tool called APP translator. This application is deployed to all accounts by default.

Using this application, you can support multi languages for your application. When a user login to the platform, the translation will be automatically displayed based on the language setting in the user profile. Note that the dictionary file in APP translator shows labels, messages, basket on a web interface by default. You also can use the following way to **add** more words in the dictionary file, to use the APP translator in more situations.

```
${i18n([name_of_the_variable],[Text to be
translated])}
```

- name_of_the_variable
  You will find the variable in the dictionary file in APP Translator
- Text to be translated
  This is the part you need to add translations

Use case - Email notification

The following example shows an email title, whose language can be dynamically changed in an email sending notification activity. You can use the same way for the contents as well.

```
${i18n('manager_approval_subject','New Holiday
Request')}
```

Use case - Variable based list

A variable based list is created when the page is loaded. You can add translations in the variable based list as well.

```
var my_list = [
    {"value":"OK",
"label":${P_quoted(i18n('choice_answer_ok','OK'
))}},
    {"value":"Not Sure",
"label":${P_quoted(i18n('choice_answer_notsure'
,'Not Sure'))}}
];

var rmp_list = new RMP_List();
rmp_list.fromArray(my_list);
RMPApplication.setList("vb_choices",rmp_list);
```

Use case - Homepage icon

This use case aims to show the icon's name in different languages on the platform homepage. Open the web interface of "RunMyProcess - Homepage" project using IDE. Add the following code on function "**sectionManagement** (tag, apps)".

```
function sectionManagement(tag, apps){
    var sectionName = tag ==
'NO_TAG'?DEFAULT_SECTION_NAME:tag;
    RMPApplication.debug('Tag management:
'+tag);
    var html = '';
    apps.sort(function(a, b){return a.title &&
b.title &&
a.title.toUpperCase()>b.title.toUpperCase()?1:-
1;});

    for (var index = 0; index < apps.length;
index++){
        var appId = apps[index].id;

        if (appId != homepageId){
            var styleAttr = '';
            if (apps[index].thumb &&
apps[index].thumb != ''){
                styleAttr = 'style="background:
```

```
url(\''+apps[index].thumb+'\')"';
            }


//*********************************************
**
            //add icon name translation
            var cur_title = apps[index].title;
            if (appId === 123456) {
                cur_title =
${P_quoted(i18n("ICON_NAME","SafetyCheck-SendRe
quest"))};
            }

//*********************************************
**

            var context = { name:cur_title,
                url:apps[index].url,
                id:appId,

description:apps[index].description,
                appIconClass:
'bigIcon-defaultApps0',

descrClass:apps[index].description &&
apps[index].description.length>0?'':'hidden',
                attr:styleAttr
            };
            html +=
compile(APPLICATION_TEMPLATE, context);

        }
```

```
    }

...
```

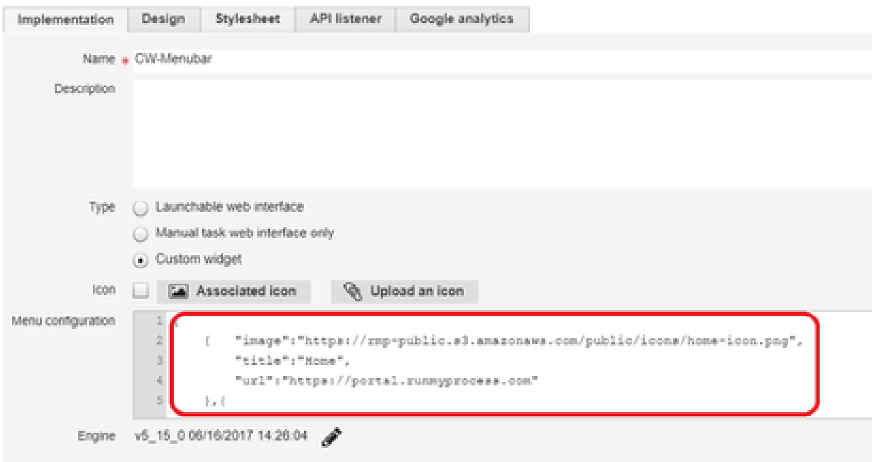
Howto - Best practice of application menu tab

This document partially quotes from codegeek provided by Rafal Urbanski

Publish date	05 Jan 2018
Contributor	Rafal Urbans Mani
Reviewer	Farva Hussai
Status	PUBLISHE

Here is the best practice of creating a common menu bar for all web interfaces in one application.

1. Create a custom widget

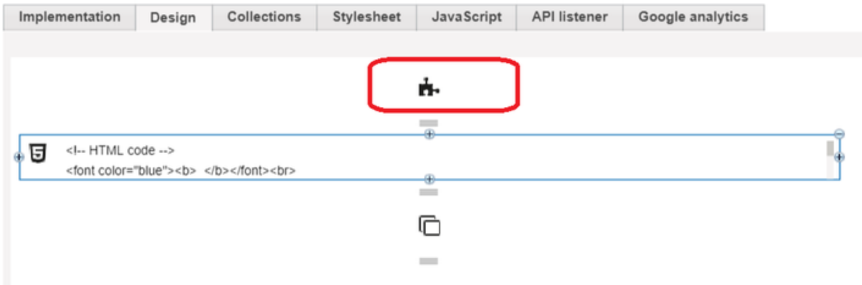


2. Add the following sample code on the menu configuration

```
[{
        "title" : "Home",
        "image" :
"https://rmp-public.s3.amazonaws.com/public
/icons/home-icon.png",
        "url" : "123456?P_mode=${P_mode}"
    }, {
    "title" : "Menu 1",
        "option" : [{
            "title" : "Menu 1.1",
            "url" :
"123456?P_mode=${P_mode}"
        }, {
            "title" : "Menu 1.2",
            "url" :
"123456?P_mode=${P_mode}"
        }, {
            "title" : "Menu 1.3",
            "url" :
"123456?P_mode=${P_mode}",
        }]
    }, {
        "title" : "Menu 2",
        "visible" :
"${has_right('229820')='true' ||
has_right('262048')='true'}",
        "option" : [{
            "title" : "Menu 2.1",
            "url" :
"123456?P_mode=${P_mode}"
        }, {
            "title" : "Menu 2.2",
            "url" :
"123456?P_mode=${P_mode}&scnMode=MQ=="
        }, {
            "title" : "Menu 2.3",
            "visible" :
"${has_right('229820')}",
            "url" :
"123456?P_mode=${P_mode}"
        }]
    }
]
```

a. "url" should be changed to the actual web interface url
b. "visible" : "${has_right('229820')='true' || has_right('262048')='true'}" means that the web interface can be seen by user group 229820 or 262048 users only

3. Add the customer widget to all web interfaces, on the top



Howto - Edit basket and basket popup on homepage

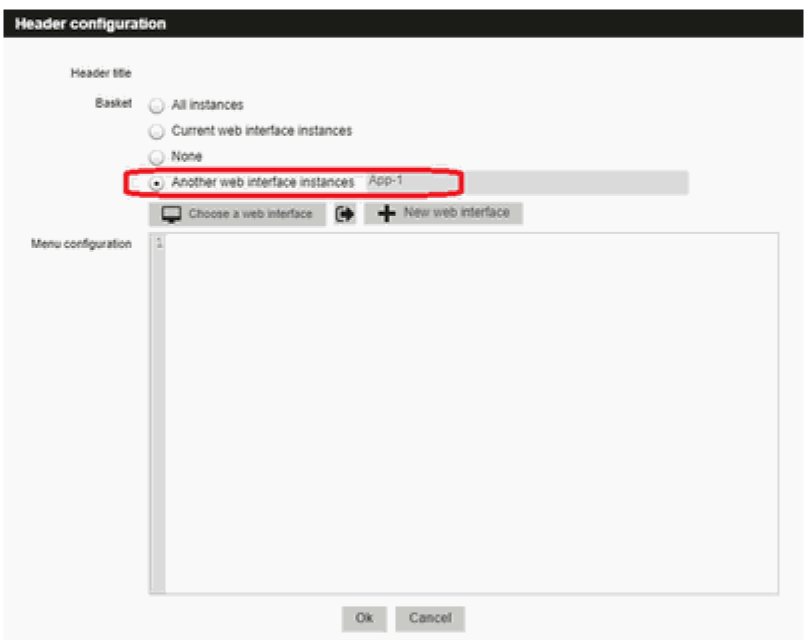
Publish date	04 May 2018
Contributor	Mani
Reviewer	Reviewer meeting
Status	PUBLISHE

Issue

There is a known issue of setting a web interface to use another web interface instance's basket. For example, you have App1 and App2 web interfaces in a same project.

- App1 icon - Visible by user group1 (e.g. admin)
- App2 icon - Visible by user group2 (e.g. user)

The task is associated to App1 only. In this case, you can setup App1's basket shows "None", and setup App2 to show basket of App1 with the followin settings:



However, the setting doesn't work as of April 2018 (ticket : server #612). The document describes a workaround.

Solution

Update "RunMyProcess - Homepage" project with the following code

```
/**
```

```
Basket management.
@See:
http://docs.runmyprocess.com/Developer_Guide/We
b_Interface/User_Experience/Basket
*/
function notify(data){

    /*Add the additional code here. Replace the
App1's ID (string) to App2's ID (string) */
    var new_assigned_to =
JSON.parse(JSON.stringify(data.ASSIGNED_TO).rep
lace(App1_id, App2_id));
    var new_data = {};
    new_data.INITIATED_BY = data.INITIATED_BY;
    new_data.ASSIGNED_TO = new_assigned_to;
    /* ---------------- */

    for (var tag in tags){
        if (tags.hasOwnProperty(tag)){
            for (var appIndex = 0; appIndex <
tags[tag].length; appIndex++){
                var id =
tags[tag][appIndex].id;
                var b = jQuery('.b'+id);
                /*Update all object names from
data to new_data */
                if (new_data.ASSIGNED_TO &&
new_data.ASSIGNED_TO[1] &&
new_data.ASSIGNED_TO[1][id]){

b.html(new_data.ASSIGNED_TO[1][id]);
                    b.addClass('rmpbactive');
                }else{

b.removeClass('rmpbactive');
                }
            }
        }
    }
}

function openBasket(id){

    /* Add the additional code here. Replace
the App2's ID to App1's ID */
    if (id === App2_id) {
       id = App1_id;
    }

    RMP_basket.popup(id, 'ASSIGNED_TO', 1);
```

```
}
```

Howto - Get basket open task number from API

> This will be added to SDK in near future.

Publish date	13 Feb 2018
Contributor	Mani
Reviewer	reviewer meeting
Status	PUBLISHED

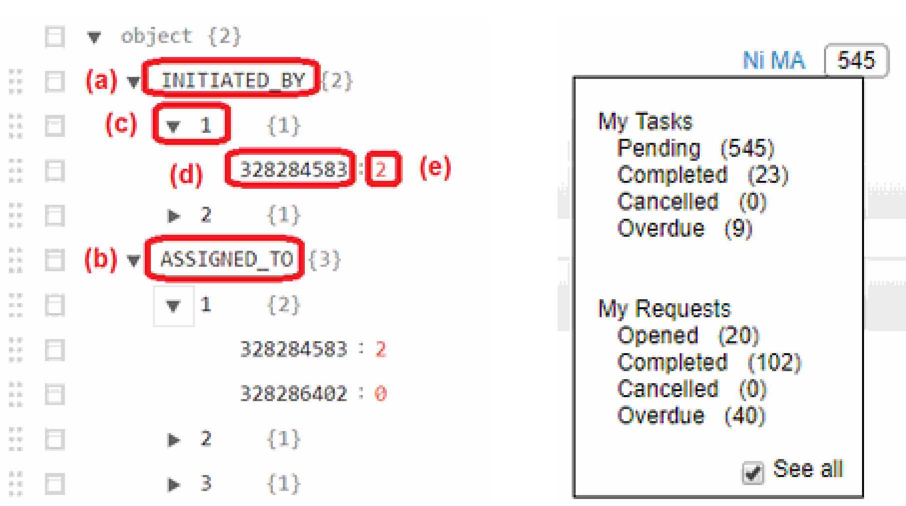
In case that the customer wants to build a portal without using RMP homepage, and show RMP open task number on this portal, it is possible to get the **current login user**'s basket information from API. Note that the user has to login to RMP (e.g. SSO)

Get task number

Format

- URL: https://live.runmyprocess.com/live/***{customer_id}***/basket?P_mode=LIVE
- content-type: application/json
- accept: application/json

Output example



(*Above two screenshots are taken from different environment. The numbers don't match each other)

- (a) INITIATED_BY: My Requests
- (b) ASSIGNED_TO: My Tasks
- (c): Task status code (See details : RMP manual)

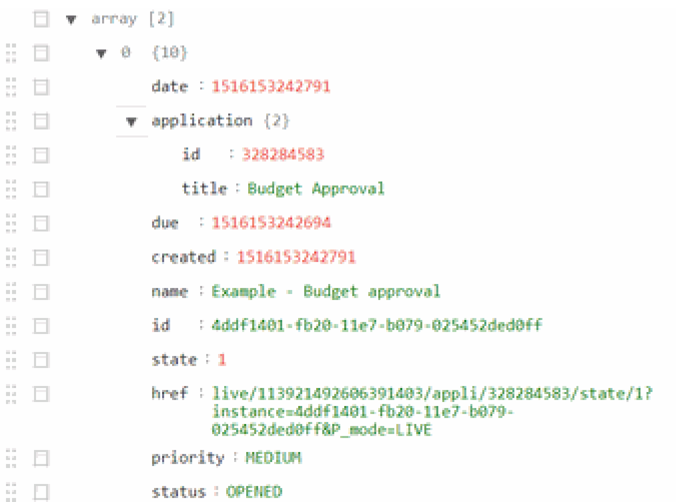
- (d): Application ID
- (e): Open task number

Get more task details

Format

- URL: https://live.runmyprocess.com/live/***{customer_id}***/basket?P_mode=LIVE&filter=ASSIGNED_TO&operator=EE&value=user
- content-type: application/json
- accept: application/json

Output example



- date: updated date (timestamp)
- application: web interface name and id
- due: due date of the task (timestamp)
- created: created date of the task (timestamp)
- name: project name
- id: instance id of the task
- state: task status code (See details: RMP manual)
- href: task url
- priority: task priority
- status: task status string

Input / Output

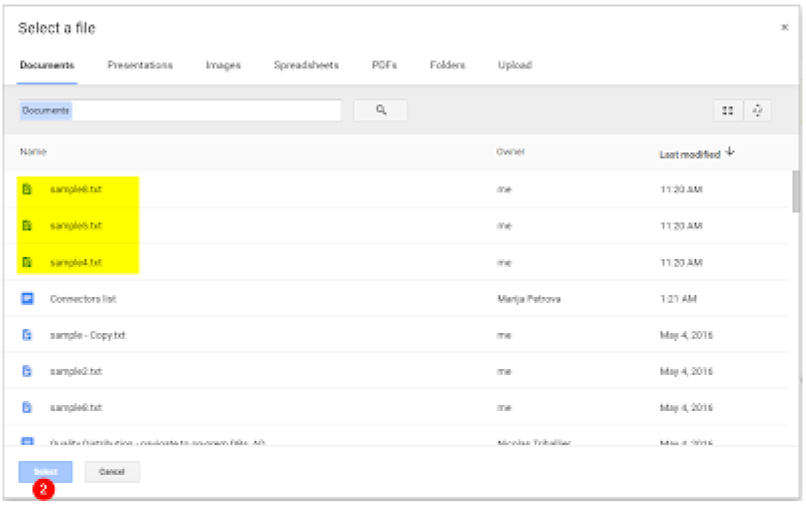
- Howto - Google Drive File Picker
- Howto - PDF templates - Formatting Numbers in PDF template
- Howto - PDF templates - Handle Japanese characters
- Howto - PDF templates - Preserving line spaces in multi-line text boxes
- Howto - QR code generation + inject in PDF

Howto - Google Drive File Picker

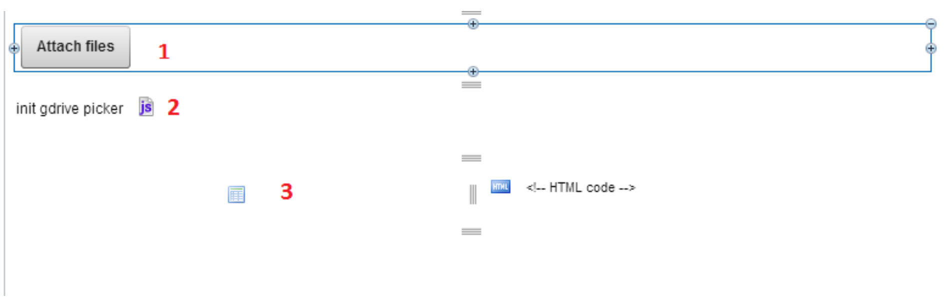
Publish date	04 May 2018
Contributor	Hugo David E

RunMyProcess native file upload widget on the web interface does not support files exceeding 25mb in size and multiple selection of files at once, and if you are using Google Apps, it could be replaced with a Google Drive File Picker.
This picker allows selecting existing files from G Drive as well as uploading new files directly to G Drive. Selected files can be viewed from array widget on RunMyProcess.

Reviewer	Reviewer meeting
Status	PUBLISHED



Web Interface design:



0. Add this JS library to the web interface from 'Javascript' tab -> 'Application Design':

1. Create a button 'Attach files' with action 'Execute Script' :
2. Create a Javascript widget with the below script:

Note: You have to obtain your Browser API key, Client ID from Google developers console to use it in the below script. You can refer below tutorial on how to obtain those keys.
_Google oAuth 2.0 configuration & Calendar APIs basic example

```
//The Client ID obtained from the Google
Developers Console.
    var clientId = 'xxx';

//G Drive Folder ID where files will be
uploaded
  var FOLDER_ID = 'xxx';

//Scope to use to access user's Drive items.
    var scope =
['https://www.googleapis.com/auth/drive'];
```

```
    var pickerApiLoaded = false;
    var oauthToken;

// Use the Google API Loader script to load the
google.picker script.
    function loadPicker() {
      gapi.load('auth', {'callback':
onAuthApiLoad});
      gapi.load('picker', {'callback':
onPickerApiLoad});
    }

    function onAuthApiLoad() {
      window.gapi.auth.authorize(
          {
            'client_id': clientId,
            'scope': scope,
            'immediate': false
          },
          handleAuthResult);
    }

    function onPickerApiLoad() {
      pickerApiLoaded = true;
      createPicker();
    }

    function handleAuthResult(authResult) {
      if (authResult && !authResult.error) {
        oauthToken = authResult.access_token;
        createPicker();
      }
    }

//createPicker function defintion in an
external versioned file

//Callback implementation to RMP array widget
function pickerCallback(data) {
 if (data.action ==
google.picker.Action.PICKED) {
 var gfiles = data.docs;
 //add new array row for subsequent pick after
initial one
 if (RMPApplication.get("arr_files")!=null){
 id_arr_files.setAddRows(true);
 id_arr_files.insertRow(id_arr_files.getRowsCou
nt().toString());
 id_arr_files.setAddRows(false);
```

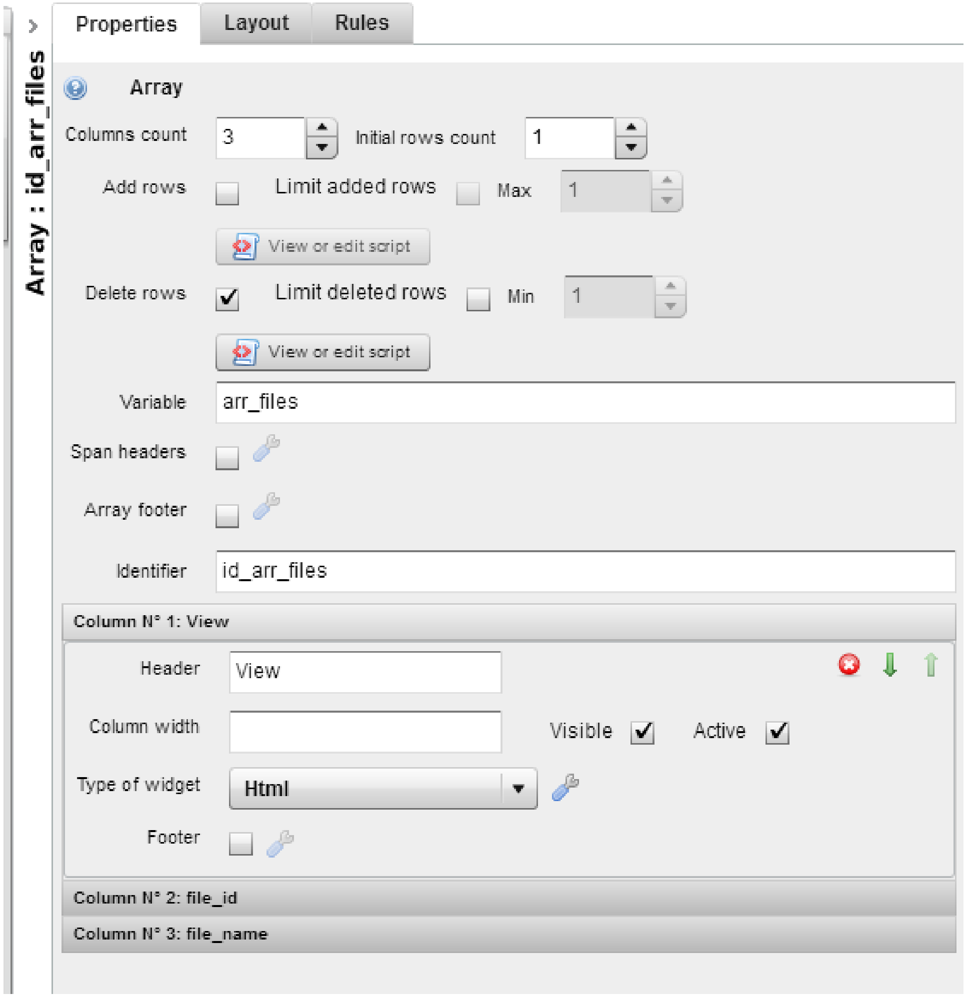
```
 }
 //loop through each picked file
 for (var i=0; i<gfiles.length; i++){
 var json_row ={};
 json_row.url = gfiles[i].url;
 json_row.name = gfiles[i].name;
 json_row.id = gfiles[i].id;
 var this_link = "<a href='"+ json_row.url +"'
target='_blank'>" + json_row.name +"</a>";
 var this_row = id_arr_files.getRowsCount();
 //set array row link and hidden var
 id_arr_files.id_file_link[this_row-1].setHtml(
this_link);
 id_arr_files.id_file_id[this_row-1].setText(js
on_row.id);
 id_arr_files.id_file_name[this_row-1].setText(
json_row.name);
 //add new array row for each file
 if(i!=gfiles.length-1){
 id_arr_files.setAddRows(true);
 id_arr_files.insertRow(this_row.toString());
 id_arr_files.setAddRows(false);
 }
```

```
 }
 }
}
```

Create an external versioned JS file to define createPicker() because the picker definition may need maintenance if Google changes its behavior :

```
//Create and render a GDrive Picker object
    function createPicker() {
      if (pickerApiLoaded && oauthToken) {
 //Used DocsUploadView to allow uploading
documents
        var uploadView = new
google.picker.DocsUploadView().setParent(FOLDER
_ID);
        var docs_view = new
google.picker.DocsView();
             docs_view.setIncludeFolders(true);

docs_view.setSelectFolderEnabled(true);
        var picker = new
google.picker.PickerBuilder()

.enableFeature(google.picker.Feature.MULTISELEC
T_ENABLED)    //allows to pick multiple files
            .addView(docs_view)
 .addView(google.picker.ViewId.DOCUMENTS)

.addView(google.picker.ViewId.PRESENTATIONS)

.addView(google.picker.ViewId.DOCS_IMAGES)

.addView(google.picker.ViewId.SPREADSHEETS)
            .addView(google.picker.ViewId.PDFS)

//.addView(google.picker.ViewId.FOLDERS)
            .addView(uploadView)
            .setOAuthToken(oauthToken)
            .setCallback(pickerCallback)
            .build();
       picker.setVisible(true);
      }
    }
```

3. Create an Array widget (variable: arr_files) :



- Add 3 columns and 1 initial row
- Disable 'Add rows' checkbox
- Column 1 : View
  - Type : Html
  - Id : id_file_link
  - Rules : Visible, Active
- Column 2 : file_id
  - Type : Text input
  - Id : id_file_id
  - Rules : Active
- Column 3 : file_name
  - Type : Text input
  - Id : id_file_name
  - Rules : Active

Done!!!

(PS : Make sure you do not have a javascript variable called 'screen')

Howto - PDF templates - Formatting Numbers in PDF template

Publish date	15 Mar 2018
Contributor	Hugo David E

The format-number() function is used to convert a number into a string.

In order to format your number, you need to define it in the data source variable (*see the "How to generate a PDF file" link*)

Then put this code in your .xsl file :

Result :

Refer to : How to generate a PDF file

Reviewer	Philippe Lubr
Status	PUBLISHE

Howto - PDF templates - Handle Japanese characters

Handle Japanese characters

ex of code in the .xsl file:

ex of variable to inject:

Download: xsl template | pdf

File	Modified
CHAR_PDF_TEST.pdf	Jan 24, 2018 by Hugo David Buriel-Vasquez
CHAR_PDF_TEST.xsl	Jan 24, 2018 by Hugo David Buriel-Vasquez

Download All

Publish date	15 Mar 2018
Contributor	Hugo David E
Reviewer	Philippe Lubr
Status	PUBLISHE

- Handle Japanese characters

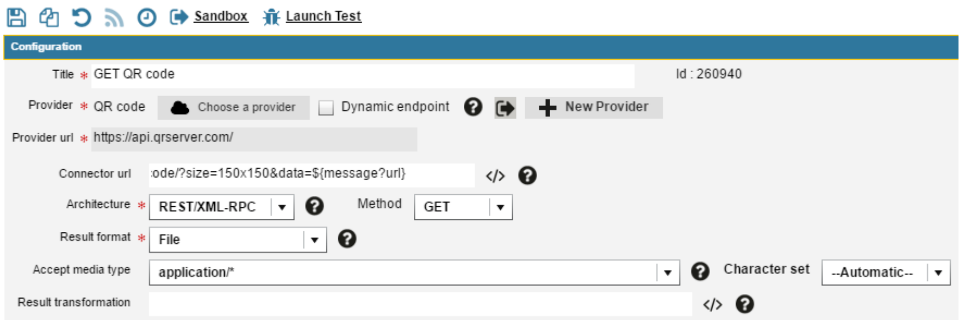
Howto - PDF templates - Preserving line spaces in multi-line text boxes

Publish date	15 Mar 2018
Contributor	Hugo David E
Reviewer	Philippe Lubr
Status	PUBLISHE

Howto - QR code generation + inject in PDF

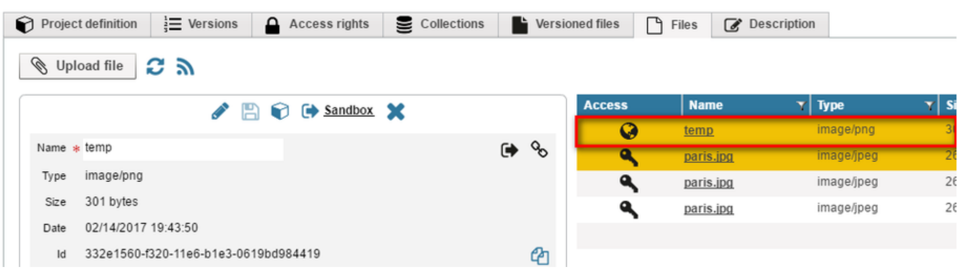
Publish date	04 May 2018
Contributor	Hugo David E
Reviewer	Reviewer meeting
Status	PUBLISHE

To generate a QR code, you'll need to use a 3rd party Webservice like:
http://goqr.me/api/ (free to use FYI).
Create a connector that will make a GET on https://api.qrserver.com/v1/create-qr-code/?size=150x150&data=Example



With the file id, you can then make the file public (necessary to be able to inject it in the pdf):

Your file is now public:



You can then inject the file url in your pdf:
https://live.runmyprocess.com/pub/${P_customer}/upload/${file_id}/temp
You should configure all those steps in a subprocess or CAPI.

Integration

- Howto - How to free a port already in use
- Howto - SEC background execution on Windows Server
- Howto - Test SEC + JDBC adapter + MySQL

Howto - How to free a port already in use

Publish date	12 Jun 2018
Contributor	Hugo David E
Reviewer	Sei FUJITA
Status	PUBLISHE

When using SEC and adapters, if you want to run a program on a particular port but if it is already in use by some other program or if it is not terminated correctly, you can check which program is running on that port by using following command on command prompt and then kill the process forcefully. You should do this only if you are sure terminating that program won't cause any issue.

```
netstat -aon | findstr 123456
```

```
O/P: TCP 0.0.0.0:9999 0.0.0.0:0 LISTENING 7008
```

where 123456 is the port number. It lists what program is using it and what is the process id of it.

`-a` Displays all connections and listening ports.

`-o` Displays the owning process ID associated with each connection.

`-n` Displays addresses and port numbers in numerical form.

You can kill the program by using its process id either from task manager or from command prompt by using the following command.

```
taskkill /F /PID 7008
```

where 7008 is the Process ID.

`/F` - Specifies to forcefully terminate the process(es).

Note: You may need an extra permission (run from admin) to kill some certain processes.

Howto - SEC background execution on Windows Server

Publish date	15 Mar 2018
Contributor	Sei FUJITA
Reviewer	Mani
Status	PUBLISHE

Issue

This is a known issue about Windows Server **command prompt**. When SEC is installed on a Windows Server, SEC java modules (Agent, Manager and Adapters)'s memory usage would be effected by the way how you start the jar file. The RMP doc shows an example using command prompt on ubuntu. However, if you execute SEC on a windows server, using windows command prompt would cause memory usage increase unexpectly.

Solution

Register a batch to launch SEC java modules (Agent, Manager and Adapters) by windows Task Scheduler.

https://msdn.microsoft.com/en-us/library/aa383614.aspx

What you have to do is only "create a bat file" and "register it as a task".

#Task scheduling is also useful for you to avoid forgetting launch SEC modules when you reboot your server.

Steps

1. Create a text file.
2. Copy & paste the sample code below and replace the sample file paths to yours.

    sec_start.bat

3. Save it as a bat file like "sec_launcher.bat".
4. Register the bat file as a task on "Task Scheduler"
   Please use Windows Task scheduler to do that.

   You can find an example to use Task scheduler here.
   https://blogs.technet.microsoft.com/orchestrator/2012/05/18/using-windows-task-scheduler-to-invoke-scheduled-runbooks/
   You can find detailed information about task scheduler here.
   https://msdn.microsoft.com/en-us//library/windows/desktop/aa446802(v=vs.85).aspx
   "Launch your bat file when the system starts up" trigger is usually good for you.

Howto - Test SEC + JDBC adapter + MySQL

Publish date	12 Jun 2018
Contributor	Hugo David E
Reviewer	Sei FUJITA
Status	PUBLISHE

1.Setup WAMP

1.1 Installer
http://www.wampserver.com/en/
> Windows
> Apache
> MySQL
> PHP
install then boot the servers (stop skype before)

2. WAMP Settings

Please find WAMP's icon in your menu bar.
2.1 port setting
Right click the icon > Tools > a port used by Apache > Use a port other than 80
change the port from 80 to the other like 8000.
#8080 is used by SEC manager.
2.2 Database setting
-login php console
Left click the icon > phpMyAdmin > login
Default User name: "root"
Default Password: ""(blank)

-Create a database
input database name(e.g. "sec_test") > Create
-Create table
input table name(e.g. Name: "cars", No of columns: 2) > Go
Setup table as you like(e.g. id&name)
-Add test data
Insert > input value > go

-DB Test
Left click the WAMP's icon > mySQL > mySQL console
Default User name: "root"
Default Password: ""(blank)
Kick commands below.
mysql> use sec_test
mysql> SELECT * FROM cars limit 10;
=>Your test data should be displayed.

3. SEC Setting

- install & kick SEC agent + manager
http://docs.runmyprocess.com/Integration_Guide/SEC/

- install JDBC driver
https://dev.mysql.com/downloads/connector/j/
>Platform independent

- install & kick JDBC adapter
http://docs.runmyprocess.com/Integration_Guide/SEC/Adapters/JDBC
Modify JDBC.config
>sqlSource: jdbc:mysql://localhost:{port of MySQL running on WAMP}/{your database name}?characterEncoding=UTF-8&useSSL=false&serverTimezone={your time zone}
e.g. jdbc:mysql://localhost:3306/sec_test?characterEncoding=UTF-8&useSSL=false&serverTimezone=JST
>sqlDriverPath: the location of the JDBC driver you download
#Access route:
RMP > SEC Agent > Manager > JDBC adapter > JDBC driver > MySQL

4. Test

4.1 run SQL query in manager (POST on the manager, with JSON content, contains SQL query as string)
4.2 run the SEC, then create a new connector > call that connector from RMP
Request body sample

Mobile / RunMyApp

About - Barcode RunMyApp can scan

This document quotes from codegeek provided by Rafal Urbanski

Publish date	09 Jan 2018
Contributor	Rafal Urbans
Reviewer	Thibault Grar
Status	PUBLISHE

Here is a list of barcode standard, which can be scaned by RunMyApp. The app can use a device camera to “scan” the barcode and insert the scanned value to an input field. API description description is on RMP user manual here.

The following barcodes are supported

iOS

- UPC-A
- UPC-E
- Code 39
- Code 39 mod 43
- Code 93
- Code 128
- EAN-8
- EAN-13
- Aztec
- PDF417
- QR

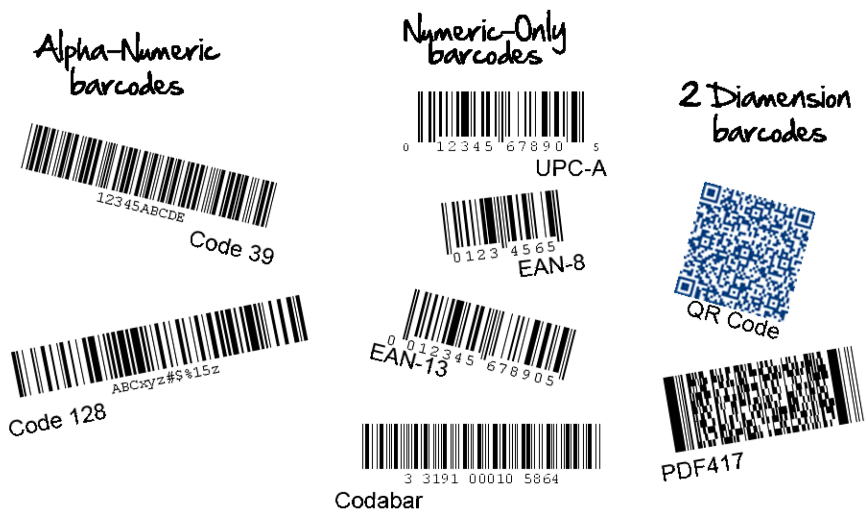
iOS8 onward:

- Interleaved 2 of 5 (ITF)
- ITF14
- DataMatrix

Android devices:

- UPC-A
- UPC-E
- Code 39
- Code 93
- Code 128
- EAN-8
- EAN-13
- PDF417
- QR Code
- ITF
- DataMatrix
- Codabar
- RSS-14

Sample barcodes:



Version	Date	Comment
Current Version (v. 3)	**Jan 23, 2018 10:06**	**Mani**
v. 2	Jan 09, 2018 10:21	**Mani**
v. 1	Jan 09, 2018 10:04	**Mani**

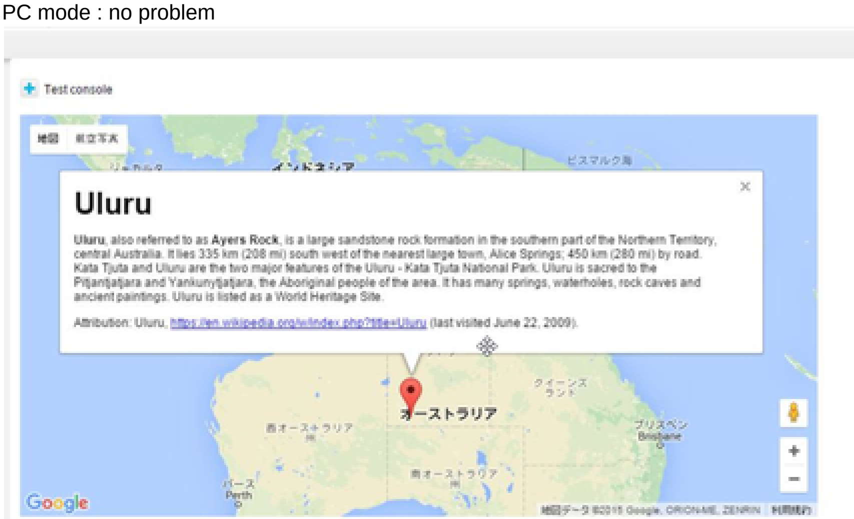
Howto - CSS tips / tricks for mobile

Publish date	26 Feb 2018
Contributor	Sei FUJITA Rafal Urbans
Reviewer	David Courta
Status	PUBLISHE

Use case - Google maps

Mobile app (RunMyApp) has a narrower window than the PC. Using the default mobile CSS, it would still have a problem to display some objects, e.g. a google map. The following document shows how to avoid this problem.

- PC mode : no problem



- Mobile mode : A marker, info window, “+-”button and the yellow person mark for street view are not displayed.

CSS Solution

Default CSS

```
@media only screen{
 table,tr, td, div, tbody {
  max-width: 100%;
 }
}
```

“max-width” restricts the width of objects on google map.

Updated CSS

```
@media only screen{
 table,tr, td, div, tbody {
  max-width: none;
 }
}
```

Please note that this CSS Selector is very "large" , a lot of HTML elements in your application will be affected by this CSS rule. In case of side effects, use HTML ID or CLASS to narrow down the selected elements.

> This resolution is based on the recommended code on the google document
>
> https://developers.google.com/maps/documentation/javascript/examples/infowindow-simple

Use case - Define a max width for a specific column

```
// .ArrayTree-selectPicklist is a example.
// td:nth-child(1) is the first column of a table.
// Change the code to a proper selector based on your app
.ArrayTree-selectPicklist tr.sideslide td:nth-child(1){
    max-width: 50px !important;
}
```

Platform

About - request_id, instance_id, document_id

This is supplementary information of RMP internal parameters.

Publish date	08 Jan 2018
Contributor	Mani
Reviewer	Martial NDOU
Status	PUBLISHE

instance_id

- A instance_id is associated to a **RMP web interface**
- The instance id can be retrieved by using the internal parameter **${entityId}**
- "instance_id" is generated when a web interface is opened

request_id

- A request_id is associated to a **RMP process**
- The request id can be retrieved by using the internal parameter **${P_request}**
- "request_id" is generated when a process is launched

document_id

- A document_id is associated to a **RMP web interface** AND a **RMP process**
- The document id can be retrieved by using the internal parameter **${P_document.id}**
- "document_id" is generated when a process is launched
- In case of a single process, document_id is same as request_id. When there are parallel processes (E.g. a same task is assigned to person A, person B and person C. Either of the person has the right to update the process), each process is associated to one document_id

Process / CAPI

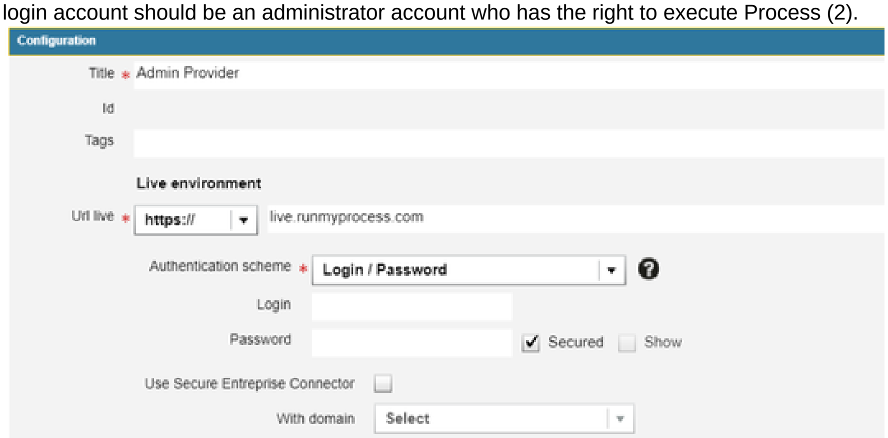
Howto - Connect two processes / CAPI

Publish date	19 Jan 2018
Contributor	Mani
Reviewer	Farva Hussai
Status	PUBLISHE

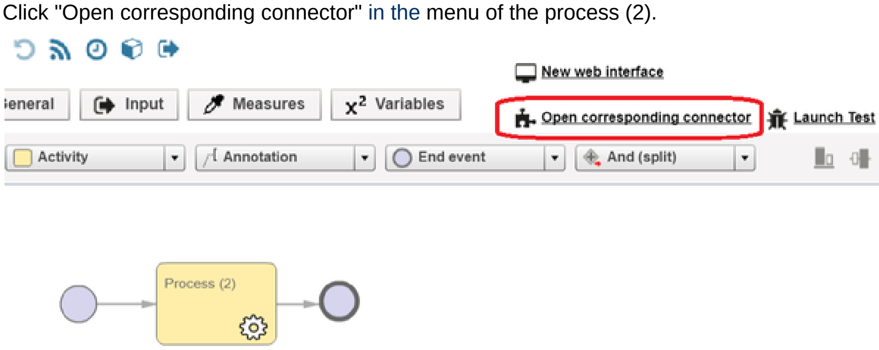
A connector can be used to connect RunMyProcess platform with other systems. It also can connect CAPIs and processes.

Procedure

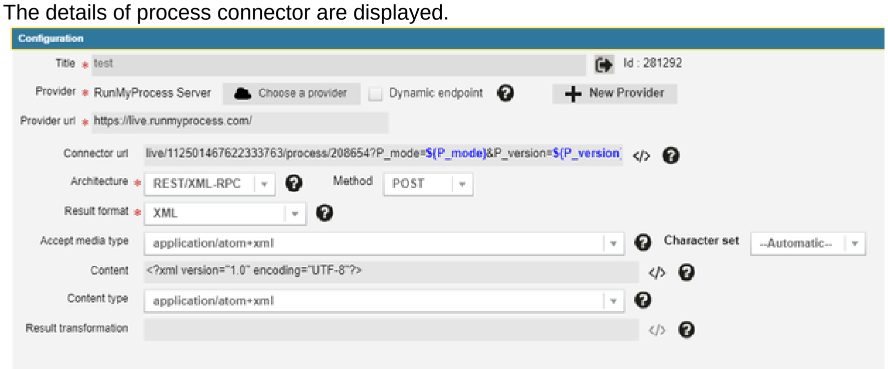
1. Create an Admin Provider.
   As the default "RunMyProcess Server" provider is read-only, you need to create a local provider. The configuration is the same as the default "RunMyProcess Server" provider. The login account should be an administrator account who has the right to execute Process (2).



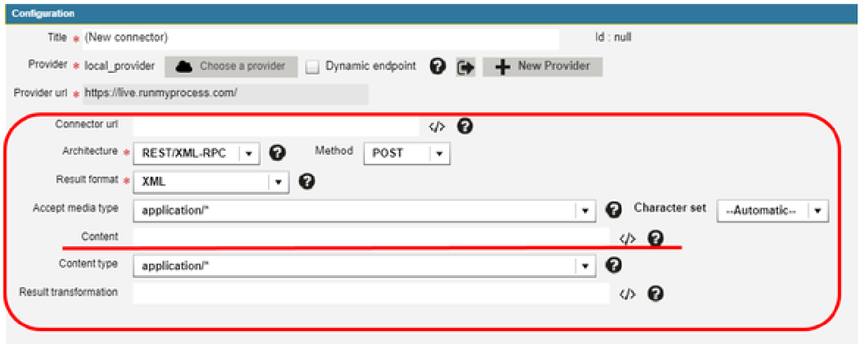
2. Click "Open corresponding connector" in the menu of the process (2).



The details of process connector are displayed.



3. Create a new connector under the Admin Provider. Copy all the configurations from the process(2)'s connector to the new connector.



4. Customize the "Content" in order to pass the parameters from CAPI/process (1) to the process (2).

```
<content type="xml">${input}</content>
```

Sample code of connector:

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
xml:base="https://live.runmyprocess.com/">
  <title>trigger_process</title>
  <link rel="self"
href="live/621049386/process/207735" />
  <id>207735</id>
  <rights>(c) RunMyProcess 2017</rights>
  <updated>2017-11-03T16:14:33Z</updated>
  <entry>
    <title>Parameters</title>
    <category term="initial" />
    <id>207735</id>

<published>2017-11-03T16:14:33Z</published>
    <content type="xml">${input}</content>
  </entry>
</feed>
```

"input" is a **JSON**. It will be sent to "Initialized parameters" when the Process (2) is launched.

5. Attach the connector to Process / CAPI (1).

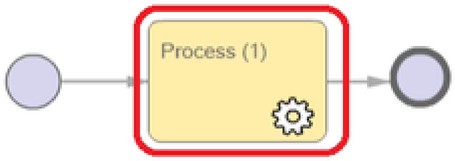
Process (2) is launched from Process or CAPI (1) now.

Note that when the connector connects a **CAPI** (called from a web interface) to a process, you need to check the input data format.

Example:
Code on web interface to start the CAPI

```
var input = {};
input.mydata = a_data_array;
id_my_capi.trigger(input, {},
successCallback, failureCallback);
```

The input data to the CAPI is a JSON: {"data", a_data_array}. In this case, if the connector to the process (2) expects a parameter named ${input} as our example above, but there is no paramters named "input" anymore. Therfore, you should add the following code to the input variables before calling the connector, to reset the parameter name: input, as a json.




```
<#assign input = {}>
<#assign input =
P_json_put(input,"mydata",mydata,false)>
${input}
```

After above operation, the input data is re-defined as ${input}, and it can be passed to the connector and the process (2) now.

Howto - Email Template

Check email provider and describe translator restrictions

Publish date	26 Feb 2018
Contributor	Hugo David E
Reviewer	Reviewer Meeting
Status	PUBLISHE

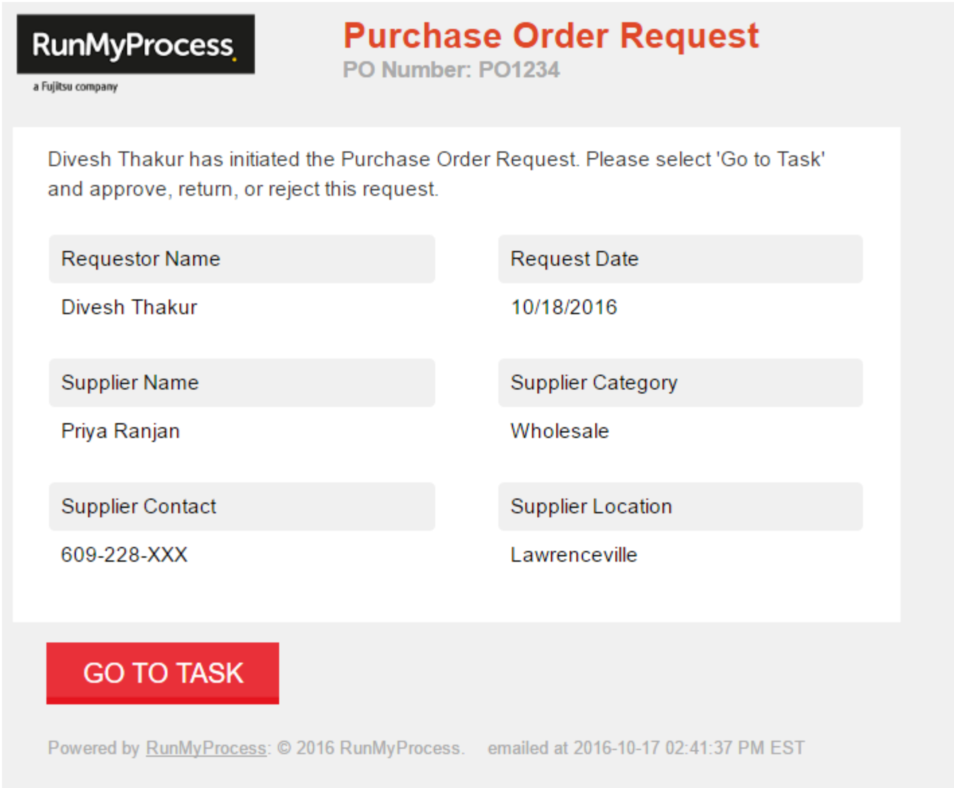
Introduction

This document will explain how to generate HTML email template within RunMyProcess

The email template below works with Gmail but has not been tested on other email services.

The i18n translation feature is compatible with the email template used below.

e.g Users will receive below email format once code is implemented.



Configure the email template and plug it to your workflow

- Design a HTML Template with CSS according to your project template. Include that HTML template as **.ftl**file (email_source.ftl) in your project.

  Output of the plain template without content would be like below screenshot

  *Note: Source code of **email_source.ftl** is attached in a file at the end of this tutorial*

Program will change the below mentioned content in the template

- this_is_app_name
- this_is_request_type_label
- this_is_po_no
- this_is_main_content
- this_task_button
- this_is_time

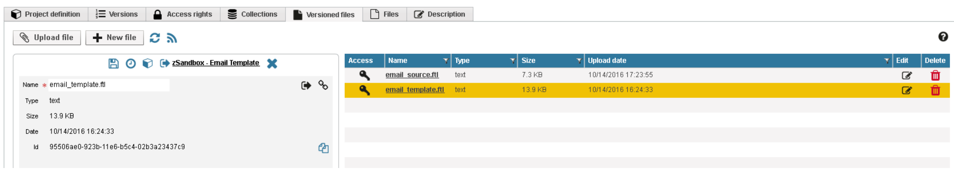
- As displaying in the e.g screenshot this template is having 4 sections
  - Header
  - Content
  - Footer
  - Email Text

All these sections we populate in the template using program and the value will be passed from the process.

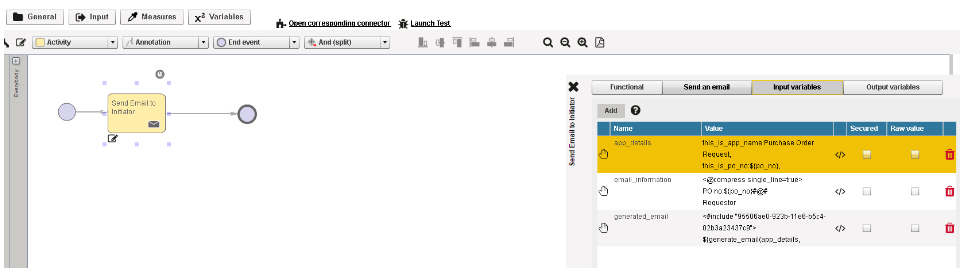
Header , Footer will be passed from process in the **app_details** parameter where as **Content and Email Text** will be passed in the **email_information** parameter.

- How to Integrate email source and Process
  - Create a Project and Include the email provider
  - Create a **.ftl** file (eg. email_template.ftl) with source code to integrate the email template source (email_source.ftl) with process
    *Note: Source code of **email_template.ftl** is attached in a file at the end of this tutorial*

Screenshot showing .ftl files



  - Create a Process with email activity



Source code for all the parameters used in this activity

app_details :

Script

```
this_is_app_name:Purchase Order Request,
this_is_request_type_label: PO Number,
this_is_po_no:PO1234,
this_is_separator:#@#

```

app_details parameter is used to feed all the header and footer variable in the email source.

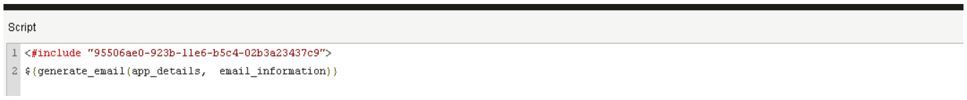
email_information:

Script

```
<@compress single_line=true>
Requestor Name:${requestor_name}#@#
Request Date:${request_date}#@#
Supplier Name:${supplier_name}#@#
Supplier Category:${supplier_category}#@#
Supplier Contact:${supplier_contact}#@#
Supplier Location:${supplier_location}#@#
header_text:${P_initiator.name} has initiated the Purchase Order Request. Please select 'Go to Task' and approve, return, or reject this request.
</@compress>
```

email_information parameter is used to feed the email_content in the email source.

generated email:

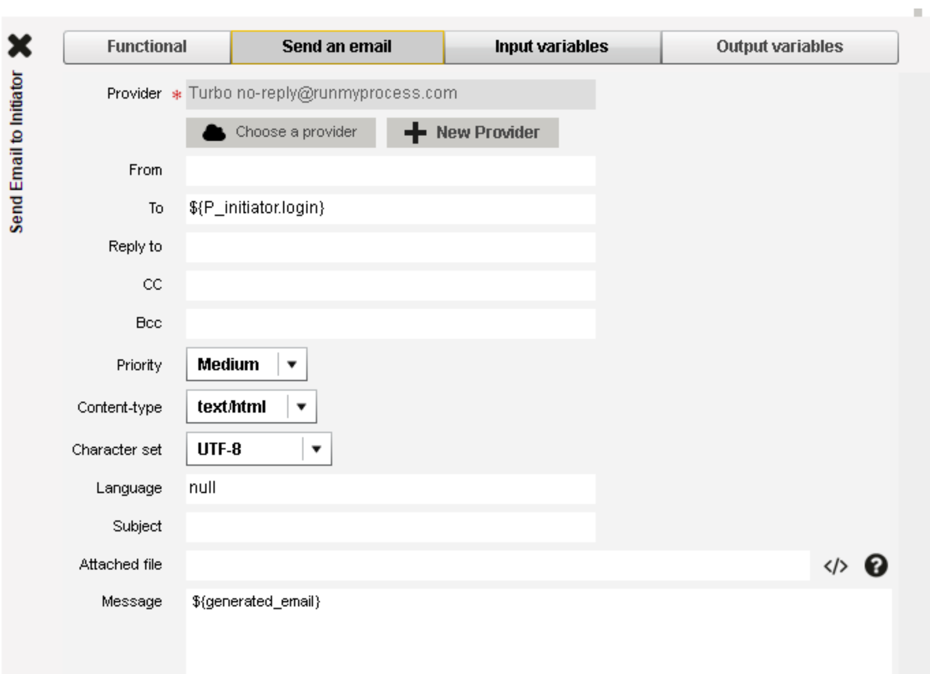


Here we are including the email_template.ftl file and calling a function which is defined there. This function is responsible for all the processing of data which we are passing through *app_details*and*email_information*.

"**95506ae0-923b-11e6-b5c4-02b3a23437c9**" in the above screenshot is the id of the email_template.ftl file.<u>You have to replace this id</u>with the id of your file email_template.ftl uploaded in your project.

In a similar way we are including the id of the email_source.ftl ("**c415bce0-9240-11e6-b973-0619bd984419**") file in the email_template.ftl file.<u>You have to replace this id</u> with the id of your email_source.ftl file uploaded in your project.

Finally the ${*generated_email}* is passed to the email message in the email_configuration tab as seen in the below screen shot.



Congratulations, you can now send nice emails from your RunMyProcess applications!

File	*Modified*
email_source.ftl	*Jan 24, 2018 by Hugo David Buriel-Vasquez*
email_template.ftl	*Jan 24, 2018 by Hugo David Buriel-Vasquez*

Download All

Open email_template

Howto - Reassign a task to another user

Publish date	09 Jan 2018
Contributor	Rafal Urbans
Reviewer	Mani
Status	PUBLISHE

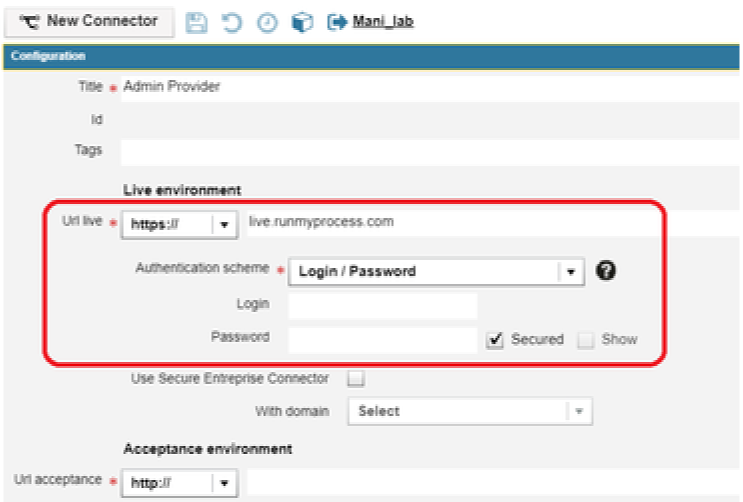
This document partially quotes from codegeek provided by Rafal Urbanski

When a task (process) is assigned to User A, but you need to re-assign it to User B, the normal way is to setup a delegation in advance. If the standard delegation function doesn't fit for your requirements somhow, for example

- the user has not a delegation, but the user account is inactived (the user leaves the organization)
- the user has a delegation, but you want to avoid group delegation (read the related document here)

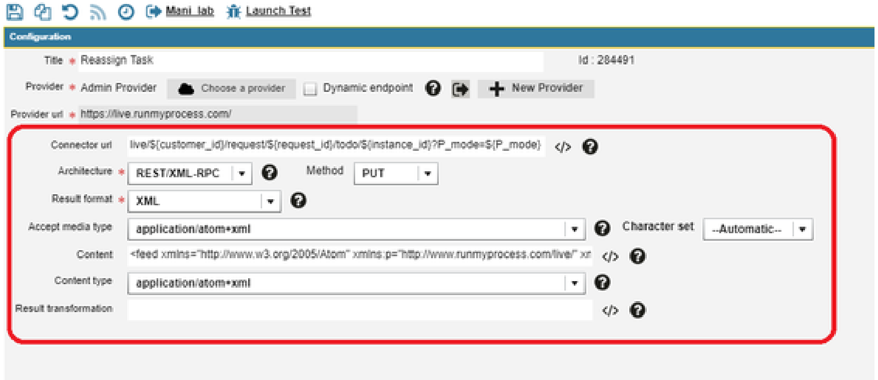
you can use the following way to force re-assign a task from User A to User B with the admin privileges.

1. Create a new Admin Provider



   Url: https://**live.runmyprocess.com/**
   Login / Password: This should be an **administrator** role
   Copy the same configuration to ACCEPTANCE/TEST/LIVE environment

2. Create a new connector under the Admin Provider: Reassign Task



Connector url:

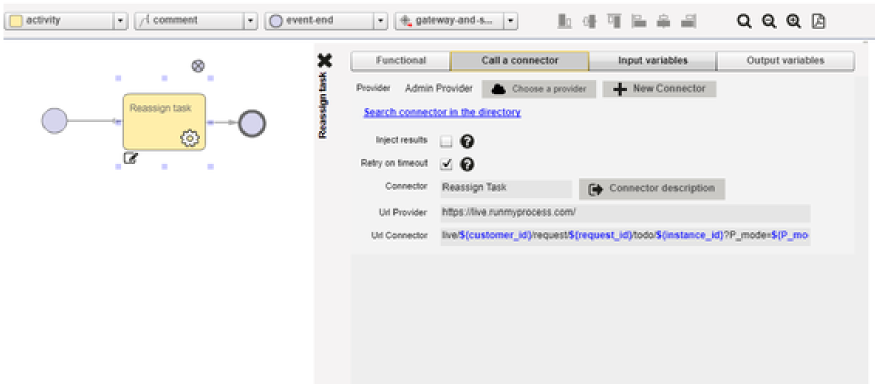
```
live/${customer_id}/request/${request_id}/t
odo/${instance_id}?P_mode=${P_mode}
```

Method: PUT
Accept media type: application/atom+xml
Content:

```
<feed xmlns="http://www.w3.org/2005/Atom"
xmlns:p="http://www.runmyprocess.com/live/"
xml:base="https://live.runmyprocess.com/">
<title></title>
<link rel="self"
href="live/${customer_id}/request/${request
_id}/todo/${instance_id}"/>
<rights>(c) RunMyProcess</rights>
<entry>
<content
src="live/${customer_id}/request/${request_
id}/todo/${instance_id}"/>
<category term="assignedto"
label="${logintm}"/>
</entry>
</feed>
```

Content type: application/atom+xml

3. Create a CAPI to call the connector
   The input variables are as following:
   - customer_id - the account id
   - request_id - the request id of the process
   - instance_id - the instance id of the process
   - logintm – the new email address of the person to whom the task will be re-assigned



Refer About - request_id, instance_id, document_id to understand how to get the input parameters.

4. Trigger the CAPI from a web interface. The selected task can be re-assign to another user.

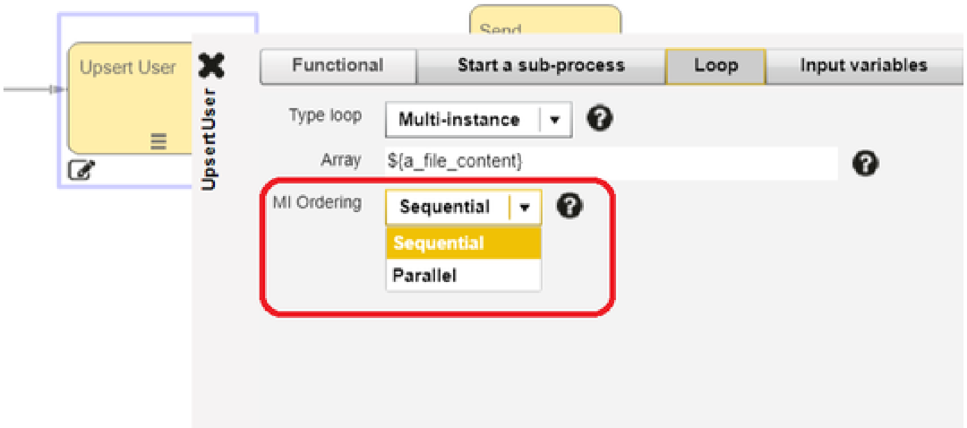
```
var input = {};
input.customer_id = "aaaaa";
//${P_customer}
input.request_id = "aaaaa"; //${P_request}
input.instance_id = "aaaaa"; //${entityId}
input.logintm = "aaa@aaa.com"; //the person
you want to reassign the task

id_reassign_capi.trigger(input,{},successCa
llback,failureCallback);
```

Howto - Synchronize multiple asynchronized processes

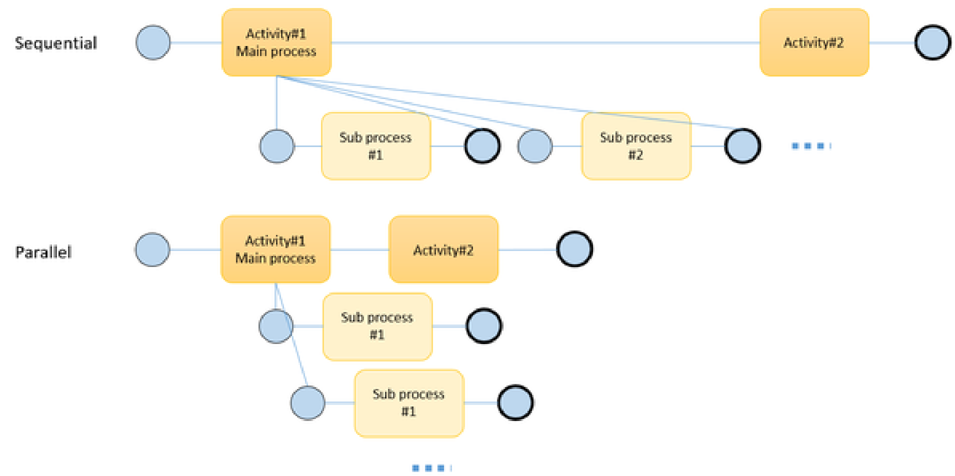
Publish date	05 Jan 2018
Contributor	Mani
Reviewer	Richard Mang Thibault Grar
Status	PUBLISHE

Problem

When you have a main process, triggering multple child processes, you can define them as sub-process. Sub-process can be started sequential or parallel (see RMP doc).



The difference of sequential and parallel sub-process is whether they are synchronized or not.



Parallel way has better performance, however, it is not synchronized. In some cases, e.g. when the sub process number is big, or the activity is heavy, the following way can simulate a main process sub process structure, but with better performance AND synchronized result.

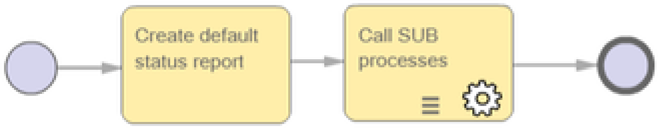
Solution

In the following steps, we call the main process as "father process", and the multiple sub-processes as "child processes". Instead of defining sub-process, we define the father process and child process connected by a connector. An action is executed only when ALL child processes complete, for example, send a notification email, update data on SAP, etc.

father process connector child process (loop)

1. Create a collection
2. Create a default report before starting the child processes. The collection should include the following properties:
   - A unique report id (e.g a timestamp)
   - A unique id for identifying the child processes (e.g login mail address)
   - A field for specifying the child process status: NOT_START, SUCCESS, FAIL, etc

father process:



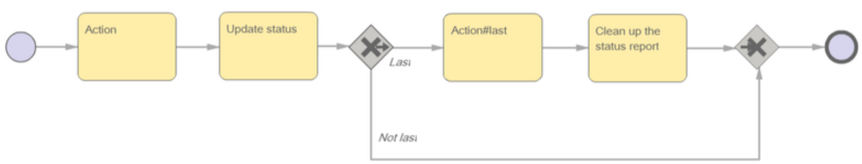
Sample code:

```
<#list data as s>
    <#assign obj = {}>
    <#assign obj =
P_json_put(obj,"report_id",report_id,true)>
    <#assign obj =
P_json_put(obj,"email",s.email,true)>
    ....
    <#assign obj =
P_json_put(obj,"status","NOT_START",true)>

${save_object(obj,"process_status_report")}
</#list>
```

3. Refer Howto - Connect two processes / CAPI to define a connector bewteen father process and child process. Call the connector "parallel".

4. Inside of your child process, update the status in all the child processes, e.g. "DONE". At the end of the child process, check the count of "NOT_START". If this is the last child-process, then execute the last action you want to execute (sending a notification email, etc.) and delete the status report associated to the current report_id.

child process:



Sample code (check if this is the last child process)

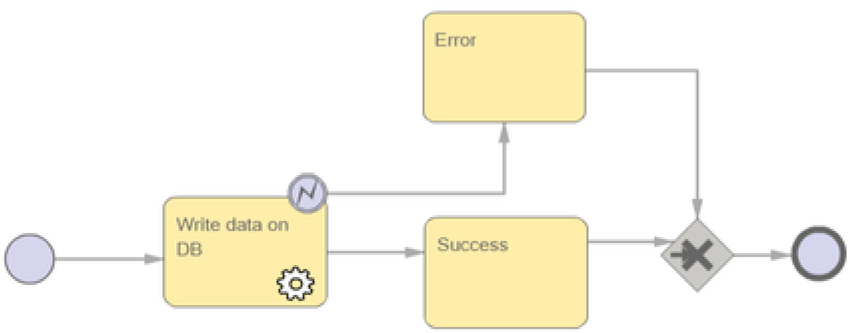
```
<#assign query1 = {}>
<#assign query1 =
P_json_put(query1,"report_id",report_id,tru
e)>
<#assign query1 =
P_json_put(query1,"status","NOT_START",true
)>
<#assign result1 = list_objects( query1,
"process_status_report" )>
<#if result1?size gt 0 >
no
<#else>
yes
</#if>
```

Howto - Thread pool to leverage traffic

Publish date	09 Jan 2018
Contributor	Mani
Reviewer	Thibault Grar
Status	PUBLISHE

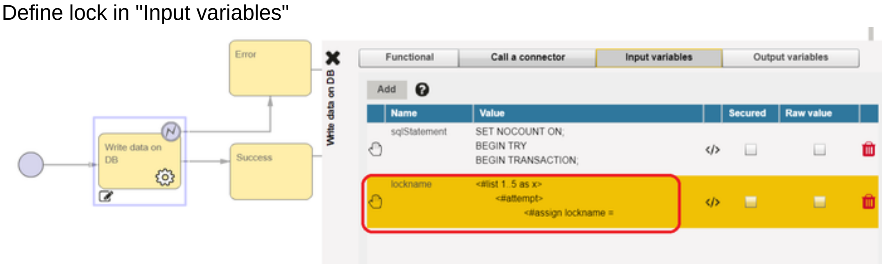
If you have an activity which requires long time execution or many server resources in the process, for example, using SEC to communicate an on-premise system, long calculation for a huge collection, etc, the long time activity may generate a bottleneck when the number of access is too many. In that case, you can consider to use a thread pool design pattern to avoid the prespected bottleneck.

In RunMyProcess, lock / unlock are the API for creating a thread pool in backend.



In above example, "Write data on DB" calls a connector, which accesses to a database on an on-premise system through SEC. When the process is called parallelly, the connector may have a bottleneck, due to any network issue. to resolve the problem, you can try to use the following procedure to create a thread pool and leverage the traffic.

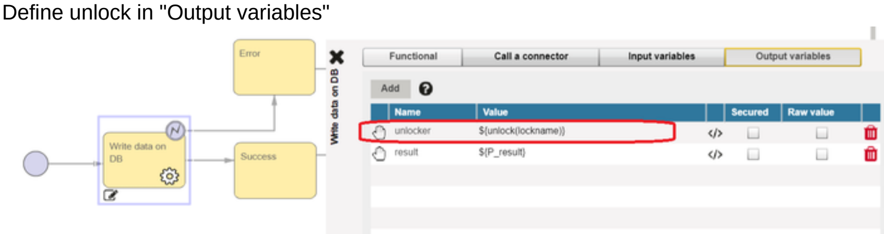
1. Define lock in "Input variables"



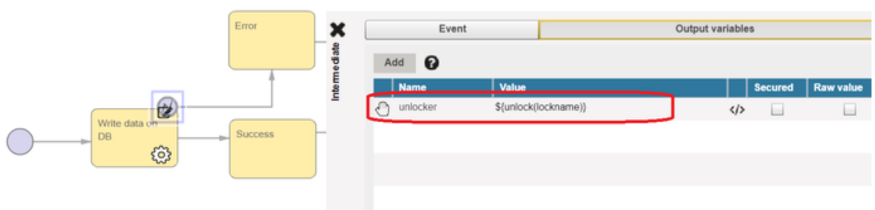
Sample code

```
<#list 1..5 as x>
 <#attempt>
  <#assign lockname =
"A5DB_${timestamp()?number % 5}">
  ${lock(lockname, 40000)}
  <#break>
 <#recover>
  <#assign lockname = "DUMMY">
 </#attempt>
</#list>
${lockname}
```

2. Define unlock in "Output variables"



3. Define unlock in error catch



- The thread pool in this example size is 5. It means max to 5 connectors can be executed at the same time, and the other requests will wait
- Waiting time is set to max to 40 seconds. Once the connector execution finishes, the lockname is unlocked, and the same name lock would be used by other requests
- If a request can not get any available lock name after 5 tries, it will call the connector directly
- **Put the lock as the final in-activity script before the action, and put the unlock at the first line after the action, in order to minimize the lock time duration**

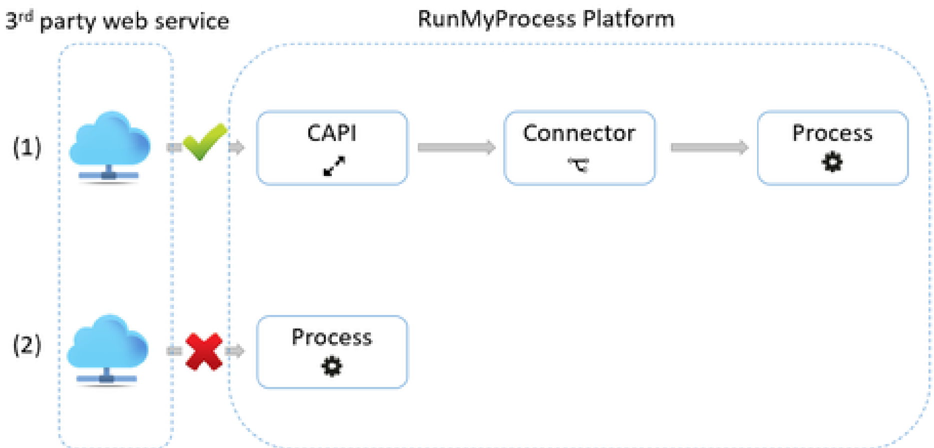
Contact RunMyProcess if you need to do a performance tuning of your application.

Howto - Trigger a process by an external web service

Here is supplementary information of trigger a process by web service on developer guide.

You can start a process directly by an external web service through HTTP request. However, it is recommended to trigger the process from a CAPI and its connector. The reason is that ...

Publish date	05 Jan 2018



Contributor	Mani
Reviewer	Richard Mang
Status	PUBLISHE

1. Refer to Howto - Connect two processes / CAPI to connect CAPI and the process.
2. Edit your HTTP request and define the input parameter using JSON format.

```
exports.handler = function(context, event,
callback) {
 var request = require('request');

 //Define HTTP request
 var opts = {
     url:
'https://live.runmyprocess.com/live/621049386/h
ost/16417/service/280134?P_mode=ACCEPTANCE',
     method: 'POST',
     json: true,
     body: {
         'input' : {
       'key1': value1,
       'key2': value2
         }
     },
     headers: {
      'Accept':'application/json',
      'Content-Type': 'application/json',
   'Authorization':'Basic
dHdpbGlvQHJ1bm15cHJvY2Vzcy5jb206NFVCS1pnSjY='
     }
 };

 // Execute HTTP Request
 request(opts, function (error, response, body)
{
  callback(null);
 });
}
```

{input: {"key1": value1, "key2":value2}} will be sent to "Initialized parameters" when the process is executed.

Howto - Trigger a process on a specific day per month

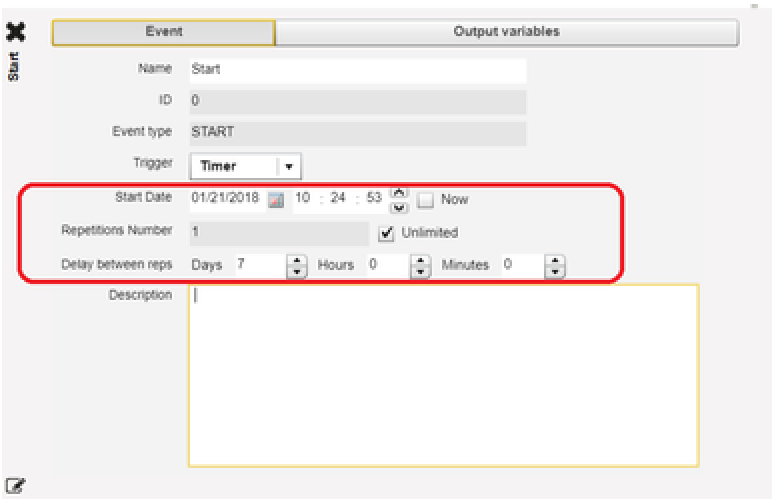
This document partially quotes from codegeek provided by Rafal Urbanski

RunMyProcess can define a scheduled process starting

- every X days
- every X hours
- every X minutes

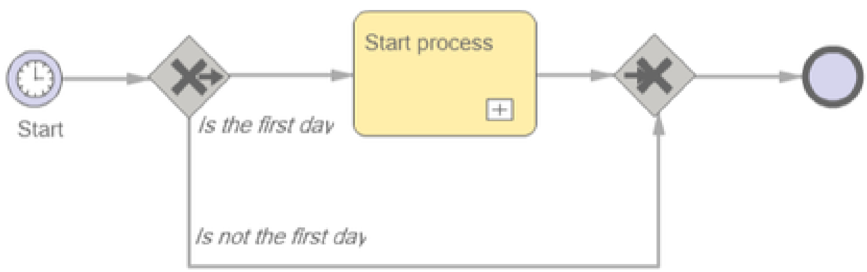
however, it is not straightforward to define a specific day per month (e.g. every 1st day of a month).

Publish date	17 Jan 2018
Contributor	Rafal Urbans Mani
Reviewer	Farva Hussai
Status	PUBLISHE



The following workaround shows how to trigger a process in the first day of a month.

Create a new process. The process is defined to be executed **everyday**. In the first gateway, check if it is the first day of a month. If it is the first day, then trigger your actual process as a sub-process. If it is not the first day, then do nothing.



Here is an example how to check if it is the first day of a month

```
${now("dd")} == 1
```

Howto - Use a prefix in email notifications to show which mode we're in

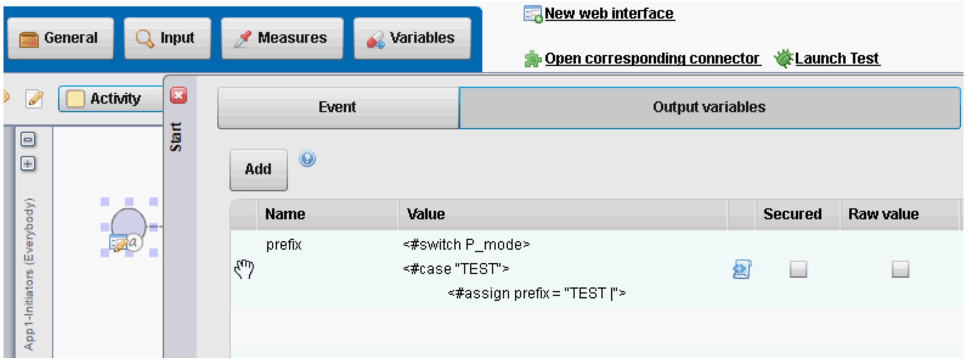
Publish date	04 May 2018
Contributor	Hugo David E
Reviewer	Reviewer Meeting
Status	PUBLISHE

Context

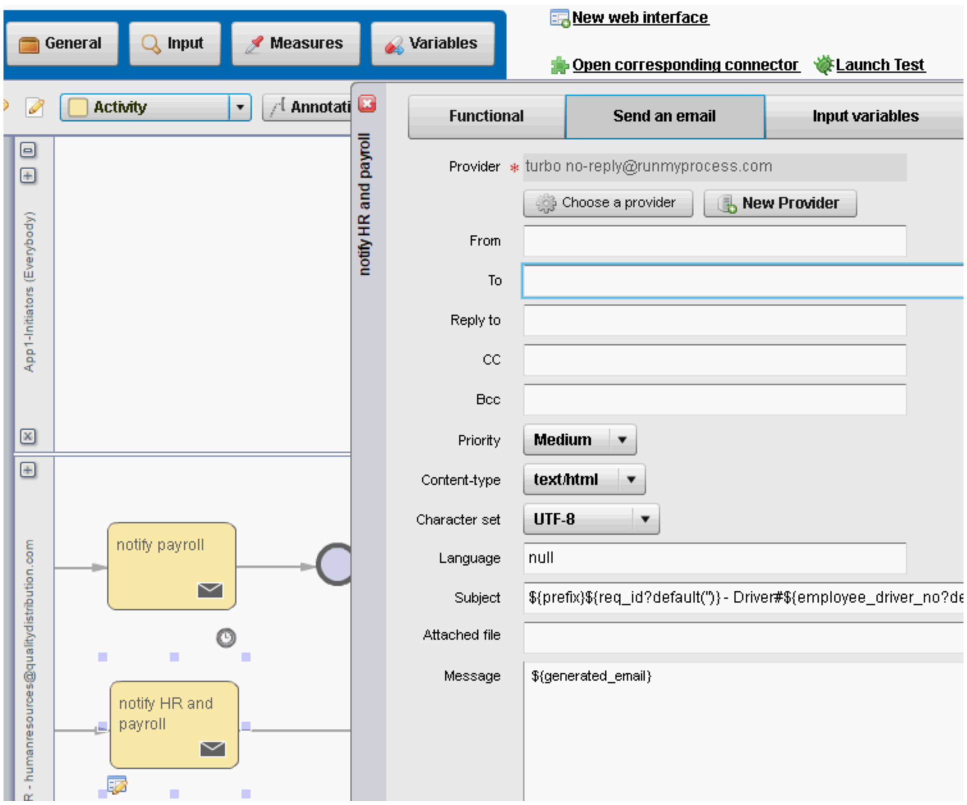
When you send emails to end users, those may be TEST, UAT(ACCEPTANCE) or LIVE notifications. To help end users (and RMP developers) differentiate those emails, use a specific prefix in your email.

How

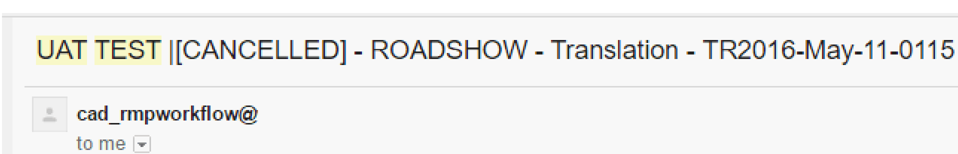
In the entry point of your process, create a new variable prefix with the following FM code:



Then use that variable as prefix in the subject of each email notification:



Your acceptance emails will now show up like this:



Done!

Provider / Connector

- Howto - Build a connector

Howto - Build a connector

The Best practices to be followed to build the connector

File	Modified
Building RMP Connector_ Bestpractices.pdf	Dec 21, 2017 by Satish Jadhav

Drag and drop to upload or browse for files

Publish date	28 Dec 2017
Contributor	Satish Jadha
Reviewer	Mani
Status	PUBLISHE

Role / User

- Howto - Avoid force group delegation

Howto - Avoid force group delegation

n the following use case

- You assign a manual task to a user or a user group
- delete the user when there is a un-processed manual task

the opened task will be automatically re-assign to the whole user group (force group delegation) ~~no matter the user has a delegation or not. (Known issue: Gitlab server 450)~~

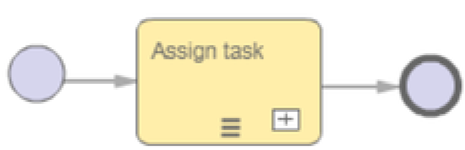
Publish date	13 Feb 2018
Contributor	Mani
Reviewer	Richard Mang
Status	PUBLISHE

To avoid this behavior, you can

- Define a delegation to the user in advance (before starting the process)
- Use "Runtime Lanes".
  Note that the runtime user is not inherited when using sub-process.

Example:

MAIN process



SUB process

Refer to P_add_user_to_lane and P_remove_user_from_lane for the usage.

Web interface

- Dashboard
- Report
- UI / Widget

Dashboard

Report

Howto - Add pagination to a javascript report

Procedure - Modify the javascript report for data loading

Publish date	05 Jan 2018
Contributor	Martial NDOU
Reviewer	Mani
Status	PUBLISHE

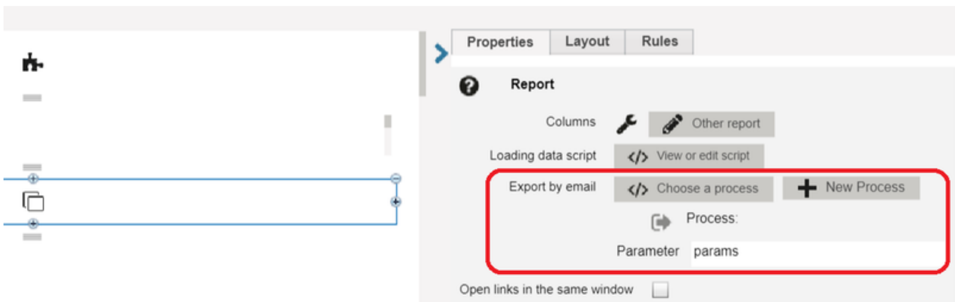
Modify the default sample javascript code (Javascript Report - 3. Set the script, the visible items max, and the identifier)

```
function successCallback(P_computed){
 var reportOptions = {
  // use RMP_Report.DYNAMIC_COUNT here
  count:RMP_Report.DYNAMIC_COUNT,
  // pagination index (mandatory)
  first:P_first,

  //A text that may be displayed in the report
pager if data items' count is not known
(optional)
  pagerCount: "xxx"
 };

 // P_computed.data is a json array that should
have the same structure configured in your CAPI
 id_report.setData( P_computed.data,
reportOptions );
 id_report.setLoading(false);
}

function failureCallback(P_error){
 alert(JSON.stringify(P_error));id_report.setLo
ading(false);
}

// the input object is input parameters used by
CAPI. Default parameters
var input = {};
input.pageSize = 20;
input.first = P_first;
input.sortedColumns =
id_report.getSortedColumns();
input.currentFilters = id_report.getFilters();
```

```
//required parameters for this use case (get
user lane)
var idLane = 123456;
var idPool = 654321;
input.lane_id = idLane;
input.pool_id = idPool;

//When use "Export by email" process, this is
the way to pass parameters to the process
var params = {};
params.lane_id = idLane;
params.pool_id = idPool;

var orgaName = "abc"
var laneName = "def"
params.pool_name = orgaName;
params.lane_name = laneName;
RMPApplication.set("params",params);

//trigger the CAPI
```

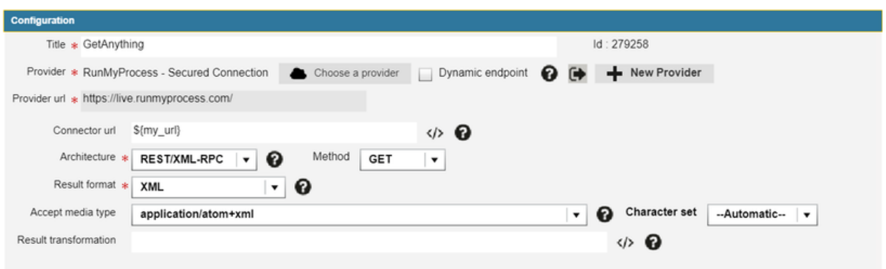
```
id_report_users.setLoading(true);
id_capi.trigger(input,{},successCallback,failureCallback);
```

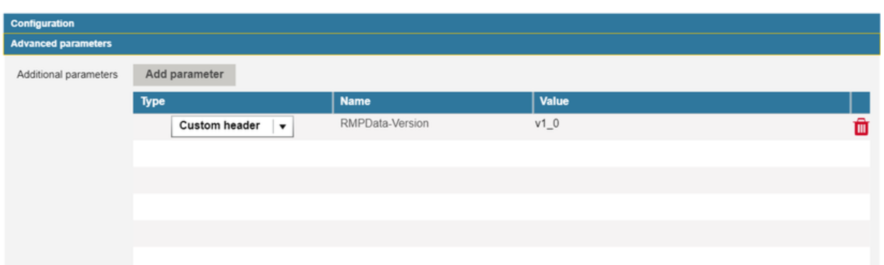
"Export by email" process refers to the following setting



Procedure - Define a "Get Anything" connector

1. Create a "Get Anything" connector under "RunMyProcess Secured Connection" provider
2. Define the connector paramters as following

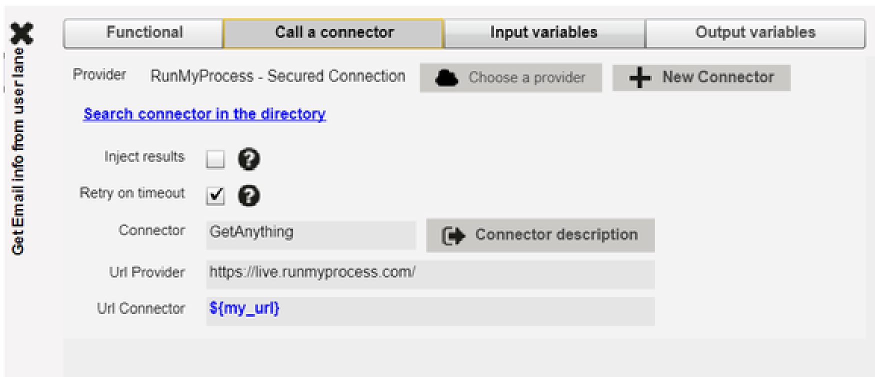




Procedure - Create a CAPI to get the data with pagination

1. Define a CAPI to get the data.

2. Set the function to be a "connector" type. It calls the GetAnything connector above



3. Define input variable to collect the required data range from the connector

Name: my_url
Value:

Modify {account_id} to the actual customer id.

```
<#–Define the base query URL  -->
<#assign base =
"config/{account_id}/pool/"+pool_id+"/lane/
"+lane_id+"/user/?first="+first+"&nb="+page
Size+"&filter=STATUS&operator=EE&value=ACTI
VE&nb=20&orderby=NAME&order=asc">

<#–Add filter conditions, when user selects
any filter on the web interface -->
<#assign filters = "">
<#assign operators = "">
<#assign values = "">
<#list currentFilters as f>
        <#if f_index == 0>
             <#assign filters = "&filter="
+ f.filter>
             <#assign operators =
"&operator=" + f.operator>
            <#assign values = "&value=" +
f.value>
        <#else>
             <#assign filters = filters +
"%20" + f.filter>
             <#assign operators = operators
+ "%20" + f.operator>
            <#assign values = values +
"%20" + f.value>
        </#if>
</#list>
<#assign request_url =
base+filters+operators+values>
${request_url}
```

4. Define the output variables to pass the data to the javascript. The following code is the case to get users from a specified user lane.

   Name: Empty
   Value:

```
<#assign hub = {}>
<#assign orga = []>

<#if P_result.feed.entry??>
    <#assign rs = P_result.feed.entry>
    <#if rs?is_enumerable>
        <#list rs as r>
            <#assign org = {}>
            <#assign org =
P_json_put(org,"name",r.title,true)>
            <#assign org =
P_json_put(org,"id",r.id,true)>
            <#list r.category as c>
                <#if c.@term == "profile">
                    <#assign org =
P_json_put(org,"profile",c.@label,true)>
                </#if>
            </#list>

            <#assign org =
P_json_put(org,"login",r.author.email,true)
>
            <#assign orga = orga + [org]>
        </#list>
    <#else>
        <#assign org = {}>
        <#assign org =
P_json_put(org,"name",rs.title,true)>
        <#assign org =
P_json_put(org,"id",rs.id,true)>
        <#assign org =
P_json_put(org,"login",rs.author.email,true
)>

        <#list rs.category as c>
            <#if c.@term == "profile">
                <#assign org =
P_json_put(org,"profile",c.@label,true)>
            </#if>
        </#list>

        <#assign orga = orga + [org]>
    </#if>
</#if>

<#assign hub = P_json_put(hub,"data",orga)>
${hub}
```

You should now be able to see your data on the javascript report, with the available properties of "name", "id", "login", "profile", and the pagination.

Howto - Make movable rows in report widget

This document quotes from codegeek provided by Rafal Urbanski

Publish date	05 Jan 2018
Contributor	Rafal Urbans
Reviewer	David Courta
Status	PUBLISHE

Sometimes it is required to give user ability to rearrange the order of data in the report widget. In this article I will describe how to achieve this.

In order to make movable rows in report widget, place and configure your report widget on the form. Then add a JS widget and paste the following code:

```
$(document).ready(function() {
 setTimeout(function(){
  var fixHelperModified = function(e, tr) {
      var $originals = tr.children();
      var $helper = tr.clone();
      $helper.children().each(function(index) {

$(this).width($originals.eq(index).width())
      });
      return $helper;
  },
      updateIndex = function(e, ui) {
          $('td.index',
ui.item.parent()).each(function (i) {
              $(this).html(i + 1);
          });
      };

  jQuery(".TreeLeafItem tbody").sortable({
      helper: fixHelperModified,
      stop: updateIndex
  }).disableSelection();
 },5000);
});
```

Please notice that there is a 5 sec time out set in the function, this is to force the code to be executed after the report is loaded. There could be a better way to detect a data load to the report, i.e. using JavaScript to populate the report where we could also trigger the solution.

Note that above method only changes the array display order **visually**. The data order is not saved on the server side. When the page is refreshed, or change the page, it returns to the initial status.

Howto - Sort result in a report widget

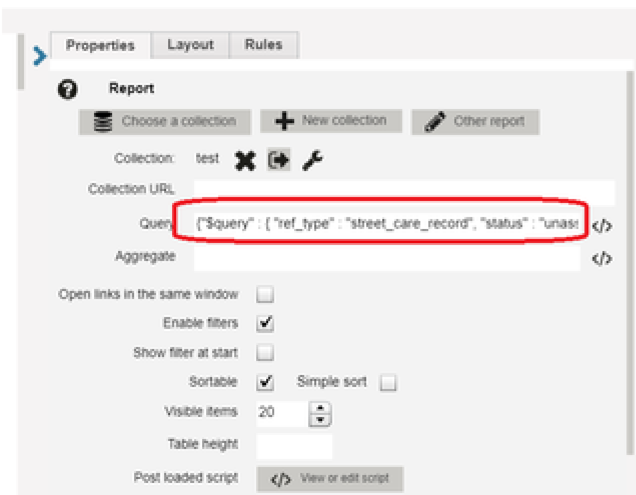
This document quotes from codegeek provided by Rafal Urbanski

Publish date	05 Jan 2018
Contributor	Rafal Urbans
Reviewer	Mani
Status	PUBLISHEI

Sometimes we are required to sort the results displayed in the report widget by the particular column. In this article, I will show you a sample query, which automatically sort the data by the values in the given column.

So, let’s imagine that we have the following columns in our report: **ref_type**, **status** and **PriorityCode** . Now the requirements say that we need to display the results for a particular ref_type value and status, sorted by **PriorityCode**. In order to achieve this we need to put the following query in the QUERY field of the report widget:

```
{
 "$query" : {
  "ref_type" : "street_care_record",
  "status" : "unassigned"
 },
 "$orderby" : {
  "PriorityCode" : 1
 }
}
```



You can refer to more query samples Query and Aggregation

UI / Widget

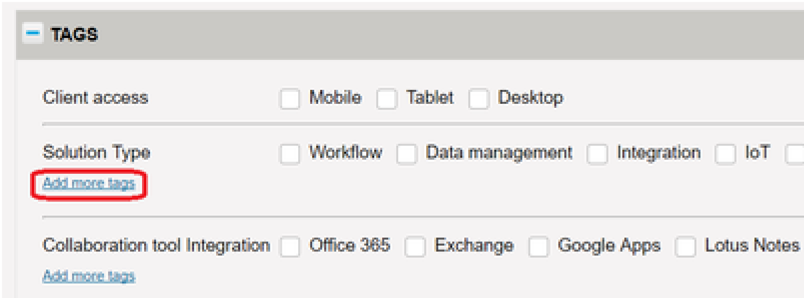
- Howto - Change visual type of a widget
- Howto - Create a modal dialog using Custom Widget
- Howto - Deal with P_index in an array widget
- Howto - Duplicate an existing web interface instance
- Howto - Google address autocomplete
- Howto - Notification Banner

Howto - Change visual type of a widget

Case1: Show a button as a link

Publish date	05 Jan 2018
Contributor	Mani

Purpose: Your widget is a button. You want to display it as a link. In the following case, "Add more tags" widget is a button.

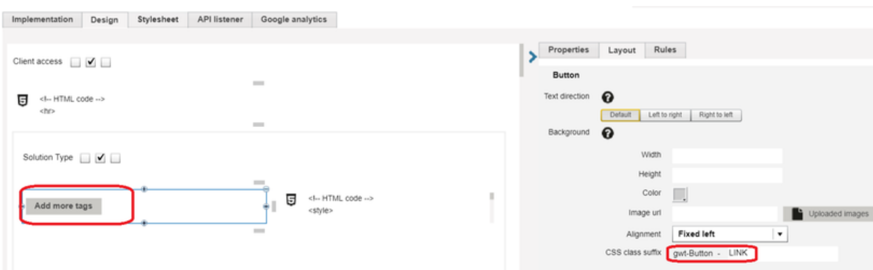


Reviewer	David Courta
Status	PUBLISHE

- Case1: Show a button as a link
- Case2: Show a link as a button

How to do

1. Create a button on the web interface. Define the CSS class as a customized class.

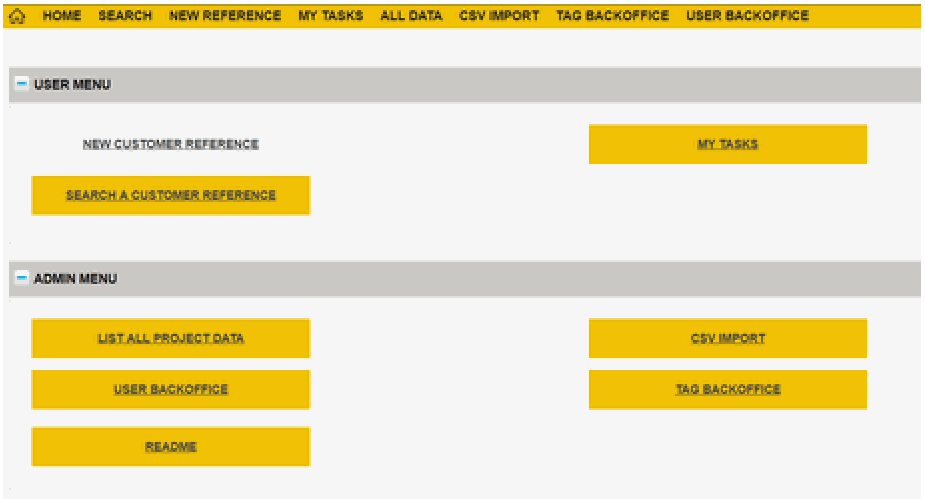


2. Add gwt-Button-LINK class into your CSS file

```
.gwt-Button-LINK {
    background: none!important;
    border: none;
    padding: 0!important;
    font-family: arial,sans-serif;
    color: #069;
    text-decoration: underline;
    cursor: pointer;
}
```

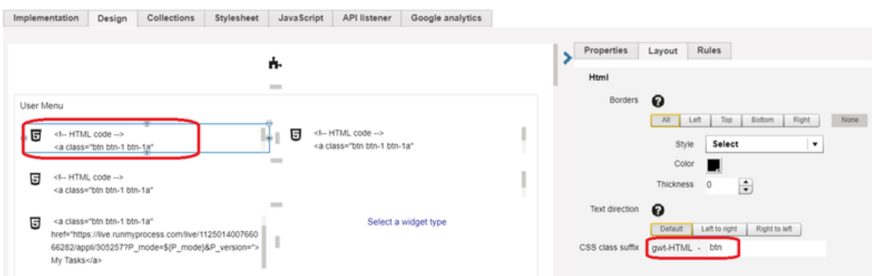
Case2: Show a link as a button

Purpose: Your widget is a HTML widget, you want to display it as a button (with mouse hover). In the following cases, all 8 menus are HTML widgets.



How to do

1. Create a HTML widget on the web interface. Change the CSS class suffix to "**gwt-HTML-btn**
".



2. Edit the HTML widget

```
<!-- HTML code -->
<a class="btn btn-1 btn-1a"
href="appli/123456?P_mode=${P_mode}&P_version=">Menu_name</a>
```

"href" is the linked application web interface URL.

3. Add the following code to your CSS file

```
/* HTML button */
.btn {
 border: none;
 width:308px;
 font-family: inherit;
 font-size: 14px;
 color: inherit;
 background: #F0C105;
 cursor: pointer;
 padding: 15px 10px;
 display: inline-block;
 text-transform: uppercase;
 text-align: center;
 font-weight: 700;
 outline: none;
 position: relative;
 -webkit-transition: all 0.3s;
 -moz-transition: all 0.3s;
 transition: all 0.3s;
}

.btn:after {
 content: '';
 position: absolute;
 z-index: -1;
 -webkit-transition: all 0.3s;
 -moz-transition: all 0.3s;
 transition: all 0.3s;
}

/* Button 1 */
.btn-1 {
 border: 3px solid #fff;
 color: #414141;
}

/* Button 1a */
.btn-1a:hover,
.btn-1a:active {
 color: #414141;
 background: #fff;
}
```

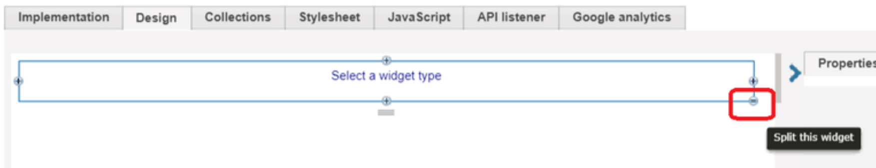
Howto - Create a modal dialog using Custom Widget

A modal dialog is a window that forces the user to interact with it before they can go back to use the parent application. In RunMyProcess, a modal dialog can be created using custom widget.

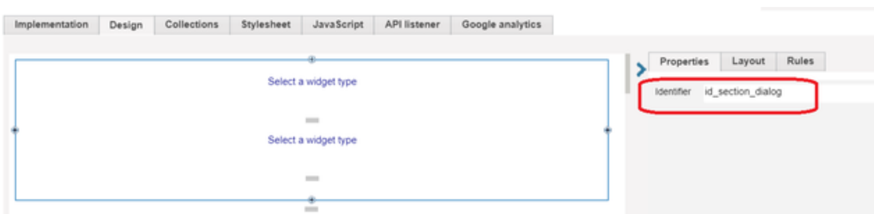
Publish date	05 Jan 2018
Contributor	Mani
Reviewer	Farva Hussai
Status	PUBLISHE

Split widget

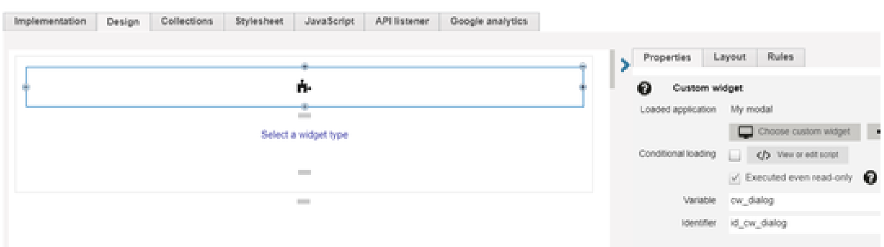
1. Create a split widget



2. Define an identifier for the split widget, and set it invisible.



3. Add a customer widget



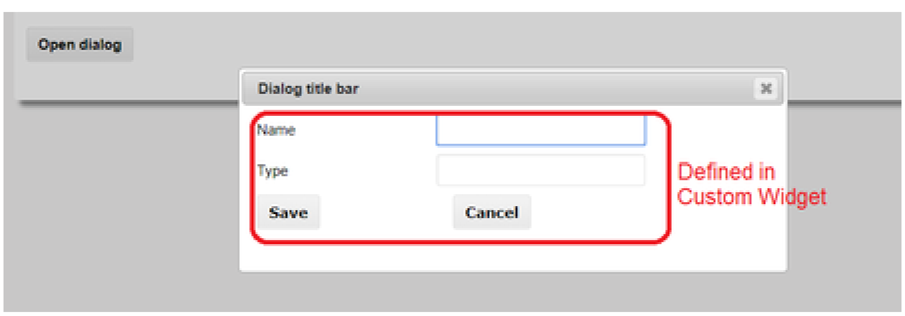
4. Design your dialog inside of the customer widget

Javascript code

Add the following javascript code. Load the JS when you started the web interface.

```
function open_my_dialog() {

  var windowComp;
    function openWindow(){
        var id = "id_section_dialog";
            windowComp =
$('[id="'+id+'"]').dialog({
            autoOpen: false,
            modal: true,
            stack: false,
            width: 450,
            height: 300,
            title: "Dialog title bar",
            open: function(){

jQuery('.ui-widget-overlay').bind('click',funct
ion(){
                    console.log("mouse click
out of the dialog event");

jQuery('[id="'+id+'"]').dialog('close');
                });
            }
        });
        $(windowComp).dialog('open');
        id_section_dialog.setVisible(true);
    }
    openWindow();
}
```

You have a modal dialog now. Click outside of the dialog, the dialog is closed

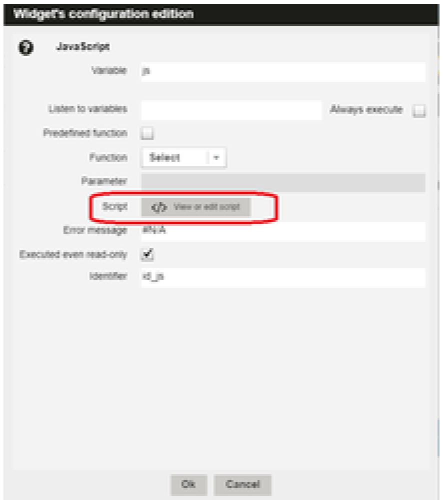


Howto - Deal with P_index in an array widget

P_index is an internal parameter which can be used as an index in a task loop, or a row number in an Array widget. For the array case, when a user removes a row from the middle of the array, the P_index is not updated dynamically. The following is a workaround to handle with row index in javascript, instead of using P_index.

Publish date	05 Jan 2018
Contributor	Mani
Reviewer	Farva Hussai

Define a JavaScript column in the array (invisible)



Status	PUBLISHE

Define the script

Use JQuery to catch the change event, and calculate the actual row index from the event. The selector can be an item where you want to catch the change event.

```
$("#id_my_array\\.id_col1_"+P_index).change(fun
ction() {
   //"this" is the event when you create/remove
a row
   var iIndex = this.id.lastIndexOf("_");
   var actual_index = this.id.substring(iIndex
+ 1, this.id.length);
   //you can check how P_index differs from the
actual row number
   console.log("P_index = " + P_index + "
actual index = " + actual_index);
});
```

Howto - Duplicate an existing web interface instance

This operation is usually done from a web interface report. You need to have a column (measure) containing the instance identifier of the original request. The following function (duplicate) is the function that will duplicate the form:

Publish date	05 Jan 2018
Contributor	Richard Mang
Reviewer	Richard Mang
Status	PUBLISHE

Javascript

```
/*
The following function will trigger a CAPI that
```

```
will return all the variables and values of the
original instance.
*/
function duplicate(instanceId){
 var input = {
  "instanceId" : instanceId
 };
 var options = {};
 id_of_your_capi.trigger(input, options,
 function (result) {
  /* OK */
  var blacklist = {
  "variable_to_not_copy1" : "",
  "variable_to_not_copy2" : ""
 };
 inject_json(result.j_params, blacklist);
 }, function (error) {
  /* KO */
  alert("Error while duplicating the request");
 });
}

/*
The following function allows to specify the
elements you don't want to copy!
*/
function inject_json(my_json, blacklist) {
 for (var i in my_json) {
  if (blacklist[i] == undefined) {
   var special_value = false;
   if (typeof(my_json[i]) == "object") {
   //if my_json[i] is an array or json

    if (my_json[i].length != undefined) {
    //my_json[i] is an array

     if (my_json[i].length == 0) {
      //empty array []
      RMPApplication.setVariable(i, "[]");
      special_value = true;
     }
    } else {
     //my_json[i] is a json
     var nb_keys = 0;
     for (var j in my_json[i]) {
      nb_keys++;
     }
     if (nb_keys == 0) {
      //empty json {}
      RMPApplication.setVariable(i, "{}");
```

```
          special_value = true;
         }
        }
       }

       if (!special_value) {
        RMPApplication.setVariable(i, my_json[i]);
       }
      }
     }
    }
```

CAPI

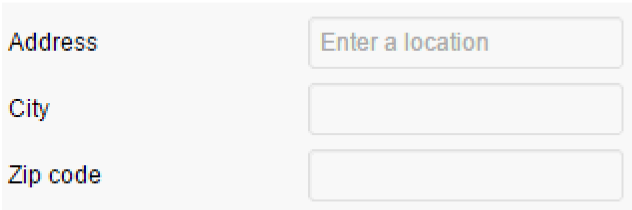
Inside the CAPI, you should call a connector that returns the instance information: The result of this connector should be stored inside a variable called 'result'. Below, you will find Freemarker code that allows you to get this 'result'.

```
<#function get_array my_father my_son>
<#if my_father?is_hash>
 <#if my_father[my_son]?exists>
  <#if my_father[my_son]?is_sequence>
   <#assign my_array = my_father[my_son]>
  <#else>
   <#assign my_array =
[my_father[my_son]?eval]>
  </#if>
 <#else>
  <#assign my_array = []>
 </#if>
<#else>
 <#assign my_array = []>
</#if>
<#return my_array>
</#function>

<#assign entries =
get_array(P_result.feed,"entry")>
<#assign j_params = {}>

<#list entries as x >
 <#if x.category.@term = "initial">
  <#assign j_params_initial =
x.content.P_value>
 <#elseif x.category.@term = "computed">
  <#assign j_params_computed =
x.content.P_value>
 </#if>
</#list>

<#assign j_params = j_params_initial>
<#list j_params_computed?keys as x>
 <#assign j_params =
P_json_put(j_params,x,j_params_computed[x],true
)>
</#list>
${j_params}
```

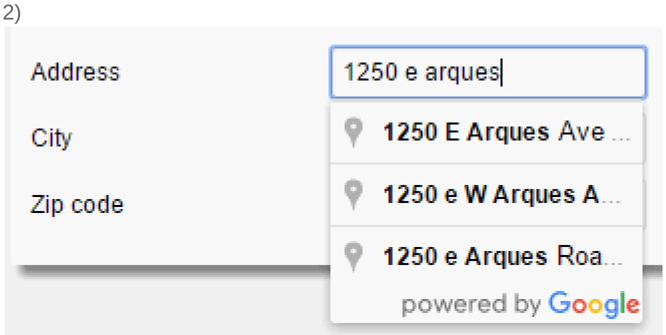

Introduction

Goal: configure a Google address autocomplete widget, so we enter partial address, that loads the matching items, and when we select a matching item, it load the fields of the address (city, zip code etc..)

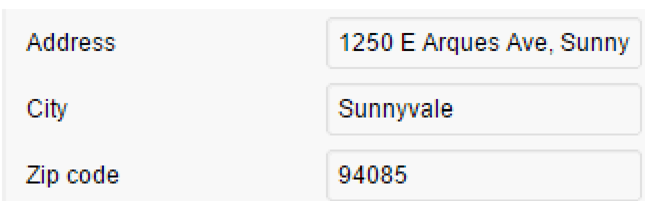
1)



2)



3)



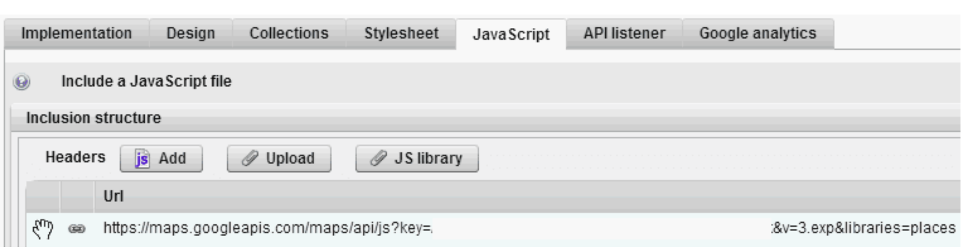
Build it

Get an API key

See https://developers.google.com/maps/signup?hl=en
=> an API key is not required, but strongly encouraged to use.
Go to https://developers.google.com/maps/documentation/javascript/get-api-key to get a key

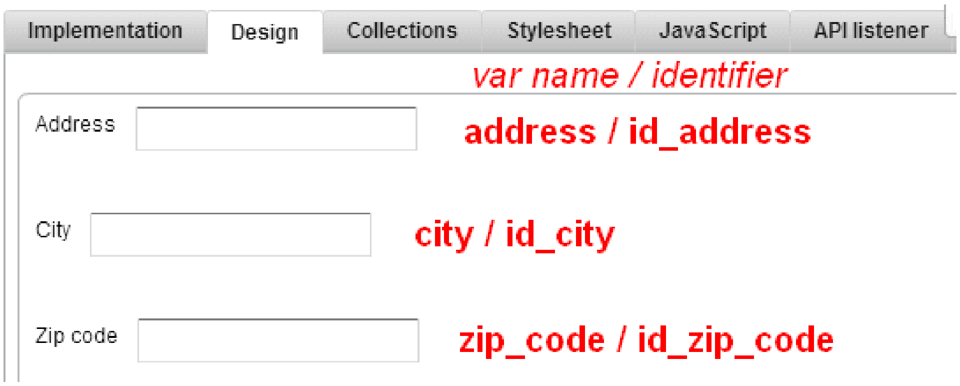
Configure the RMP Webinterface

Add in the header of the WI this .js:



Configure those 3 text input widgets:

Publish date	13 Feb 2018
Contributor	Hugo David E
Reviewer	reviewer meeting
Status	PUBLISHE



Then add a hidden JS widget at the bottom with the following code:

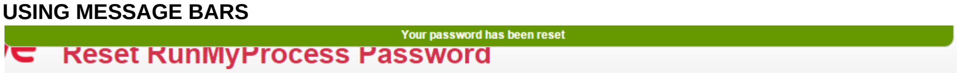
Save and give it a try. You're done!

Howto - Notification Banner

If you want to show a custom notification banner instead of default RunMyProcess banners which usually shows at the bottom right of the page after an action is performed, you can follow below approach and it shows banner at the top of the page like below screenshot.

Publish date	04 May 2018
Contributor	Hugo David E
Reviewer	Reviewer Meeting
Status	PUBLISHE

USING MESSAGE BARS



js call:

```
show_messagebar('red','An error occurred when
adding new product');
show_messagebar('green','Product added');
```

js function:

```
function show_messagebar(color, text) {
 //color = green OR red
 $('#_messagebar_' + color).text(text);
 $('#_messagebar_' + color).slideDown(800);
 setTimeout(function () {
 $('#_messagebar_' + color).fadeOut('slow');
 }, 5000)
}
```

html widget:

```
<div class="gwt-HTML-messagebar"
id="_messagebar_green" style="position: fixed;
top: 0px; display: none; visibility: visible;
background-color: rgb(102, 153, 0);"></div>
<div class="gwt-HTML-messagebar"
id="_messagebar_red" style="position: fixed;
top: 0px; display: none; visibility: visible;
background-color:red;"></div>
```

css:

```
.gwt-HTML-messagebar {
    background: #CC0000;
    color: white;
    visibility: hidden;
    border: solid 1px lightgray;
    padding: 3px 100px 5px 110px;
    -moz-border-radius-bottomleft: 10px;
    border-bottom-left-radius: 10px;
    -moz-border-radius-bottomright: 10px;
    border-bottom-right-radius: 10px;
    z-index: 200;
    font-family: Gotham, Arial;
    position: absolute;
    left: 100px;
    right: 100px;
    top: -10px;
    overflow: hidden;
    overflow-y: auto;
    text-align: center;
    font-weight: bold;
    max-width: 800px;
    margin-right: auto;
    margin-left: auto;
    font-color: #000000 important!;
}
```

Design Consideration

Performance considerations

Publish date	05 Jan 2018
Contributor	Mani
Reviewer	Richard Mang
Status	PUBLISHE

Basic rule

- Process with back-end (Process/CAPI) instead of using Javascript

Front-end

1. Load minimum files/components when open the web interface
   - Footer file and header file
   - CSS and pictures
   - Conditional load the widgets
   - Utilize tabs, sections to minimize the load components
2. Array widget
   More then 30 rows probably be slow (depends on how many widgets/js are inside of the array). Considering to use conditional load, to reduce the loading contents or using javascript

directly.

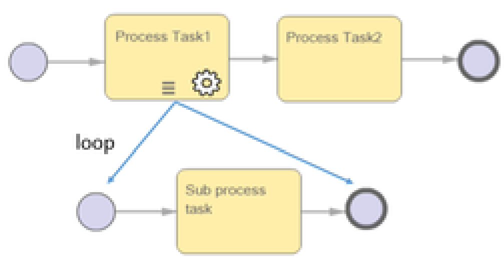
Back-end

1. Limitation of the platform
   Even though the platform is well designed as a cloud application, it is not unlimited. Read the limitation carefully before designing your application.

2. Synchronization and Asynchronization
   Composite API (CAPI): A CAPI is executed synchronously. The result of triggering a CAPI is the result of the execution.
   Process: Process is triggered asynchronously. The result of triggering a process is the result of sending the HTTP request. Once you receive a 200 OK status code, your process is executed asynchronized in backend.

   A CAPI has a limitation of 30 seconds execution time, so it should be used for "light" program only. If not, use process instead. In that case, you can use a connector to link the father and child processes. Refer Howto - Connect two processes / CAPI for detail steps.

   In the following case, suppose "Sub process task" is an activity taking time to finish, upper example would have better performance.
   Click here to expand...

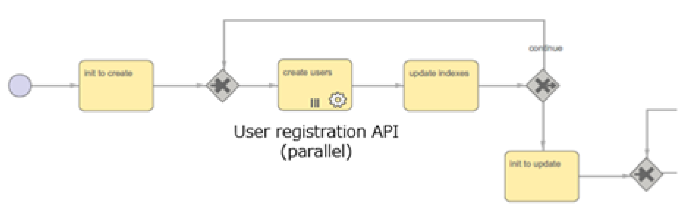


   Good example:
   father process connector (loop) child process

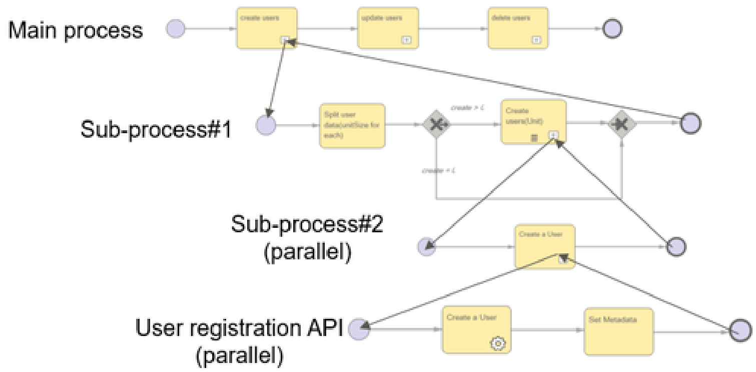
   Bad example:
   father process connector (loop) CAPI

3. Parallel execution
   Sub process and CAPI can be executed in parallel. Note that big number of parallel sub-process execution would reach platform limitations. When you see "Service Unavailable (503) ", it means the platform can not handle the parallel processes with the current definition.
   Click here to expand...
   Possible solutions:

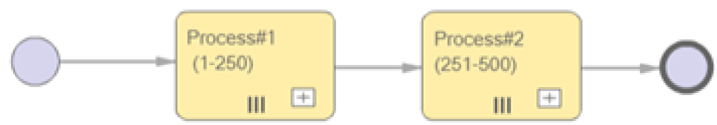
   a. Avoid using nested sub-process
      Good example: Parallel CAPI



      Bad example: MAIN Process Sub Process (Parallel) Single CAPI



b. Avoid using big number of parallel execution in one task. Divide the loop to different tasks.
Good example:



Bad example:

4. DB indexing
RMP uses Mongo DB as the main database. When the data size is big, it may impact on your application's performance. Using RMP API to create index will improve the performance.

5. Thread pool
Howto - Thread pool to leverage traffic

Restrict collection access scope

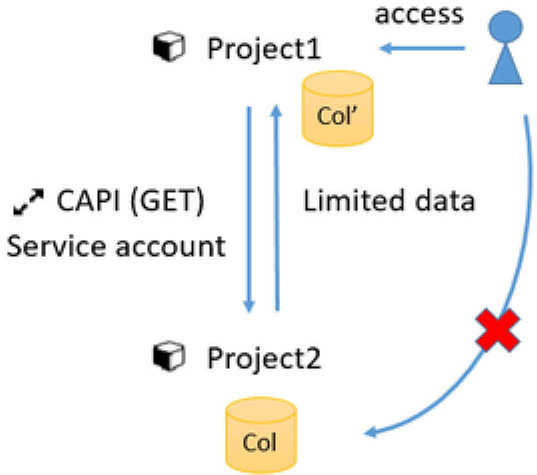
You should aware that a user can access the whole collection using the following URL, if he has the access right to the web interface, and the collection is attached to the web interface.

```
https://live.runmyprocess.com/live/{customer_id}/object/{collection_name}/?P_mode={mode}
```

For the collection that you don't want to show everything to the users, you may have the following solutions

- Use RMP API for encoding/cipher for the important information
- Divide the data to different projects

Publish date	05 Jan 2018
Contributor	Mani
Reviewer	Richard Mang
Status	PUBLISHE



Things to be considered during design

Internal account

Sometimes you need a dummy account for communications among CAPIs, batches, etc, not a real person. In that case, beware that a fake account is **NOT** allowed. A fake email would possibly generate bounce messages. Once your application generates an amount of bounce messages, the platform may stop all the messages sent from @runmyprocess.com for all customers.

Be careful that the following use case will generate automatic email notification unintentionally

- Collection backup

It is recommended to use a real email account all the time.

Publish date	05 Jan 2018
Contributor	Mani
Reviewer	Richard Mang
Status	PUBLISHE

Test Tool / Test methodology

Library / Code example

- CSS
- Freemarker
- Javascript
- Query and Aggregation
- Reference Project
- Regular Expression

Freemarker

- Change upload file's property
- Create a query object using Freemarker
- Date and time in Freemarker
- Encrypt / Decrypt a string
- Freemarker - has_content vs ??
- Get user, user metadata
- round, floor and ceilling
- Write JS in a process

Change upload file's property

Change property sample: visibility private public

Publish date	15 Mar 2018
Contributor	GSite
Reviewer	Sei FUJITA

```
<#assign file_list = []>
<#list attach_files as fu>
${save_file_desc(fu.id,{"visibility":"PUBLIC"})
}
</#list>
```

Note : **attach_files** is the variable name of file upload widget

Status	PUBLISHED

- Change property sample: visibility private public

Create a query object using Freemarker

Basic rule

Always use P_json_put to create query object.

Example1: key=value conditions

<Code>

```
<#assign pattern1 = {}>
<#assign pattern1 = P_json_put( pattern1,
"property_key1", property_value1, false)>
<#assign pattern1 = P_json_put( pattern1,
"property_key2", property_value2, false)>
${pattern1}
```

<Conditions>

- property_value1 = a and property_value2 = b

<Query output>

```
{"property_key2":"b","property_key1":"a"}
```

Example2: Add operators e.g. "$or"

<code>

Publish date	05 Jan 2018
Contributor	Mani
Reviewer	Mani
Status	PUBLISHED

```
<#assign pattern1 = {}>
<#assign pattern1 = P_json_put( pattern1,
"property_key1", property_value1, false)>
<#assign pattern1 = P_json_put( pattern1,
"property_key2", property_value2, false)>

<#assign pattern2 = {}>
<#assign pattern2 = P_json_put( pattern2,
"property_key3", property_value3, false)>
<#assign pattern2 = P_json_put( pattern2,
"property_key4", property_value4, false)>

<#assign patt_arr = []>
<#assign patt_arr = patt_arr + [pattern1]>
<#assign patt_arr = patt_arr + [pattern2]>

<#assign query_obj = {}>
<#assign query_obj = P_json_put( query_obj,
"$or", patt_arr, false)>
${query_obj}
```

<Conditions>

- property_value1 = a and property_value2 = b
  or
- property_value3 = c and property_value4 = d

<Query output>

```
{"$or":[{"property_key2":"b","property_key1":"a
"},{"property_key4":"d","property_key3":"c"}]}
```

<Re-formated view>

For more query examples, refer to Query and Aggregation

Date and time in Freemarker

This document partially quotes from codegeek (article1, article2) provided by
Rafal Urbanski

Publish date	21 Mar 2018
Contributor	Mani Rafal Urbans
Reviewer	Sei FUJITA David Courta
Status	PUBLISHE

Date and time in Freemarker

Sample code

- The freemarker code runs on the platform, where the time zone is UTC.
- Useful link to calcuate timestamp and human date string: https://www.epochconverter.com/
- You can use **Javascript** to calculate dates and times more precisely (for example if you want to get the date thrre months from today).
  Refer to Write JS in a process

- Sample code
  - Current server time (timestamp)
  - Current server time (string)
  - Change timestamp to date string
  - Setup timezone
  - Get day, month, year string from timestamp
  - Check if it is a future date or not

Current server time (timestamp)

Current server time (string)

Change timestamp to date string

Setup timezone

Examples: "GMT", "GMT+2", "GMT-1:30", "CET", "PST", "America/Los_Angeles".

Refer to https://docs.oracle.com/javase/7/docs/api/java/util/TimeZone.html

Get day, month, year string from timestamp

Check if it is a future date or not

Encrypt / Decrypt a string

Publish date	15 Mar 2018
Contributor	GSite
Reviewer	Sei FUJITA
Status	PUBLISHE

Sample code

Result

string_to_encrypt: hey what's up
cipher_config: {"ciphered":"A1LXKwUtyau5QtB5Qwr0lw==","iv":"RejwQ5J1Ch0oHg9fozC8iw=="}
encrypted_string: A1LXKwUtyau5QtB5Qwr0lw==
decrypted_string: hey what's up

Freemarker - has_content vs ??

You can use "??" and "has_content" to know whether a variable "exists" in FreeMarker but the behavior of them is difference.

In summary, "??" returns just whether a variable is defined but "has_content" returns whether a variable is defined and it has value.

Publish date	15 Mar 2018
Contributor	Sei FUJITA
Reviewer	ReviewerMeeting
Status	PUBLISHE

Pattern 1

If you don't define "x" in your code, the result is

??: false
has_content: false

Pattern 2

If you define "x" as "ABC", the result is

??: true
has_content: true

Pattern 3

If you define "x" as "", the result is

??: true
has_content: false

because x is defined but the value is ""(empty).
cf.

If you define "x" as 0(falsy value), the result is

??: true
has_content: true

Please note "has_content" returns whether a variable is defined and it has value not its value is falsy.

Pattern 4

If you define "x" as [], the result is

??: true
has_content: false

because x is defined but the value is an empty object.

Pattern 5

If you define "x" as {}, the result is

??: true
has_content: false

because x is defined but the value is an empty hash.

Note: handling a hash with "has_content"

If you don't define "x" but access "x.foo", FreeMarker engine returns error: "Expression x is undefined on line 1, column 16"

To avoid this error, update your code like this.

"x.foo" with round bracket is useful and good to read.

Get user, user metadata

Publish date	15 Mar 2018
Contributor	GSite Mani
Reviewer	Sei FUJITA
Status	PUBLISHE

Get user emails from a lane, output = String

```
<#assign my_string ="">
<#list get_lane_users(ORG_ID,ROL_-ID) as x>
<#assign my_string = my_string + x.login>
<#if x_has_next>
  <#assign my_string = my_string + ",">
</#if>
</#list>
${my_string}
```

Get user emails from a lane, output = array

```
<#assign data = get_lane_users(ORG_ID,ROLE_ID)>
<#assign all_emails =
inject_objects(data,"login")>
<#assign array_email =
transpose(all_emails).login>
${array_email}
```

Add all user in a role to runtime lane

```
<#assign data = get_lane_users(ORG_ID,ROLE_ID)>
<#list data as user_i>
  <#--assign users to the runtime lane -->
  <#assign users =
P_add_user_to_lane(ROLE_ID_RUNTIME,user_i.id)>
</#list>
```

Refer to P_remove_user_from_lane(laneId, userId)

Get user metadata

```
<#assign user_meta =
get_user_metadata(USER-LOGIN-ID)>
<#if user_meta.mobile?has_content>
user has mobile metadata
<#else>
user has no mobile metadata
</#if>
```

Update user metadata

```
${P_change_metadata(LOGIN_ID, "language",
"en")}
```

Refer to P_change_metadata

round, floor and ceilling

round, floor, ceiling function are supported in FreeMarker 2.3.13, which is not yet supported by RunMyProcess as of March, 2018. There are few options to do the calcuation:

Publish date	15 Mar 2018
Contributor	Mani
Reviewer	ReviewerMeeting
Status	PUBLISHE

Calculate all in Freemarker

round

```
```

Ouput:

- 2.4 2
- 2.49 2
- 2.5 3
- -2.5 3
- -2.49 -2
- -0.5 -1
- -0.4 0

floor

```
```

Output:

- 2 2
- 2.0 2
- 2.4 2
- 2.5 2
- 2.9 2
- 2.91 2
- 2.912 2
- -1 -1
- -1.4 -2
- -1.5 -2

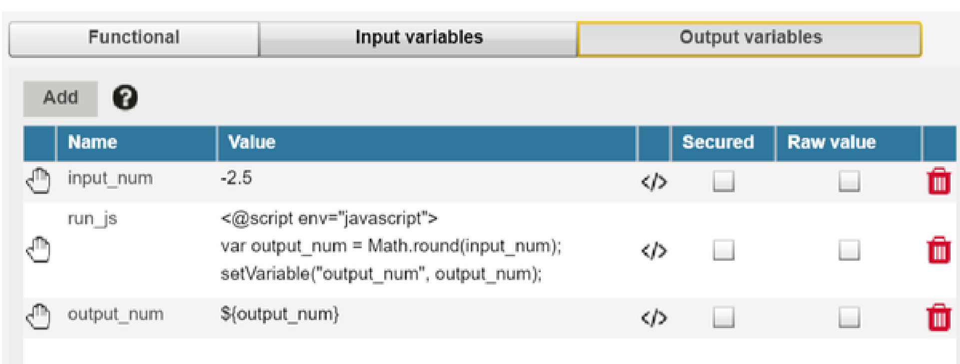
ceiling

Ouput:

- 2 2
- 2.0 2
- 2.4 3
- 2.5 3
- 2.9 3
- 2.91 3
- 2.912 3
- -0.5 0
- -1.5 -1
- -1.9 -1

Use Javascript code

Refer to Write JS in a process for the general procedure of inserting Javascript into Freemarker



round (run_js)

```
<@script env="javascript">
var output_num = Math.round(input_num);
setVariable("output_num", output_num);
</@script>
```

floor (run_js)

```
<@script env="javascript">
var output_num = Math.floor(input_num);
setVariable("output_num", output_num);
</@script>
```

ceiling (run_js)

```
<@script env="javascript">
var output_num = Math.ceil(input_num);
setVariable("output_num", output_num);
</@script>
```

Write JS in a process

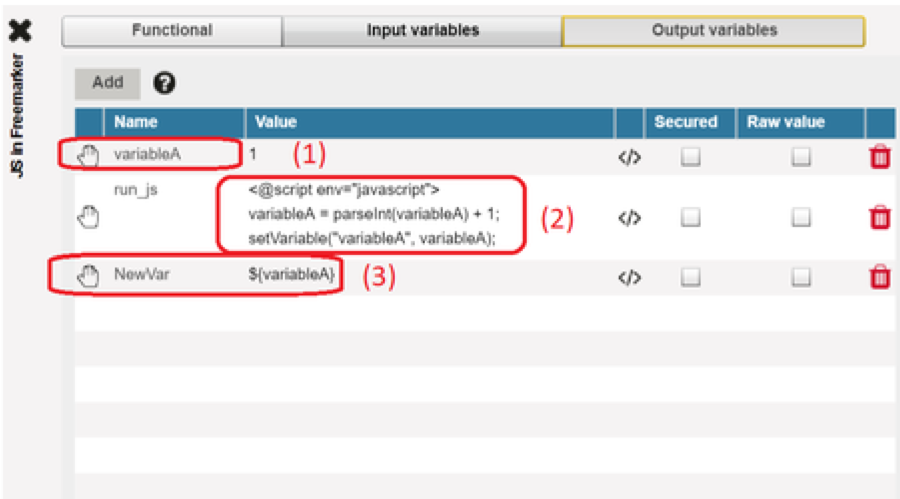
You can use the different methods to include javascript file or functions in a process. Refer to RMP doc for details.

Publish date	21 Mar 2018
Contributor	Mani
Reviewer	David Courta
Status	PUBLISHE

Include Javascript inside an activity

> This method is used when the Javascript part is small

Sample Process



- Javascript and Freemarker can not mix within one Freemarker variable code block
- You can use any process variable (defined with freemarker or coming from a web interface) inside your Javascript block
- To setup input value and use the return value to/from the Javascript code, you need to have 3 steps

  (1) Make sure the required input for the Javascript block is defined.

  (2) Execute Javascript code and use setVariable inside of the Javascript for returning the value.
  The first parameter of the setVariable function wil be the name of the variable available in the rest of the process.
  If you use an the name of an existing variable, its value will be overwritten. Otherwise a new process variable will be created.
  You can use the setVariable function several times in the same Javascript block to define several process variables.
  The name of the variable in which you insert the Javascript block ("run_js" in the example above) cannot be used to store data.

  (3) Use the updated variable in Freemarker after the JS code is executed

Sample code

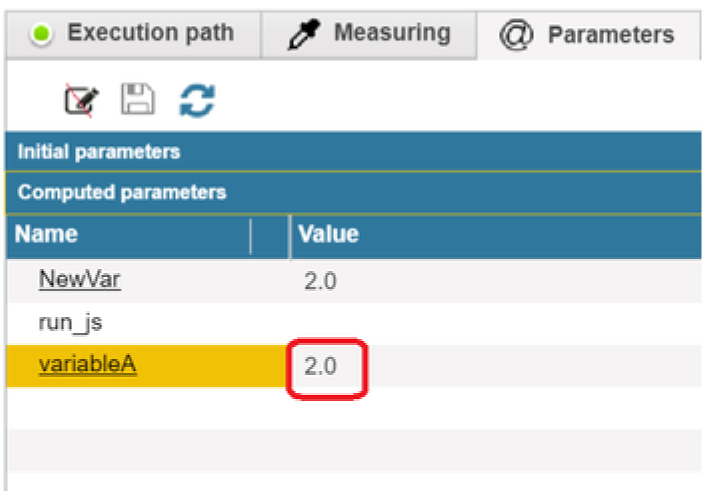
1. Setup the input variableA in the 1st line
2. Execute Javascript in the 2nd line

```
<@script env="javascript">
function myfunction () {
 variableA = parseInt(variableA) + 1;
 setVariable("variableA", variableA);
}
myfunction();
</@script>
```

3. Get the updated variable value in the 3rd line

Regarding variable type differences between Freemarker and Javascript

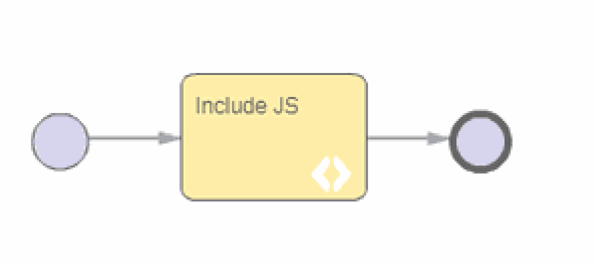
- Freemarker handles variables in "string" type. In Javascript side, you need to parse the data correctly.
- When you use "setVariable" to update a variable in Freemarker, the data type may be different between Freemarker and Javascript. In above example, variableA is defined as a integer in Javascript, but it is interpreted as a float number in Freemarker side:



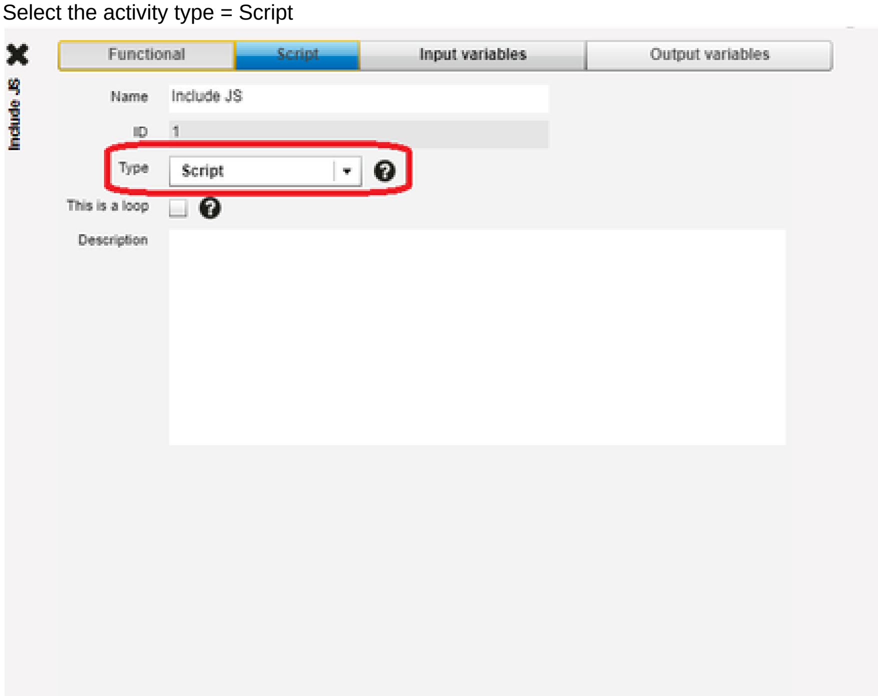
Include a JS script file

> This method is used when the Javascript part is big. Note that the included JS file is only valid WITHIN the current activity.

Sample Process

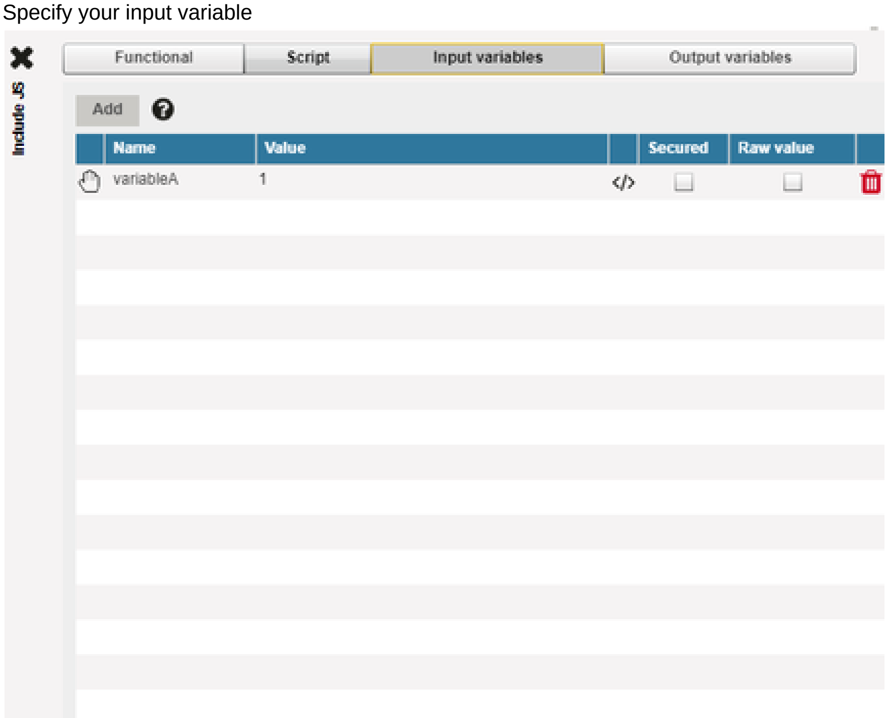


1. Select the activity type = Script

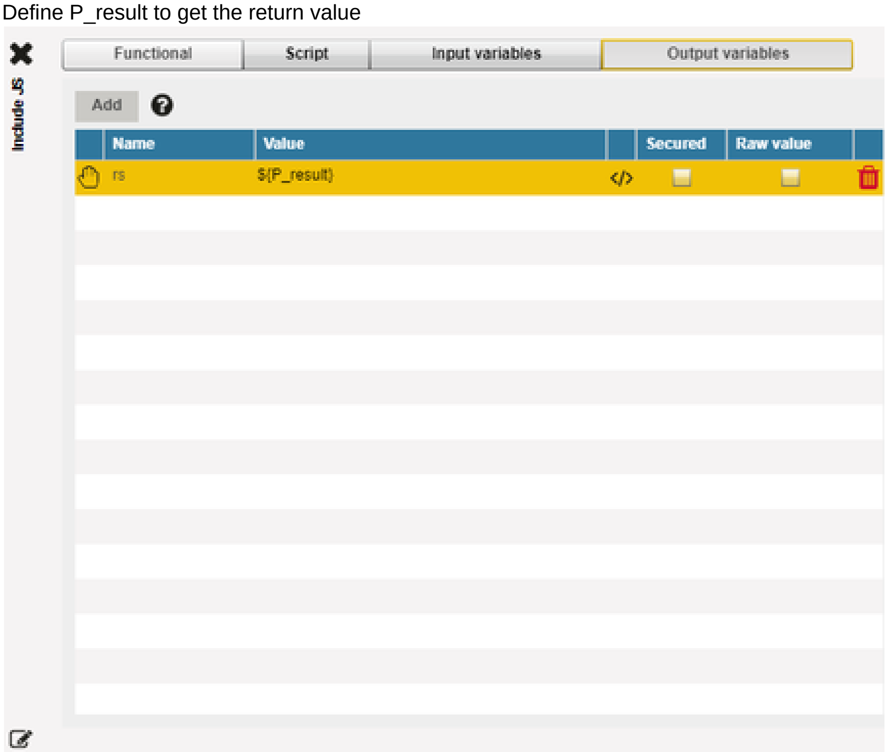


2. Select Type = Javascript and the uploaded file

3. Specify your input variable



4. Define P_result to get the return value

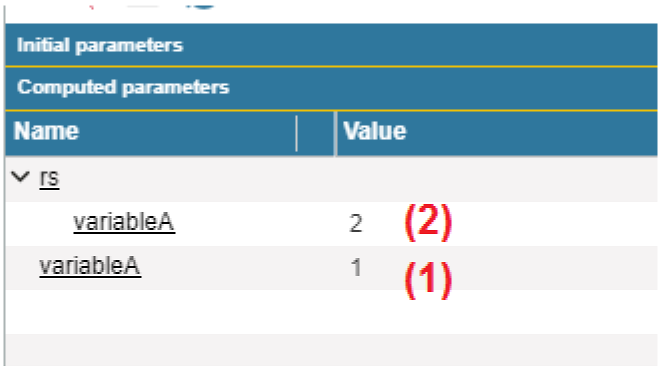


Sample code

```
myfunction();
function myfunction() {
    variableA = parseInt(variableA) + 1;
 setVariable("variableA", variableA);
}
```

Process result

Execute the process, you can get result as following



- (1) : the original variable value
- (2) : the updated value from the javascript

Query and Aggregation

Publish date	05 Jan 2018
Contributor	Mani
Reviewer	Mani
Status	PUBLISHE

Query sample on RMP manual

http://docs.runmyprocess.com/Developer_Guide/Collection/Collection_JS_Freemarker#advanced-queries

Mongodb manual

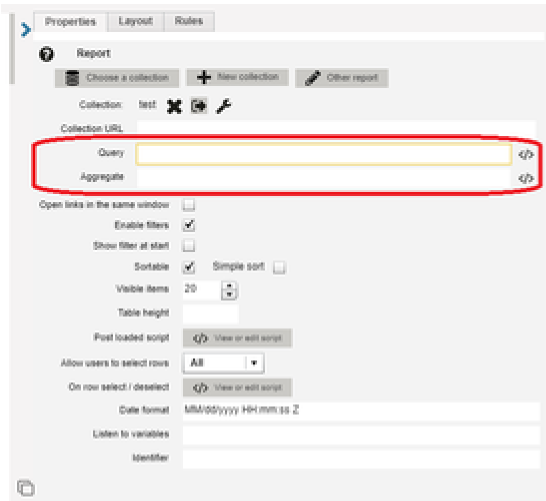
https://docs.mongodb.com/manual/tutorial/query-documents/

Query basis

- Create a query object using Freemarker
- Create a query object using Javascript

Query and Aggregation on the web interface

When you show a collection report, you can define the query/aggregation conditions directly on the widget. Refer to RMP doc here.



Examples

- Aggregation - $lookup and $unwind example
- Filter specific strings from collection
- Query - $and example
- Query - Case sensitive query
- Query - Get data from a date range

Aggregation - $lookup and $unwind example

RMP uses mongoDB, which is a non-relational database. When you have multiple collections, you can **combine** one or more columns by using values common to each other. The operator in mongoDB is $lookup and $unwind. These are similiar operation as JOIN in SQL.

Publish date	05 Jan 2018
Contributor	Sei FUJITA
Reviewer	Mani
Status	PUBLISHE

Reference:

- Mongodb manual

Example:

There are 2 collections.

Collection ONE

```
[
   {"name": "AAA", "FIELD_OF_ONE": "key1"},
   {"name": "BBB", "FIELD_OF_ONE": "key2"},
]
```

Collection TWO

```
[
   {"position": "MANAGER", "position_code": "2000", "FIELD_OF_TWO": "key1"},
   {"position": "ENGINEER", "position_code": "3000", "FIELD_OF_TWO": "key2"},
]
```

Join the two collections

You want to combine the two collections into one. Let's JOIN the 2 collections using **$lookup.**

Freemarker

```
<#local lookup = P_json_put_ext({},
'$lookup.from', 'COLLECTION_TWO', true)>
<#local lookup = P_json_put_ext(lookup,
'$lookup.localField', 'FIELD_OF_ONE', true)>
<#local lookup = P_json_put_ext(lookup,
'$lookup.foreignField', 'FIELD_OF_TWO', true)>
<#local lookup = P_json_put_ext(lookup,
'$lookup.as', 'NEW_FIELD', true)>

<#local result =
aggregate_collection('COLLECTION_ONE', lookup)>
```

result =
[
{"name": "AAA", "FIELD_OF_ONE": "key1", **"NEW_FIELD": {"position": "MANAGER", "position_code": "2000", "FIELD_OF_TWO": "key1"}**},
{"name": "BBB", "FIELD_OF_ONE": "key2", **"NEW_FIELD": {"position": "ENGINEER", "position_code": "3000", "FIELD_OF_TWO": "key2"}**},
...
]

Flat the result

In above example, collection two (foreignField) is joined to collection one (localField) as "NEW_FIELD". The type of NEW_FIELD is an array. Now we are going to use $unwind to make it flat

Freemarker

```
<#local lookup = P_json_put_ext({},
'$lookup.from', 'COLLECTION_TWO', true)>
<#local lookup = P_json_put_ext(lookup,
'$lookup.localField', 'FIELD_OF_ONE', true)>
<#local lookup = P_json_put_ext(lookup,
'$lookup.foreignField', 'FIELD_OF_TWO', true)>
<#local lookup = P_json_put_ext(lookup,
'$lookup.as', 'NEW_FIELD', true)>

<#local unwind = P_json_put_ext({}, '$unwind',
'$NEW_FIELD', true)>

<#local result =
aggregate_collection('COLLECTION_ONE', lookup,
unwind)>
```

result =
[
{"name": "AAA", "FIELD_OF_ONE": "key1", **"position": "MANAGER", "position_code": "2000", "FIELD_OF_TWO": "key1"**},
{"name": "BBB", "FIELD_OF_ONE": "key2", **"position": "ENGINEER", "position_code": "3000", "FIELD_OF_TWO": "key2"**},
]

Now you got a perfect JOINed objects.

Performance Notice

When your collection is big, $lookup and $unwind would generate huge calculation and impact on the performance. A rough calculation can be considered as following:

Nb of objects in collection#1 * Nb of objects in collection#2.

E.g.

Collection A has 1000 objects and Collection B has 2000, then the measure is 1000 * 2000 = 2,000,000.
If the result exceeds **4.5M**, the execution time could be over 30sec, which is CAPI execution limit.

Note that the actual execution time depends on more facts such as the size of object, comp lexity of query, whether the collection has index or not etc. Above formular is a rough estimation. The important thing is that you should aware that $lookup and $unwind may impact on the performance. When you test, you should start from a small number, and increase the number gradually.

Filter specific strings from collection

Useful links

- Examples on RunMyProcess Doc: here
- MongoDB doc (for a detailed description of the available options with the $regex operator): here

Not containing specific strings (Useful in Collection aggregation pipeline)

```
{
    "$match":{
        "$and":[
            {"YOUR_OBJECT_KEY":{"$regex" : "^(.(?!YOUR_STRING1))*$" , "$options" : "i" }},
            {"YOUR_OBJECT_KEY":{"$regex" : "^(.(?!YOUR_STRING2))*$" , "$options" : "i" }}
        ]
    }
}
```

Does not contain a specific string

```
var my_pattern = {};
my_pattern.destination_label = {"$regex" :
"^(.(?!gambetta))*$" , "$options" : "i" };
id_collection.listCallback(my_pattern,{},callba
ckSuccess,callbackFailure);
```

This will retrieve all the elements in which `destination_label` field **does not** contain "gambetta" (case insensitive)

Case insensitive

```
pattern = {"field":{"$regex" : "YOUR_REGEXP" ,
"$options" : "i" }};
```

Partially contain a string

```
var my_pattern = {};
my_pattern.destination_label = {"$regex" :
".*gambetta.*" , "$options" : "i" };
id_collection.listCallback(my_pattern,{},callba
ckSuccess,callbackFailure);
```

This will retrieve all the elements in which `destination_label` field contains "gambetta" (case insensitive)

Publish date	15 Mar 2018
Contributor	GSite
Reviewer	ReviewerMeeting
Status	PUBLISHED

Query - $and example

This document partially quotes from codegeek provided by Rafal Urbanski

Publish date	24 Jan 2018
Contributor	Rafal Urbans
Reviewer	Mani
Status	PUBLISHE

Data Structure


```
[{
 "employee_name":"Tester One",
 "employee_email":"tester_one@codegeeks.xyz",
 "data_started": 1437734858,
 "date_left" : 32472144000,
 "employee_division":"Paris",
 "employment_type":"permanent"
},{
 "employee_name":"Tester Two",
 "employee_email":"tester_two@codegeeks.xyz",
 "data_started": 1420119218,
 "date_left" : 1548341030,
 "employee_division":"Berlin",
 "employment_type":"contractor"
},{
 "employee_name":"Tester Three",
 "employee_email":"tester_three@codegeeks.xyz",
 "data_started": 1437734858,
 "date_left" : 32472144000,
 "employee_division":"Berlin",
 "employment_type":"permanent"
},{
 "employee_name":"Tester Four",
 "employee_email":"tester_four@codegeeks.xyz",
 "data_started": 1437734858,
 "date_left" : 1485269030,
 "employee_division":"Paris",
 "employment_type":"contractor"
},{
 "employee_name":"Tester Five",
 "employee_email":"tester_five@codegeeks.xyz",
 "data_started": 1437734858,
 "date_left" : 32472144000,
 "employee_division":"London",
 "employment_type":"permanent"
},{
 "employee_name":"Tester Six",
 "employee_email":"tester_six@codegeeks.xyz",
 "data_started": 1437734858,
 "date_left" : 1548341030,
 "employee_division":"London",
 "employment_type":"contractor"
}]
```

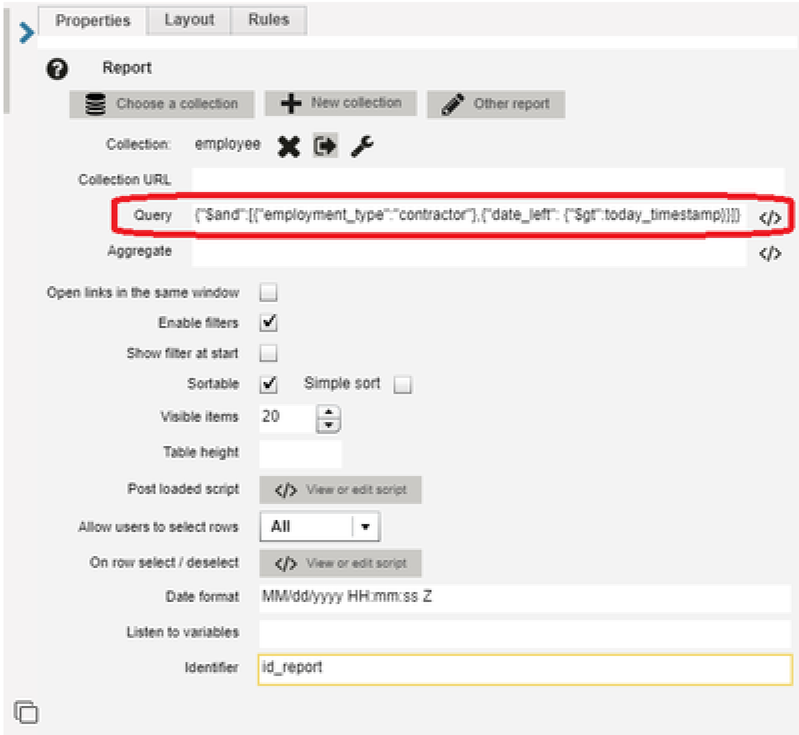
Query Condition

- employee type = contractor
  AND
- date_left = later than a specified date (e.g. today)

Sample Code Javascript

```
var current_date = new Date();
var current_time =
Math.round(current_date.getTime()/ 1000);

//find the employee who is still working for
the project
var my_pipeline =
{
    "$and": [
            { "employment_type":"contractor"} ,
            { "date_left": { $gt : current_time
} }
    ]
};


id_report.setQuery(JSON.stringify(my_pipeline))
;
id_report.refresh();
```

You also can directly setup the query on the report widget



Result

The following two matches the query conditions are shown

```
[{
 "employee_name":"Tester Two",
 "employee_email":"tester_two@codegeeks.xyz",
 "data_started": 1420119218,
 "date_left" : 1548341030,
 "employee_division":"Berlin",
 "employment_type":"contractor"
},{
 "employee_name":"Tester Six",
 "employee_email":"tester_six@codegeeks.xyz",
 "data_started": 1437734858,
 "date_left" : 1548341030,
 "employee_division":"London",
 "employment_type":"contractor"
}]
```

Query - Case sensitive query

Publish date	05 Jan 2018
Contributor	Rafal Urbans
Reviewer	Mani
Status	PUBLISHE

This document partially quotes from codegeek provided by Rafal Urbanski

Query condition

Case sensitive search

Sample code freemarker

```
<#assign my_employee =
".*"+current_employee+".*">
<#assign pattern = {}>
<#assign pattern =
P_json_put(pattern,"$regex",my_employee,true)>
<#assign pattern =
P_json_put(pattern,"$options","i",true)>
<#assign result =
list_objects(pattern,"employees_collection")>
<#assign result = inject_objects(result)>
${result}
```

Query - Get data from a date range

Publish date	05 Jan 2018
Contributor	Rafal Urbans
Reviewer	Mani
Status	PUBLISHE

This document partially quotes from codegeek provided by Rafal Urbanski

Data structure

```
[{
 "campaign_id":"abc",
 "manager_email_address":"john.smith@mycompany.
com",
 "created_date":1443700800
},{
 "campaign_id":"def",
 "manager_email_address":"jimmy.gibson@mycompan
y.com",
 "created_date":1443800800
},{
 "campaign_id":"ghi",
 "manager_email_address":"john.smith@mycompany.
com",
 "created_date":1443900800
},{
 "campaign_id":"jkl",
 "manager_email_address":"roger.moore@mycompany
.com",
 "created_date":1444000800
},{
 "campaign_id":"mno",
 "manager_email_address":"john.smith@mycompany.
com",
 "created_date":1444100800
}]
```


Query condition

You want to find all records from the following date range:
1443900000 (Sat, 03 Oct 2015 19:20:00 GMT) to **1444099999** (Tue, 06 Oct 2015 02:53:19 GMT)

Sample code freemarker

```
<#assign pattern1 = {}>
<#assign pattern2 = {}>
<#assign pattern3 = {}>
<#assign pattern4 = {}>

<#assign fromDay = 1443900000>
<#assign toDay = 1444099999>

<#assign pattern1 = P_json_put( pattern1,
"$gte", fromDay, false)>
<#assign pattern2 = P_json_put( pattern2,
"$lte", toDay, false)>
<#assign pattern3 = P_json_put( pattern3,
"created_date", pattern1)>
<#assign pattern4 = P_json_put( pattern4,
"created_date", pattern2)>

<#assign patt_arr = []>
<#assign patt_arr = patt_arr + [pattern3]>
<#assign patt_arr = patt_arr + [pattern4]>

<#assign myObject = {}>
<#assign myObject = P_json_put( myObject,
"$and", patt_arr)>

<#assign result =
list_objects(myObject,"campaign")>

<#assign result = inject_objects(result)>
${result}
```

Result

```
{
 "campaign_id":"ghi",
 "manager_email_address":"john.smith@mycompany.
com",
 "created_date":1443900800
},{
 "campaign_id":"jkl",
 "manager_email_address":"roger.moore@mycompany
.com",
 "created_date":1444000800
}
```

Regular Expression

Useful Links

- https://regex101.com/

- Regular expression samples

Regular expression samples

Useful link

https://regex101.com/

> Replace "\" with "\\" when using freemarker

2 digit decimal pattern

```
^[0-9]*\.[0-9][0-9]$
```

4 digital number

```
^[0-9]{4}$
```

France-style telephone number

```
^((?:\+33\s|0)[1-9](?:\s\d{2}){4})$
```

Matches

- +33 1 23 45 67 89
- 01 23 45 67 89

Alphabet & number

```
^[0-9a-zA-Z]*$
```

No symbolic character allowed.

Publish date	15 Mar 2018
Contributor	Mani, GSite
Reviewer	Sei FUJITA
Status	PUBLISHE

1 byte character

```
^[0-9a-zA-Z -\~]*$
```

English alphabet, number and symbolic character only.

No bracket

```
^(?!.*({|}|\[|\])).*$
```

Blocking structured inputs to protect collection.