



Workflow Designer

Execute a Workflow

JavaScript

Beside Frontend Triggers, which have an internal handling, you can also trigger Workflows from Javascript in VtigerCRM.

Simple by using this JavaScript code:

```
// CRMID you want to use as main record to execute the workflow
var crmId = 881;

// Workflow ID you want to execute
var workflowId = 1;

var execution = new WorkflowExecution();
execution.setCallback(function(response) {
    // result handler, when workflow is completed
    // When you use "javascript response" block, you will get data in
    response.respondedata
});

execution.init(crmId );
execution.setWorkflowById(workflowId );

execution.execute();
```

Methods

There are several methods you can use to customize the execution. They are optionally and must be executed before the execute() call.

```
execution.enableRedirection(false);
```

This function enable/disable redirections you use within your workflow. When set to false, the user won't be redirected to any page and page do not reload when workflow is finished.

When workflow send downloads they are still processed.

```
execution.setBackgroundMode(true);
```

Per default the workflow is blocking UI during execution AND reload current page when finished. The background mode disable both results.

When workflow send downloads or redirections they are still processed.

```
execution.setEnvironment({"var1":"value1"});
```

Set the environmental variables of the execution to the parameter you set.

PHP

You could execute a workflow manually in your own vtigerCRM extension or any PHP file.

You only need the following PHP Code:

```
$envValues = array(
    "envkey" => "envvalue1",
    "envkey2" => "envvalue2"
);

// CRMID you want to use as main record to execute the workflow
$recordcrmid = 881;

// Workflow ID you want to execute
$workflowid = 1;

$objWorkflow = \Vtiger_Module_Model::getInstance('Workflow2');
$objWorkflow->runWorkflow($workflowid, $recordcrmid, $envValues);
```

