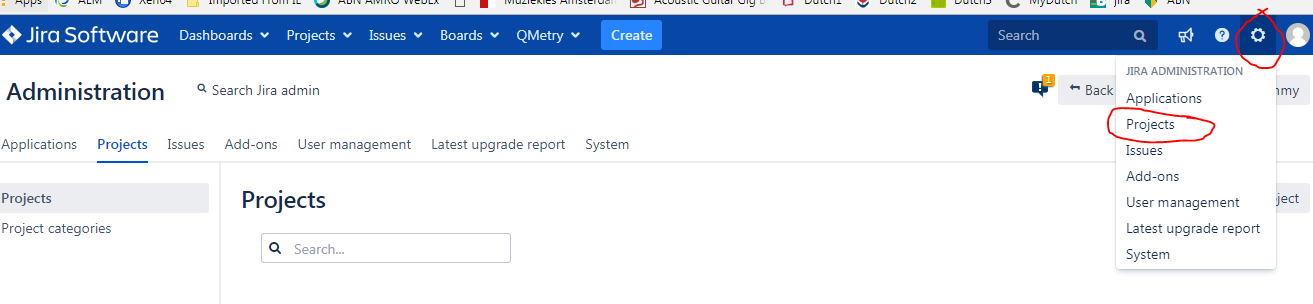
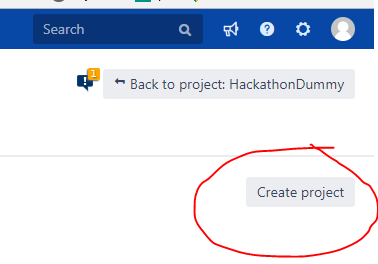
**Jira**

1. **Creating a project(As an admin user)**

ADMINISTRATION > JIRA ADMINISTRATION > PROJECTS

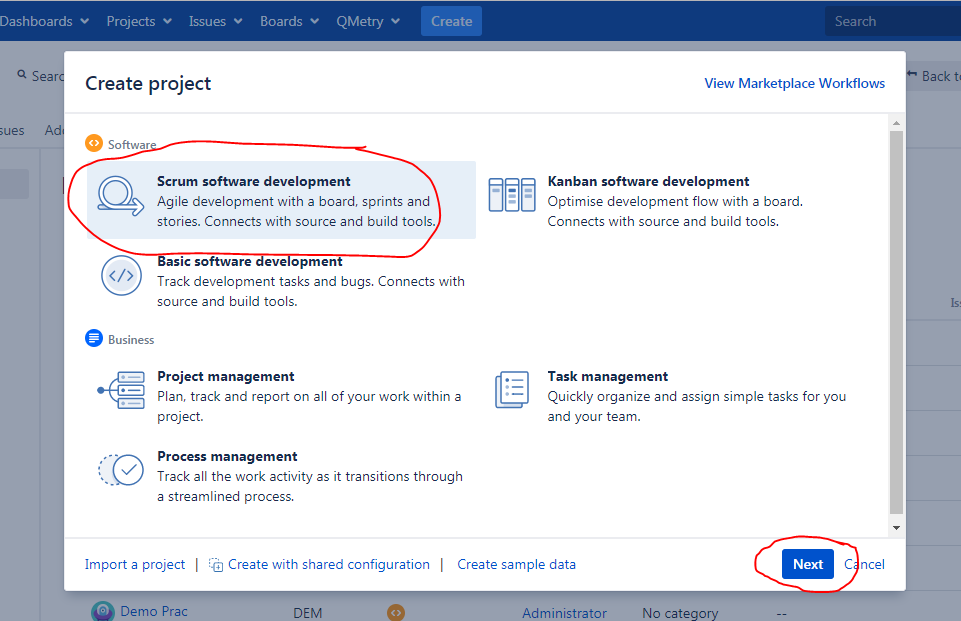


Create a project

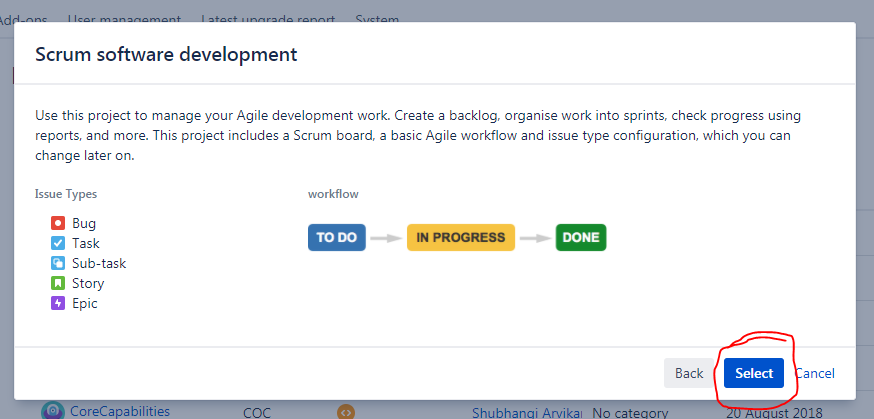


Create a relevant project

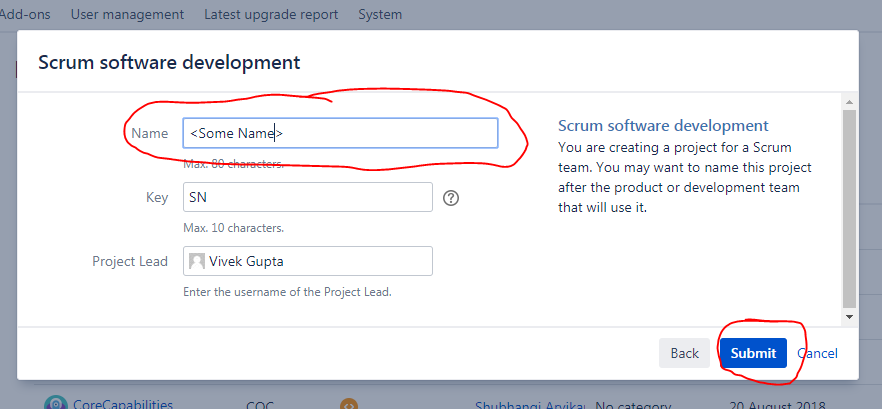
a) In this case, Scrum Software Development > Next



b) Select



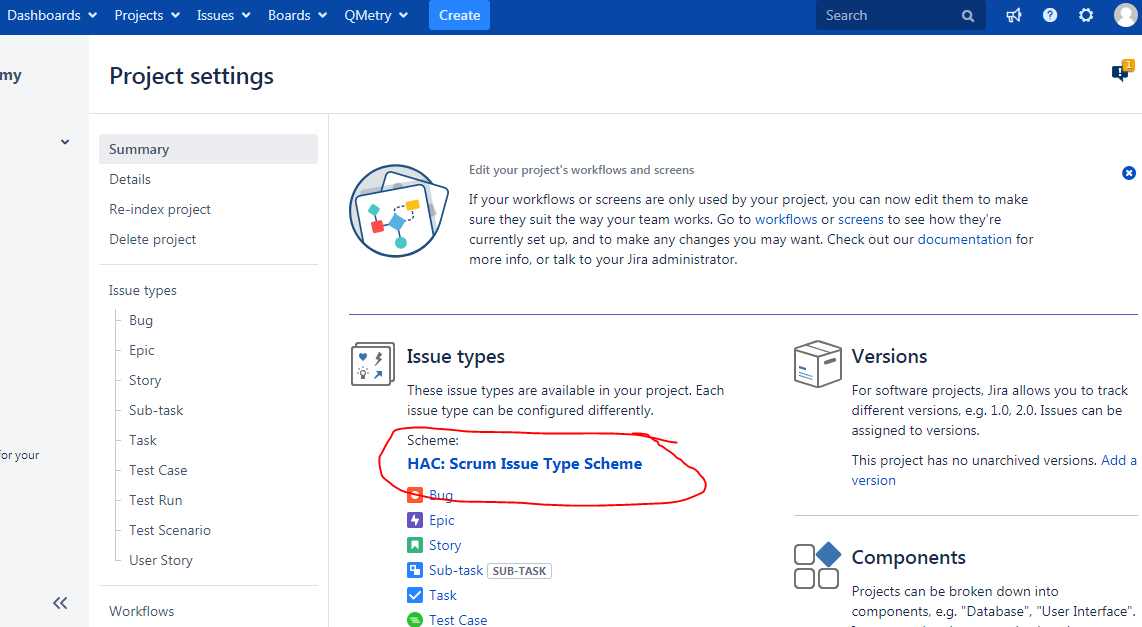
c) Give a relevant project name and submit



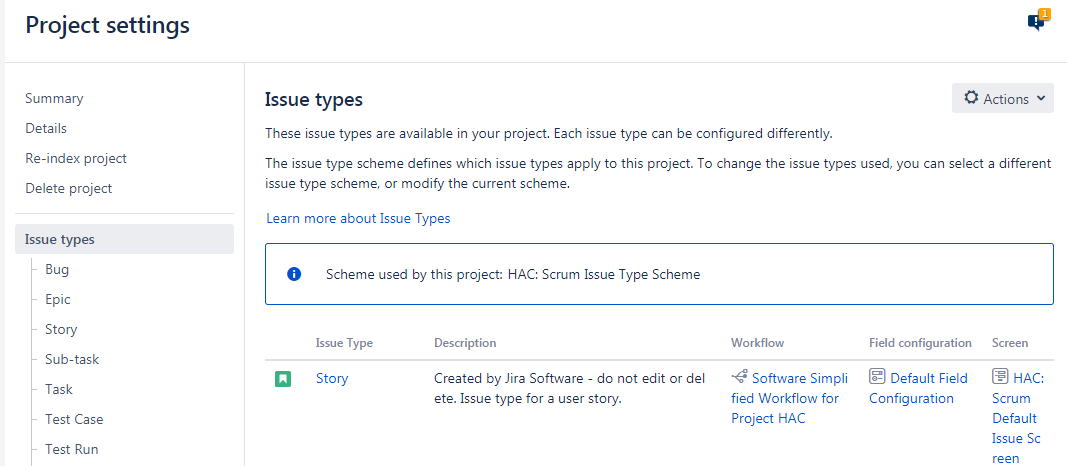
1. **Mapping different issue types within your project**

ADMINISTRATION > JIRA ADMINISTRATION > PROJECTS

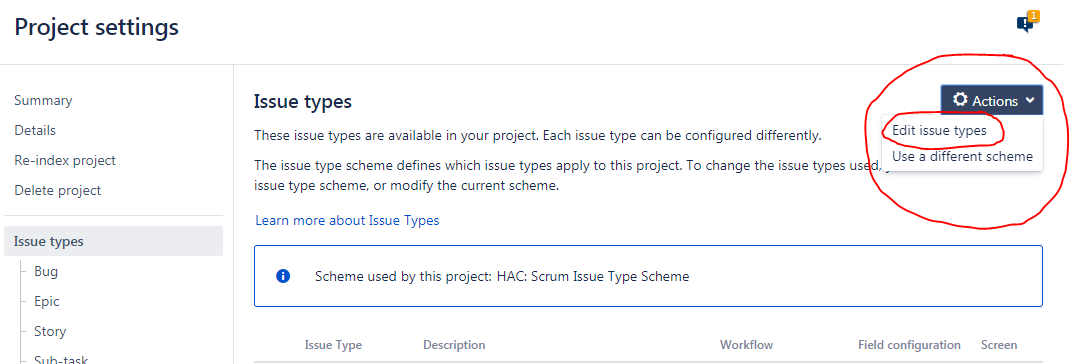
a) Select your project from the list(You are navigated to following page)



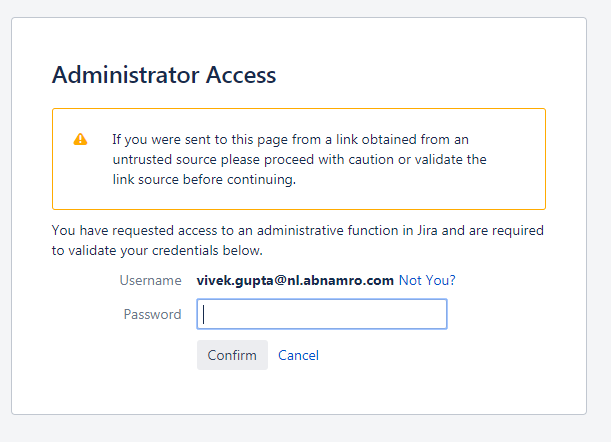
b) Select HAC: Scrum Issue Type Scheme(And you’ll be navigated to following page)



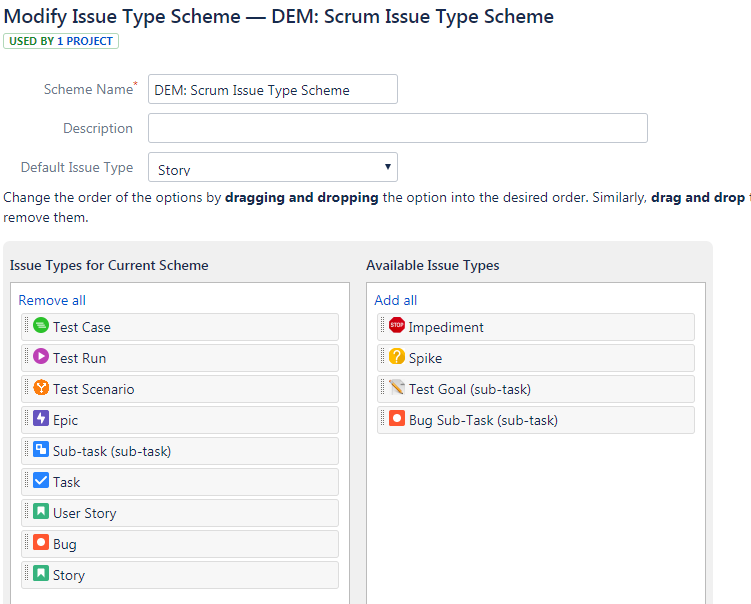
c) Go to Actions > Edit issue types



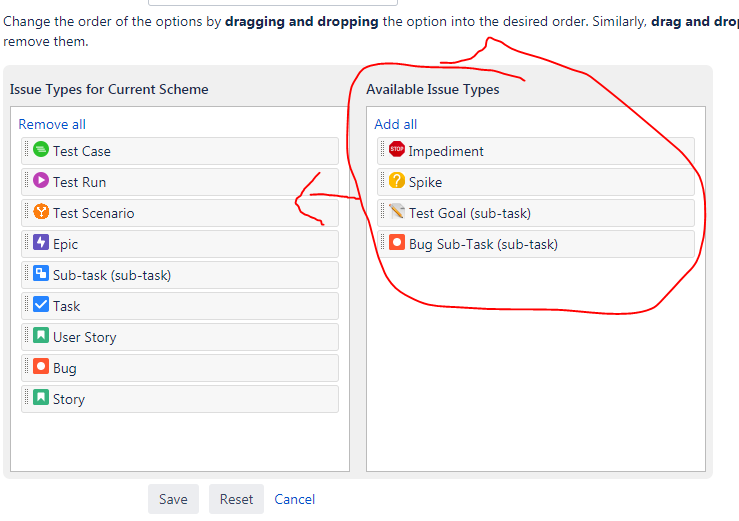
d) Confirm your authorisation



e) You’ll be navigated to following screen



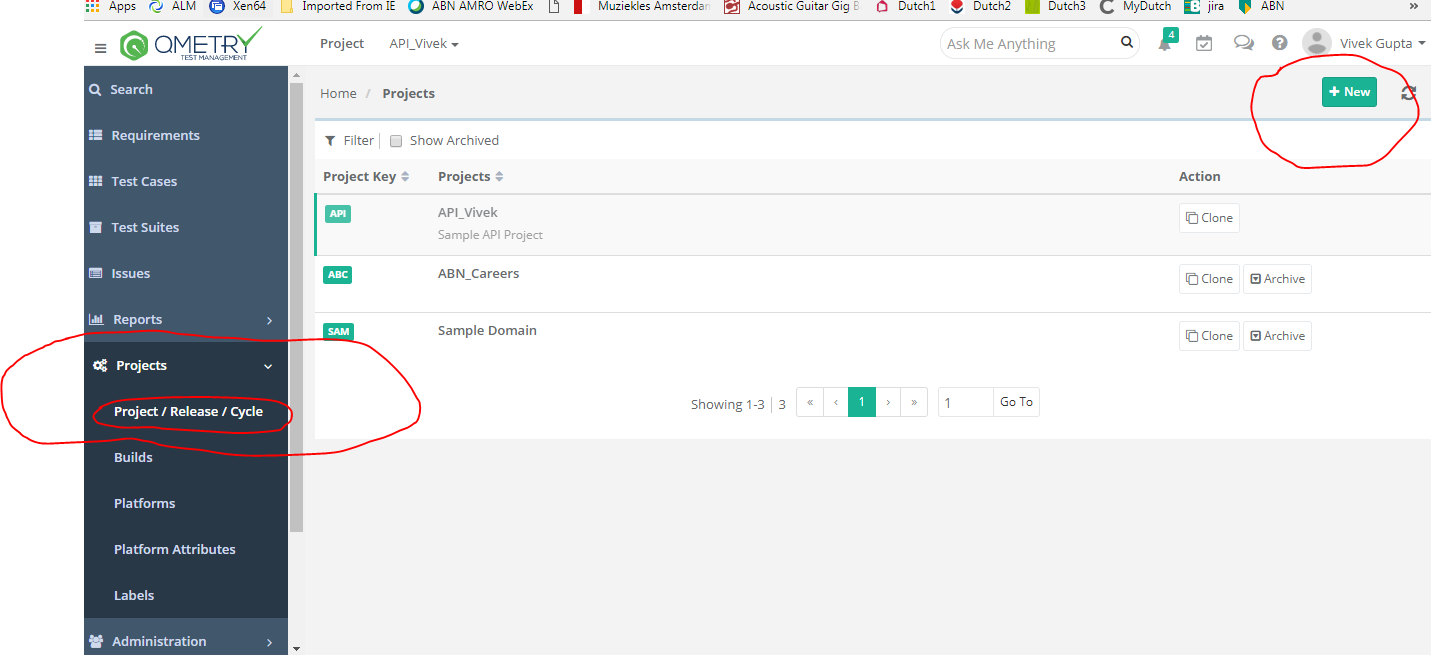
f) Add relevant issue types to your current scheme > Save



**Qmetry**

1. **Creation of a project within Qmetry and mapping/clonning it the hackathondummy project within Jira**

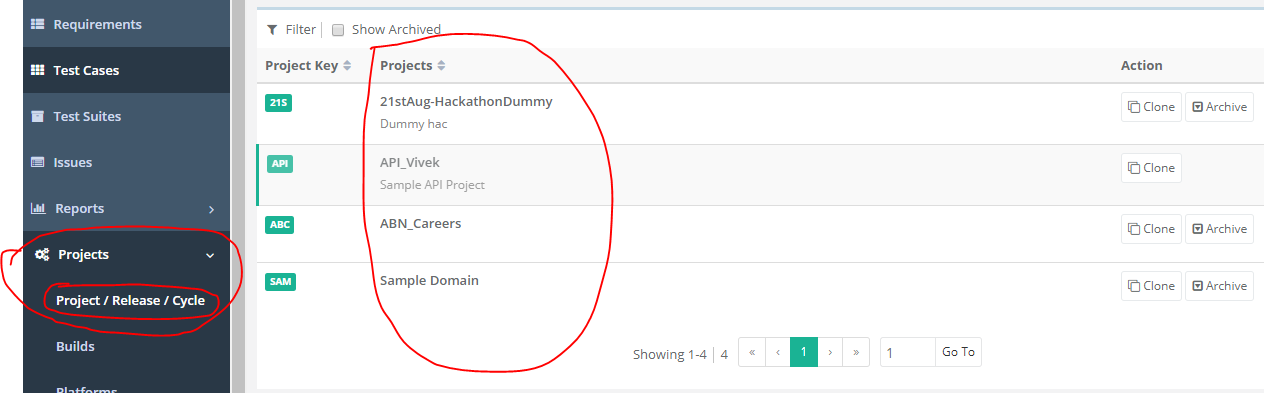
a) Project > Project/Release/Cycle > Click New



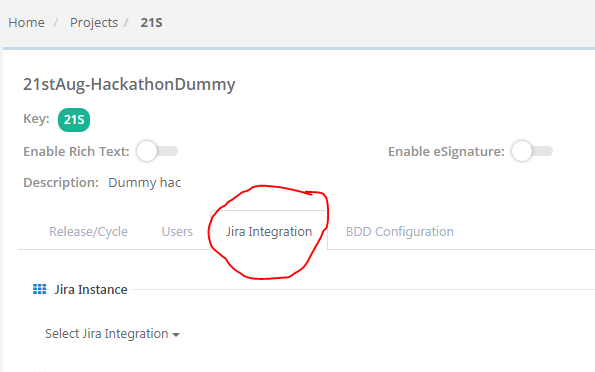
b) Filled in the relevant parameters > Create



c) Select your project from the list

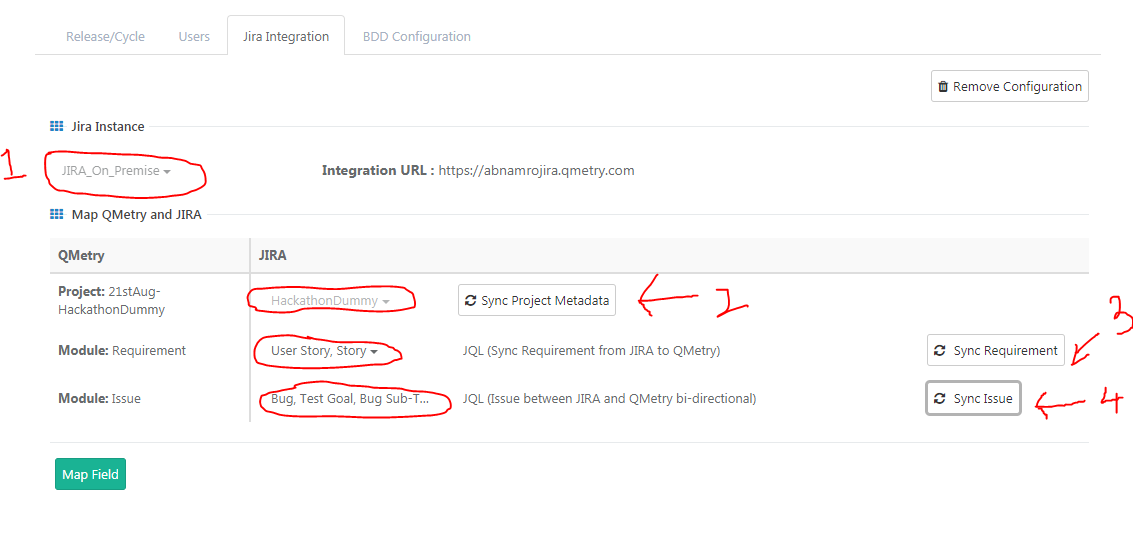


d) After selecting the project, navigate to Jira Integration tab

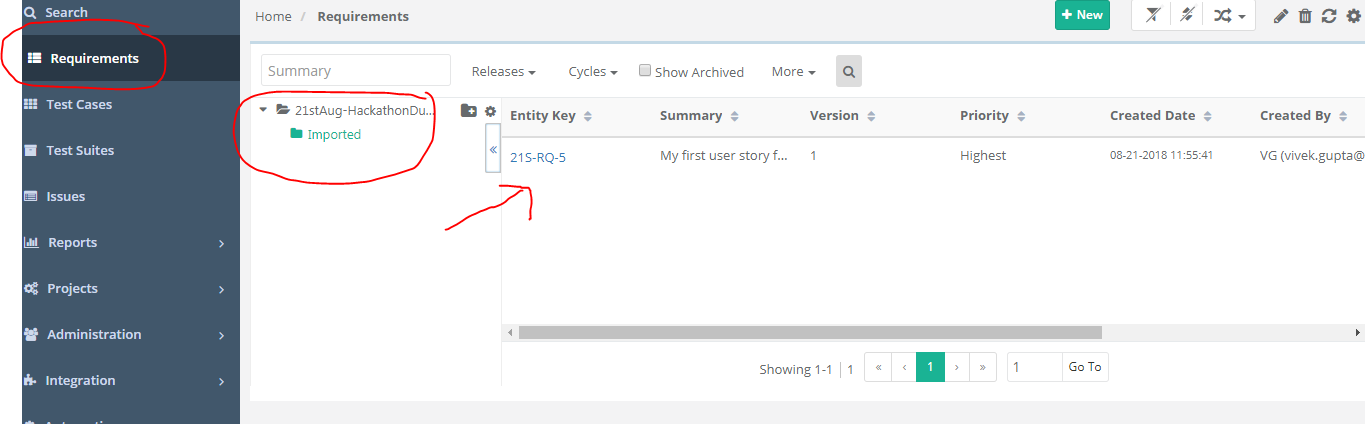


e) Following steps in order to map a jira project in to qmetry:-

1. Select Jira Instance > JIRA\_On\_Premise
2. Select Jira project that you want to integrate > Click on Sync Project Metadata
3. Select Jira requirement modules from the drop down > Click on Sync Requirement
4. Select Jira issue module from the drop down > Click on Sync Issue



f) Navigate to Requirements module > upon successful mapping you should be able to see the Jira requirements within Qmetry



**Note : Mapping is uni-directional (From Jira to Qmetry)**

**{Click on Entity Key and show the Jira requirement within Qmetry}**

**{Make changes in Jira and show it being reflected within Qmetry}**

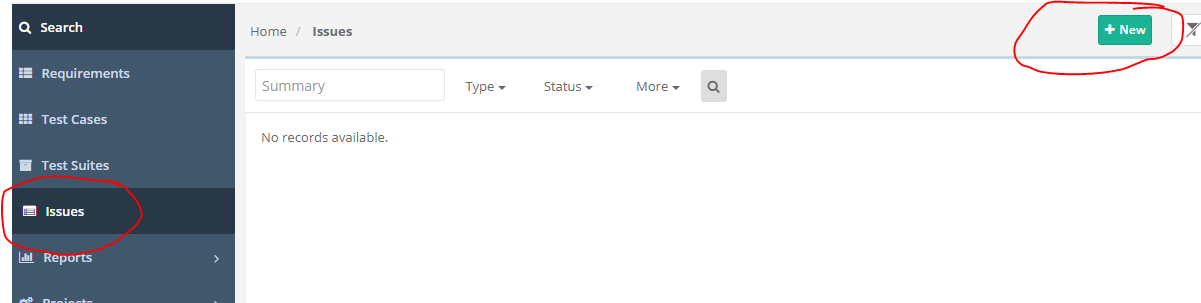
**{Addition of new columns, example of External Key which is link to jira user story}**

**{Generic features; bulk operations, filters to sort requirements, import}**

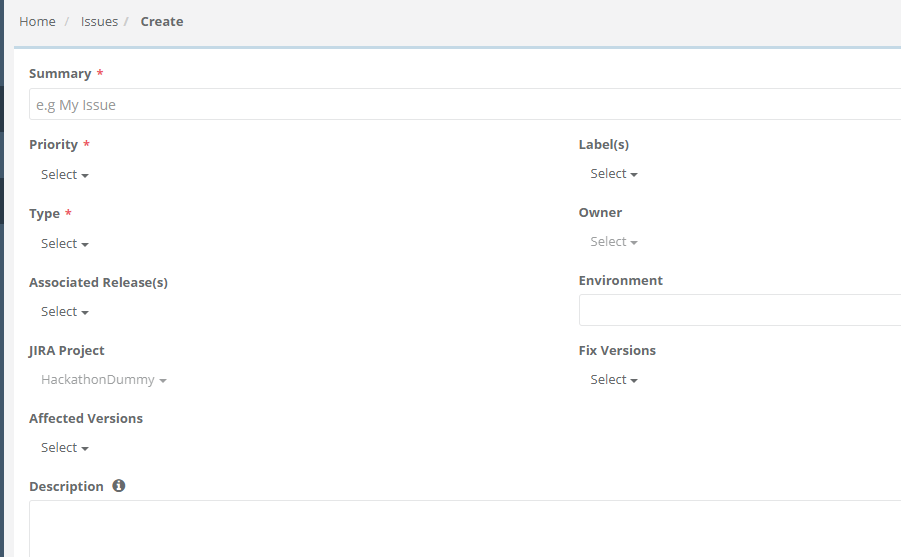
**Issue**

1. Creating an issue of type Bug from Qmetry

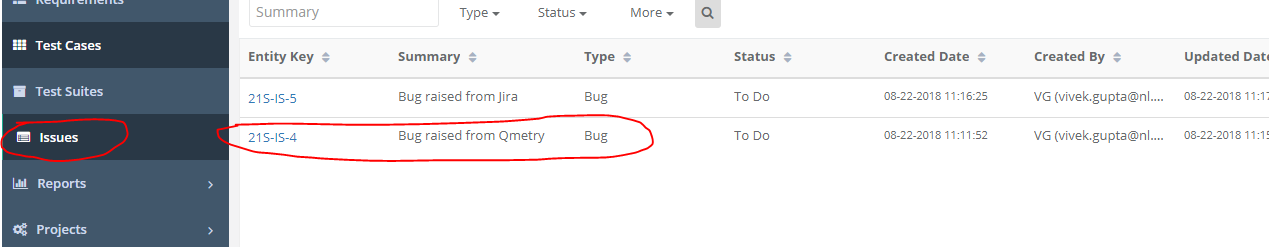
a) Issues > Click on New



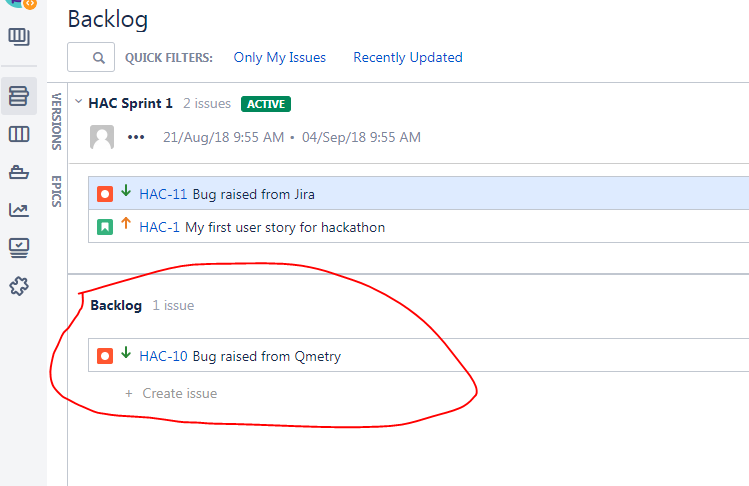
b) Fill in the issue details > Click on Create



c) An issue of type Bug is created

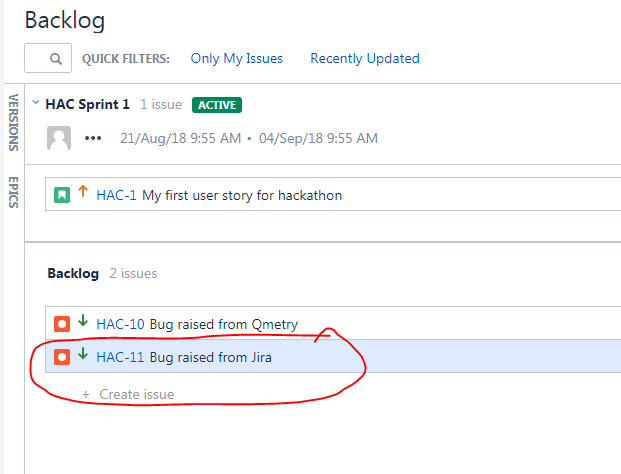


d) It’s also added in to the Jira backlog of your project

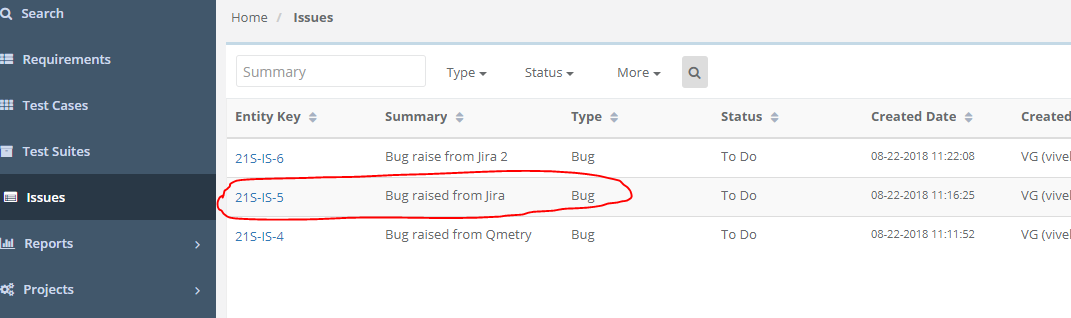


1. Creating an issue of type Bug from Jira

a) Create an issue of type bug in Jira



b) Issue can be seen in Qmetry



**{Click on Entity Key and show the Jira requirement within Qmetry}**

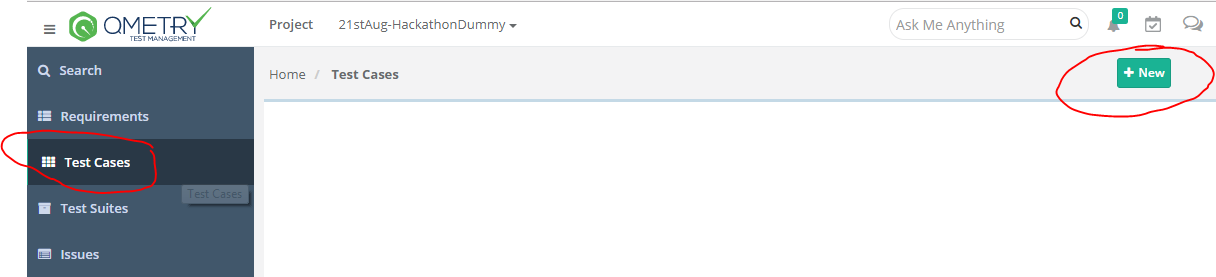
**{Make changes in Jira and show it being reflected within Qmetry}**

**{Generic features; bulk operations, filters to sort requirements, import}**

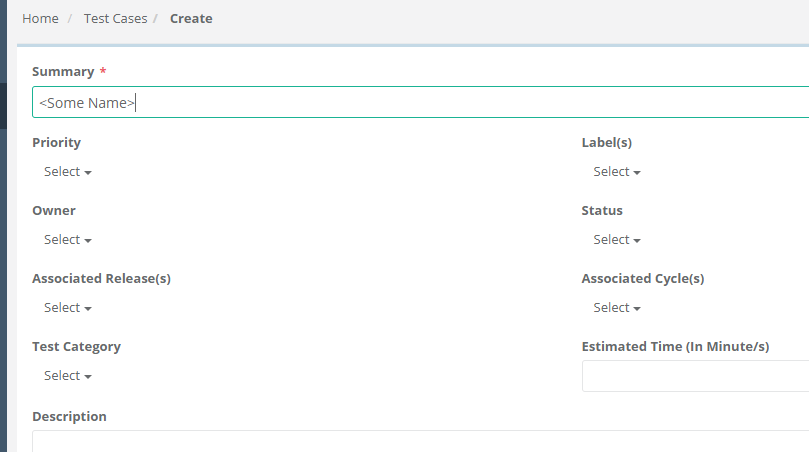
**Test Cases & Test Suite**

1. Creating a manual test case

a) Test Cases > New

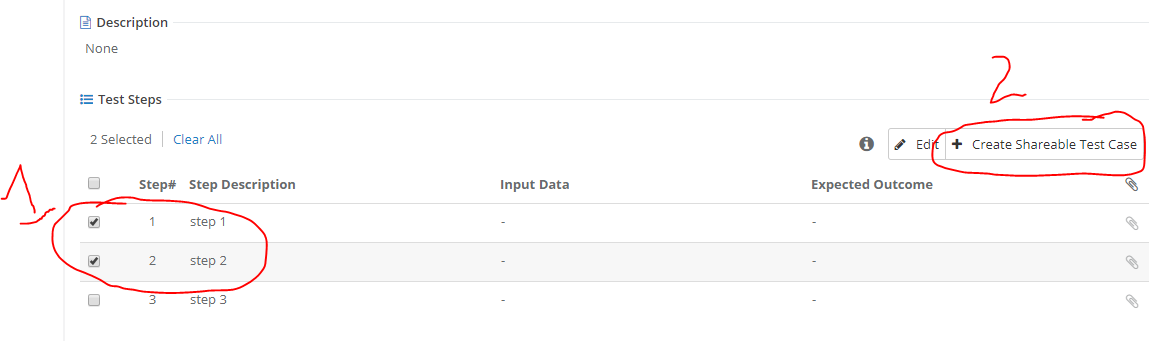


b) Fill in relevant details > Create

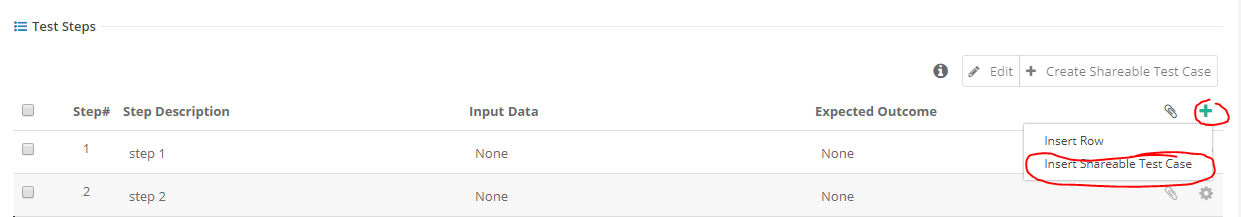


\*Creation of sharable test case using test steps

Select test steps > Click on “Create Shareable Test Case”

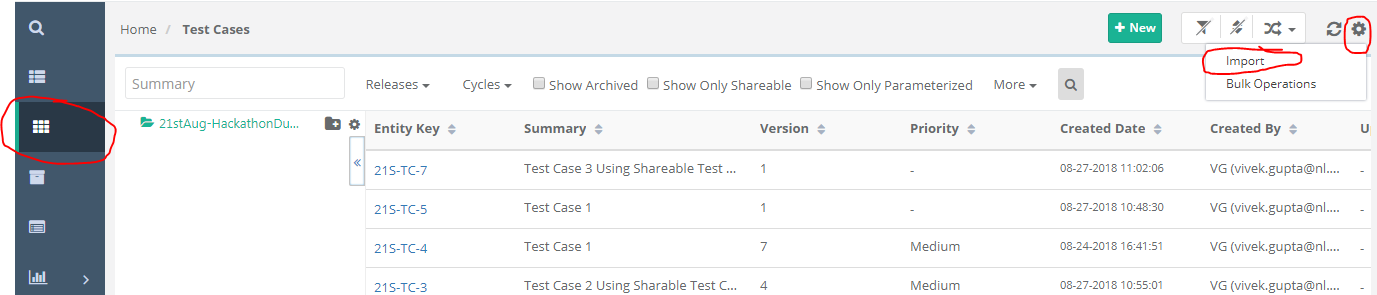


In order to use shareable test case, you can make use of the option “Insert Shareable Test Case” while adding test steps during test case creation

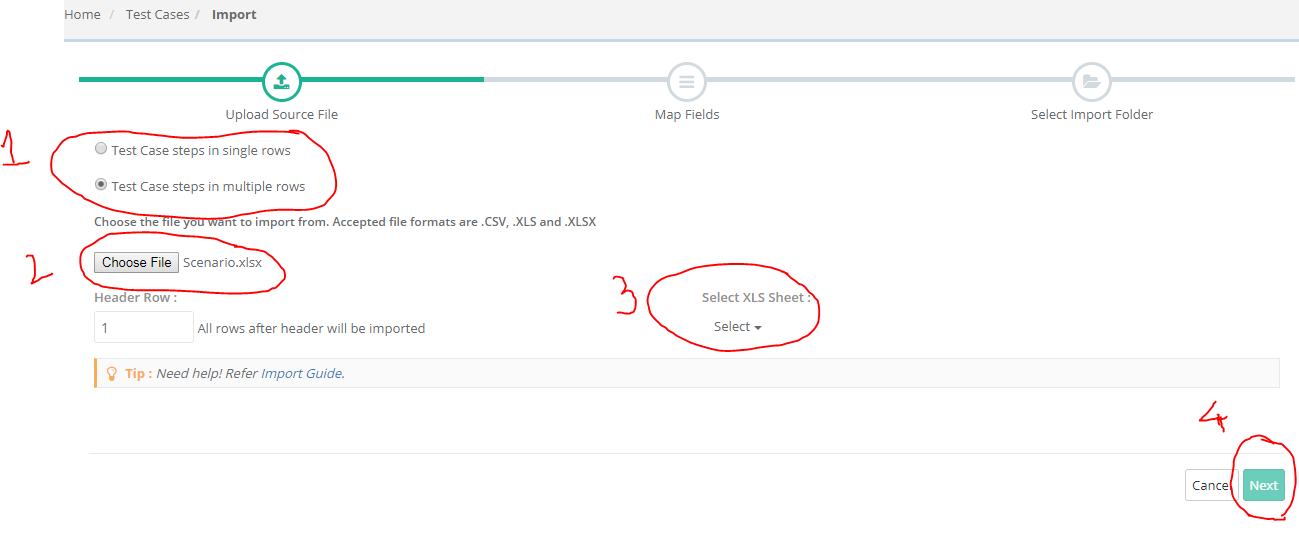


1. Import test case from external source(excel sheet in this case)
2. Click on C:\Users\C40060.LAUNCHER\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\N3SOW0V6\settings-1630709_960_720[1].png symbol > Click on “Import”

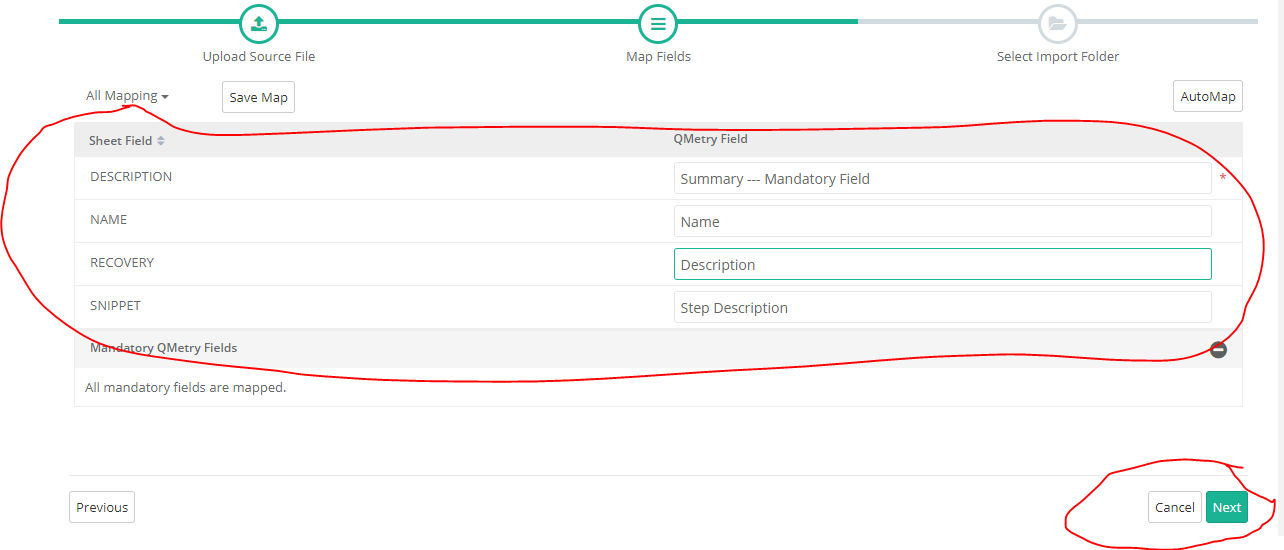




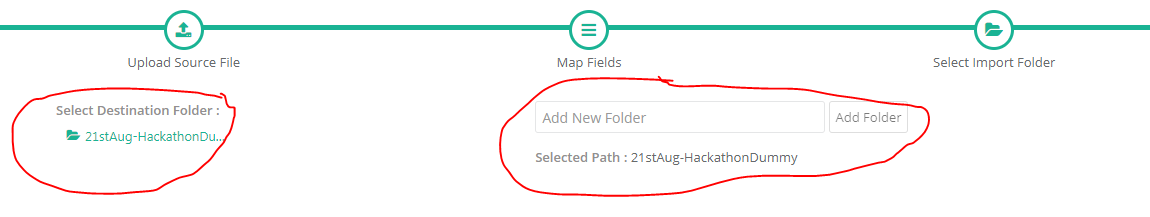
1. Fill in relevant details



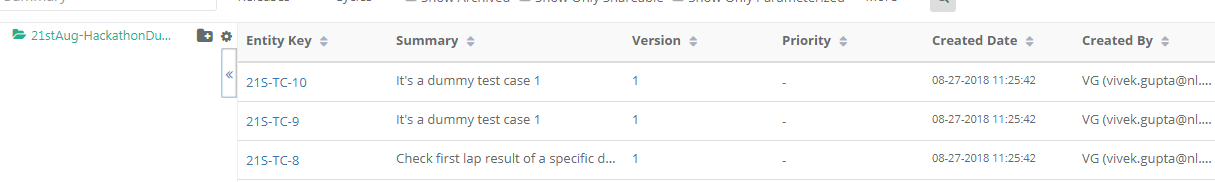
1. Mapping of excel fields with Qmetry fields



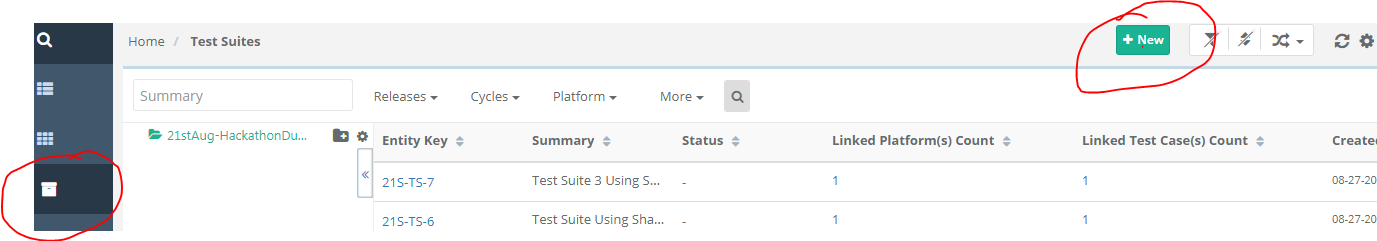
1. Select destination folder(or add a new folder)



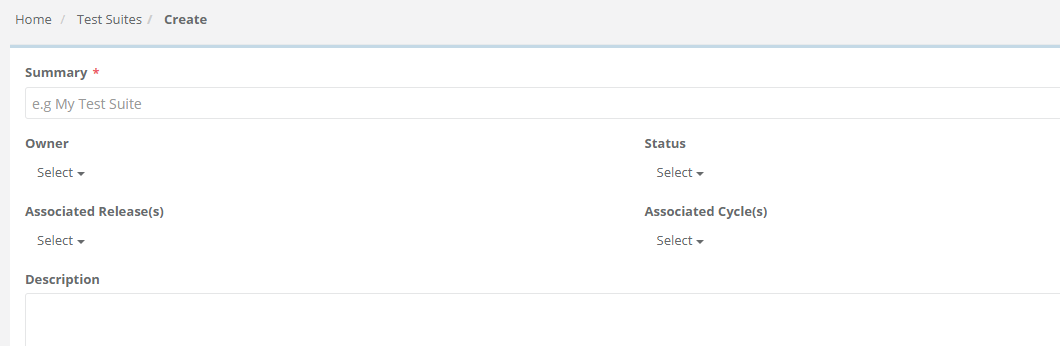
1. Test cases are imported successfully



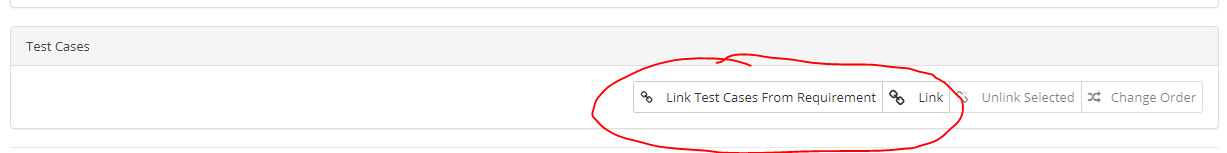
1. Creating a test suite and adding test cases
2. Test Suites > Click on “New”



1. Fill in relevant details



1. Add test case via “Link Test Cases From Requirement” or “Link”



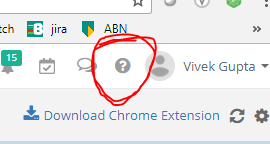
**{Create shareable test case}**

**{Associate release, cycle and platform before execution}**

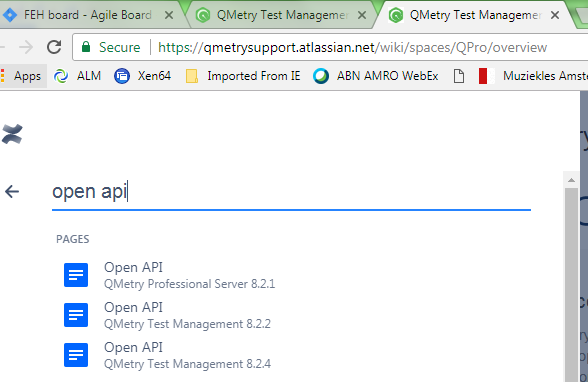
**{Generic features; bulk operations, filters to sort, import}**

**Open API**

**\***Go to help section/guide and search for “Open API”



\*Search “Open API” and look for latest version



{Help Guide : How to test QMetry Open REST API}

{Download Swagger.json}

{Import this swagger.json in to Postman to see the list of API calls}

1. Results import
2. Pre-requisite to have Junit or TestNG xml template supported by Qmetry

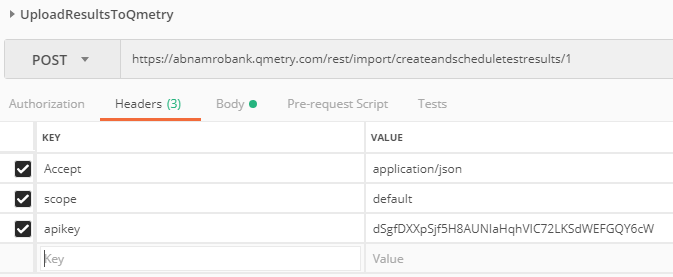
 

**Make sure within these xml file, you have a parameter called <name> which has an inbuilt link to the name of your test case within Qmetry**

1. URL: <https://abnamrobank.qmetry.com/rest/import/createandscheduletestresults/1>

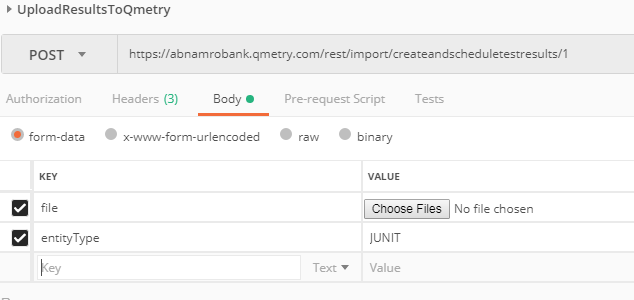
1. Headers:

|  |  |
| --- | --- |
| Accept | application/json |
| Scope | default |
| apikey | (From Qmetry) Administration > API Keys > Generate Automation API Key |



1. Body Parameters

|  |  |
| --- | --- |
| file | <Choose A File> |
| entityType | JUNIT |



1. Click on Send and you should see following message in case of a successful call(200 OK)

{

"success": true,

"code": "CO.IMPORT\_SCHEDULED",

"data": [

{

"id": 208,

"buildID": 62,

"platformID": 66,

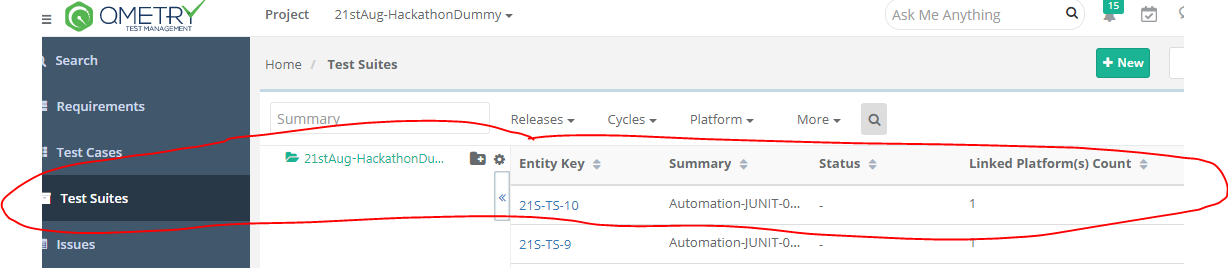
**"testsuiteId": "21S-TS-10"**

}

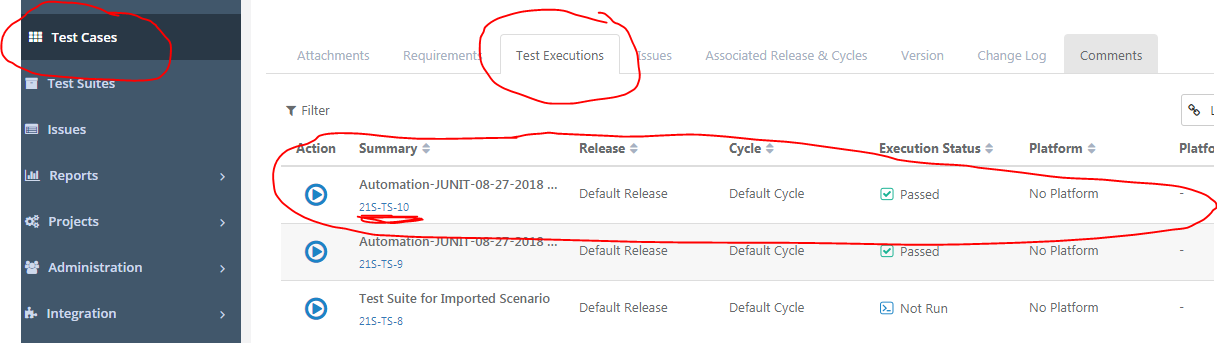
]

}

1. A new test suite will be created with latest execution results under “Test Suites” module of Qmetry(21S-TS-10 in this case)



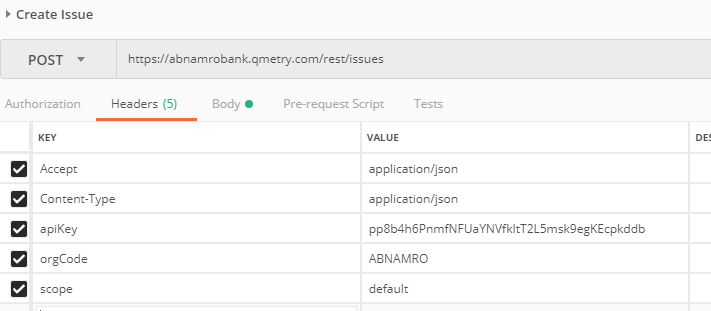
1. Execution results can be found in the” Test Case” module for that particular test case under “Test Execution” section



1. Creating a bug via Open API(Service Name: Create Issue)
2. URL: <https://abnamrobank.qmetry.com/rest/issues>

1. Headers:

|  |  |
| --- | --- |
| Accept | application/json |
| Content-Type | application/json |
| apikey | (From Qmetry) Administration > API Keys > Generate Open API Key |
| orgCode | ABNAMRO |
| scope | default |



1. Body:

{

"sync\_with":"5",

"issueType": "3910",

"issuePriority":"3923",

"issueState":"3935",

"externalProject":"111",

"summary":"Bug Creation Through API 27th August",

"description":"This Bug is created using the open Qmetry API"

}

1. Click on Send and you should see following message in case of a successful call(200 OK)

{

"success": true,

"code": "IS.ADD\_ENTITY\_SUCCESS",

"data": [

{

**"id": "21S-IS-10",**

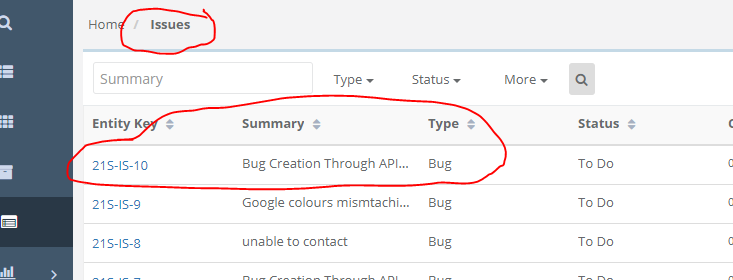
"dfId": 171

}

]

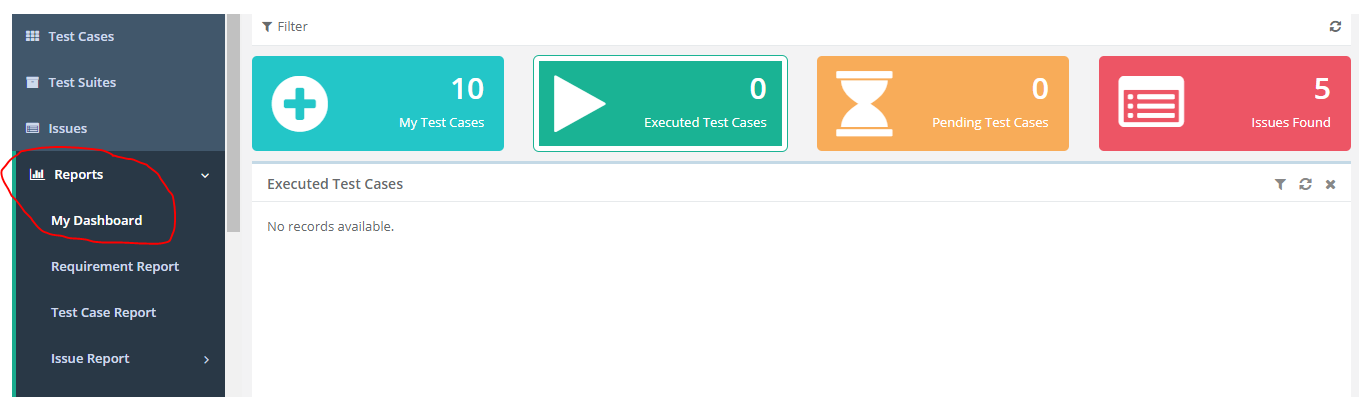
}

1. A new issue of type bug will be created under “Issues” module of Qmetry(21S-IS-10 in this case)

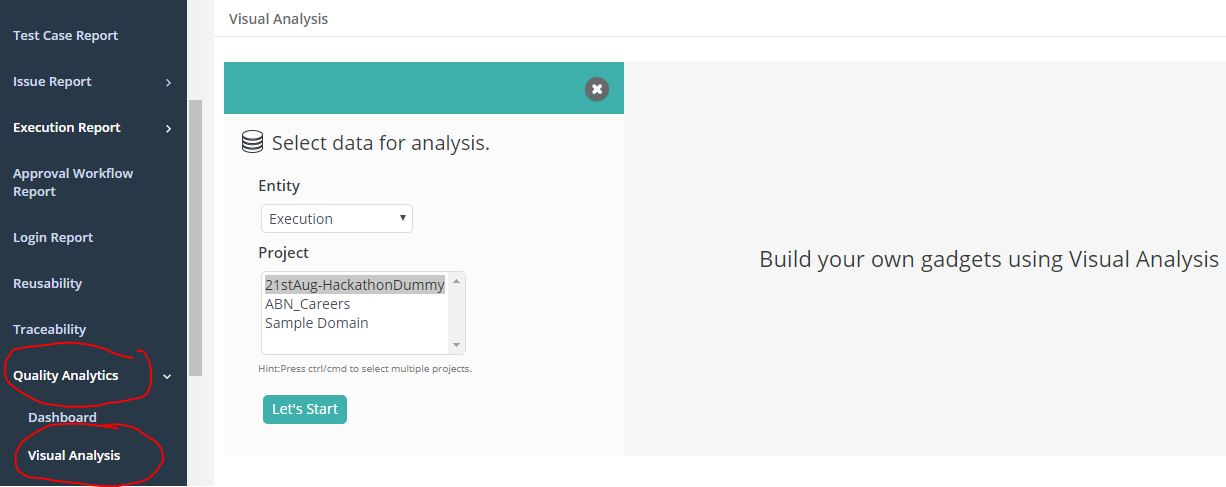


**Reports**

1. Reports > My Dashboard (Default Dashboard)

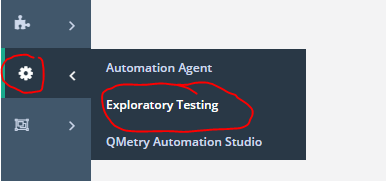


1. Reports > Quality Analytics > Visual Analysis(Custom Dashboard)

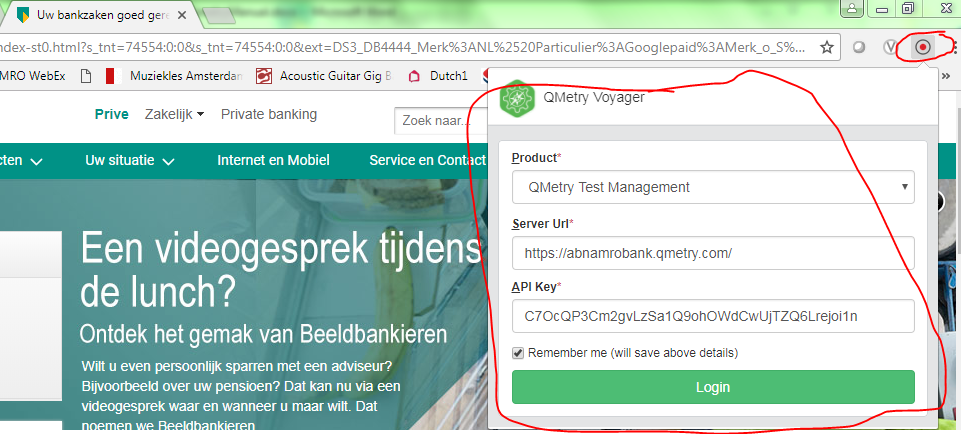


**Exploratory Testing(via Qmetry Voyager Chrome Add-on)**

1. Automation & CI/CD > Exploratory testing



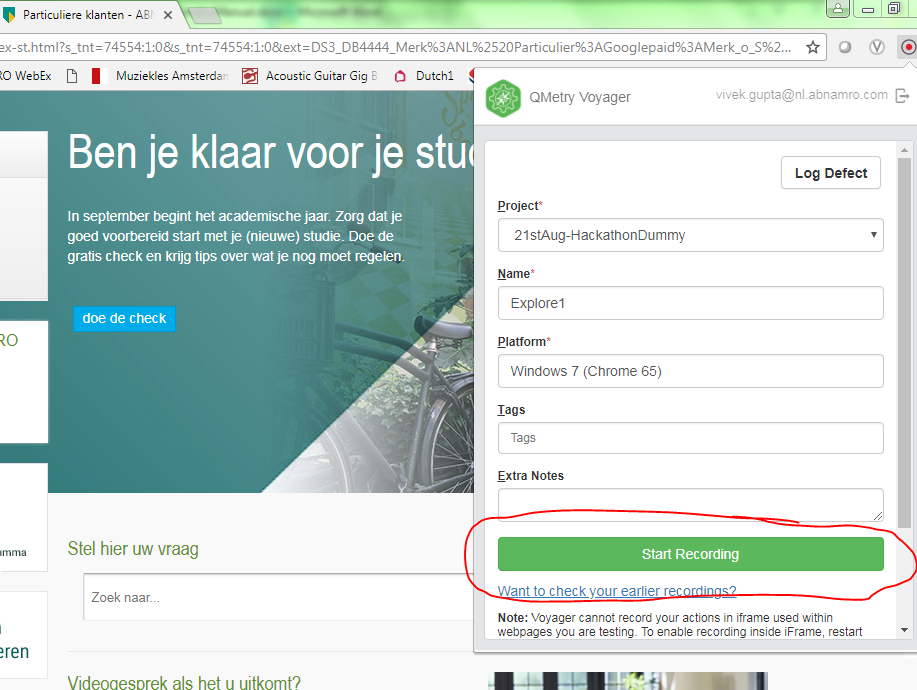
1. Following steps to be followed after adding Qmetry Voyager add-on to your chrome browser



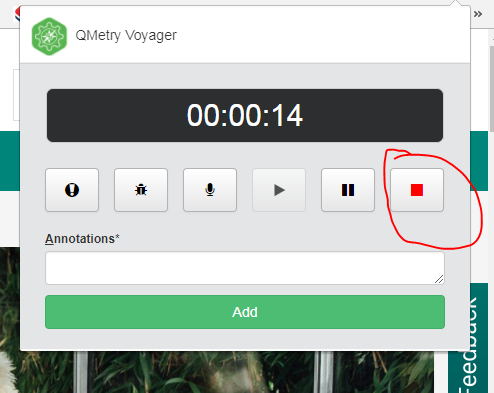
**(API key to be found under Administration > API Keys > Generate Exploratory API Key)**

Recording executed steps(User can also create bug while execution)

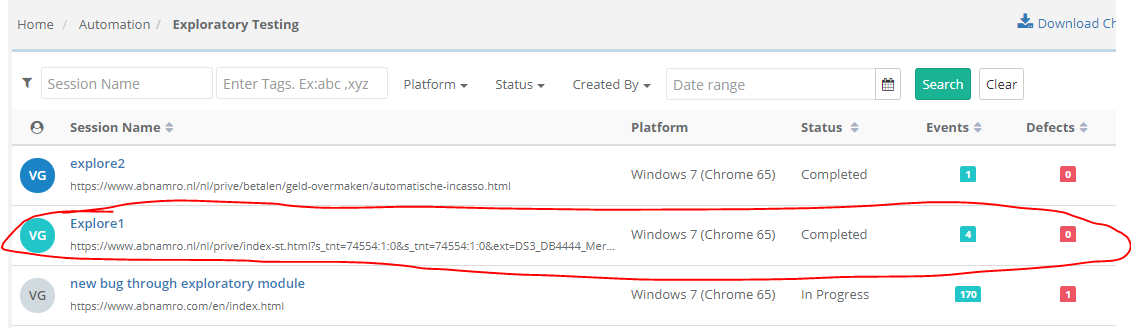
1. Fill in relevant information > Start Recording



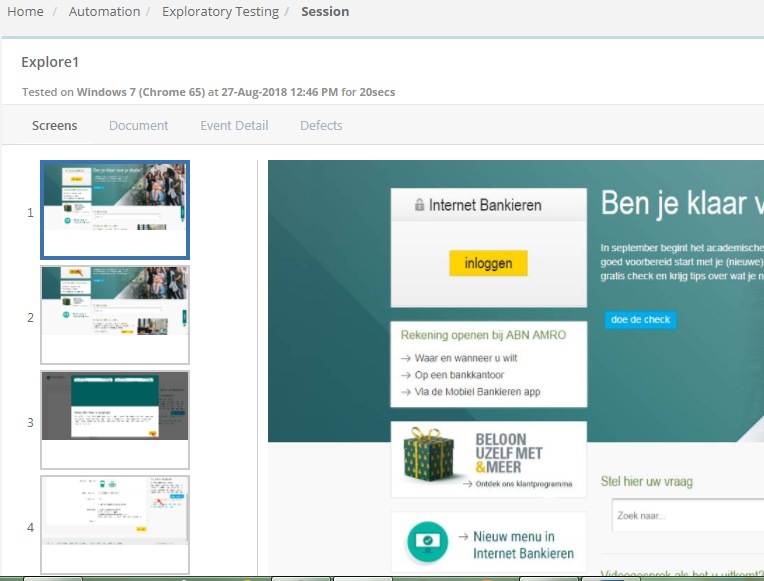
1. Click on Stop button once you have recorded the steps



1. Steps are recorded within exploratory testing module along with screenshots 1

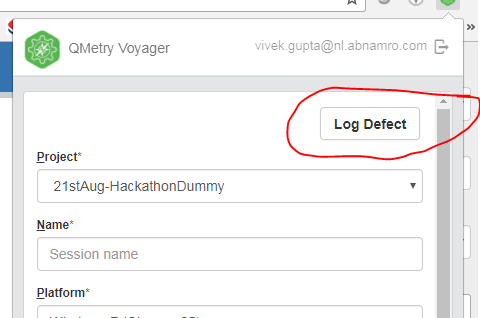


1. Steps are recorded within exploratory testing module along with screenshots

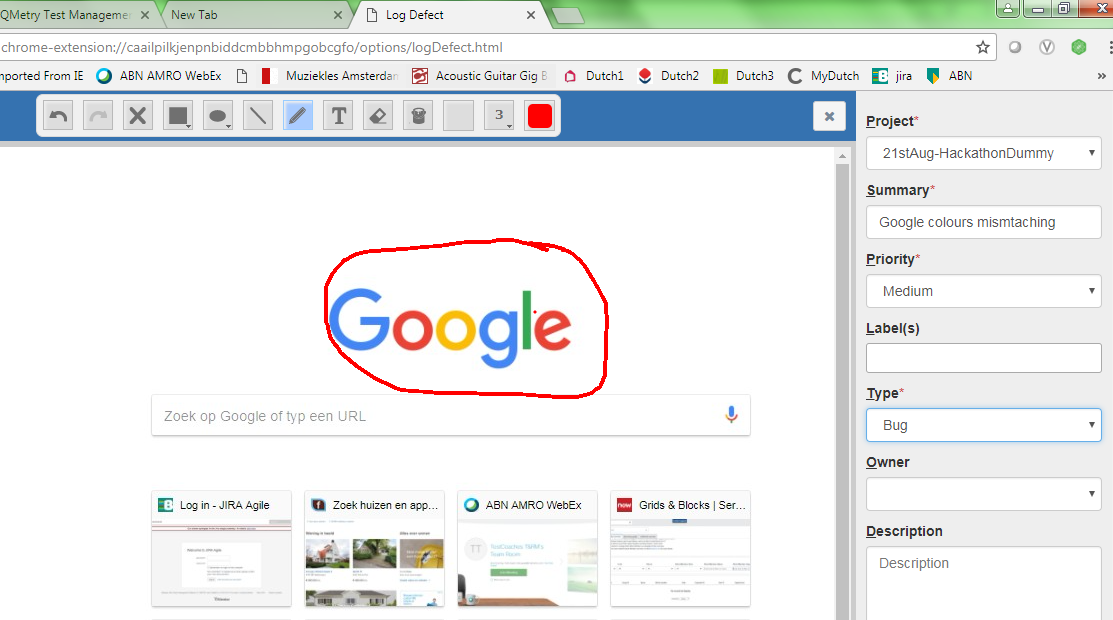


Logging a defect

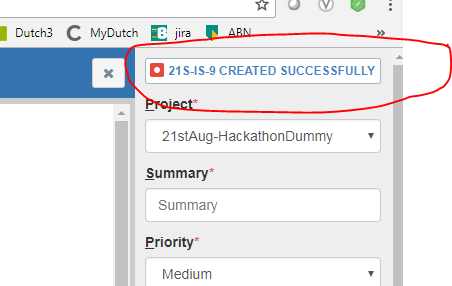
1. Click on Qmetry Voyager plugin on chrome browser



1. Fill in relevant information > Click on “Log Defect”



1. Bug is successfully logged



1. Details can be seen under issue module of qmetry

