Jira Introductory Course

Ing. Santiago Rodríguez Paniagua.
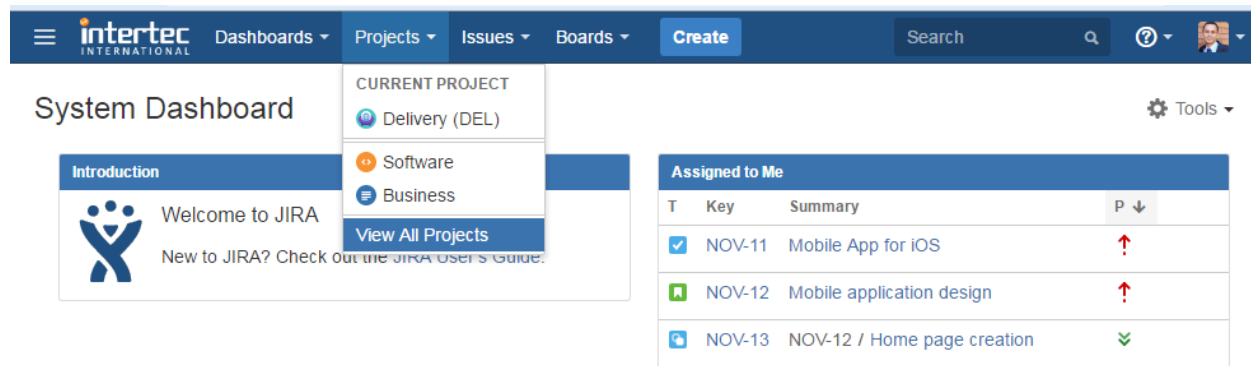
2016

Contents

JIRA Concepts

What is a Project

A JIRA project is a collection of issues, and is defined according to your organisation's requirements. For example, a JIRA project could be:

- a software development project
- a marketing campaign
- a helpdesk system
- a leave request management system
- a website enhancement request system

Every issue belongs to a project. Each project has a *name* (e.g. **The Company WebSite Issues**) and a *key* (e.g. **CompWeb**).

The project key becomes the first part of that project's *issue keys,* e.g. **CompWeb -101, CompWeb - 102,** etc:
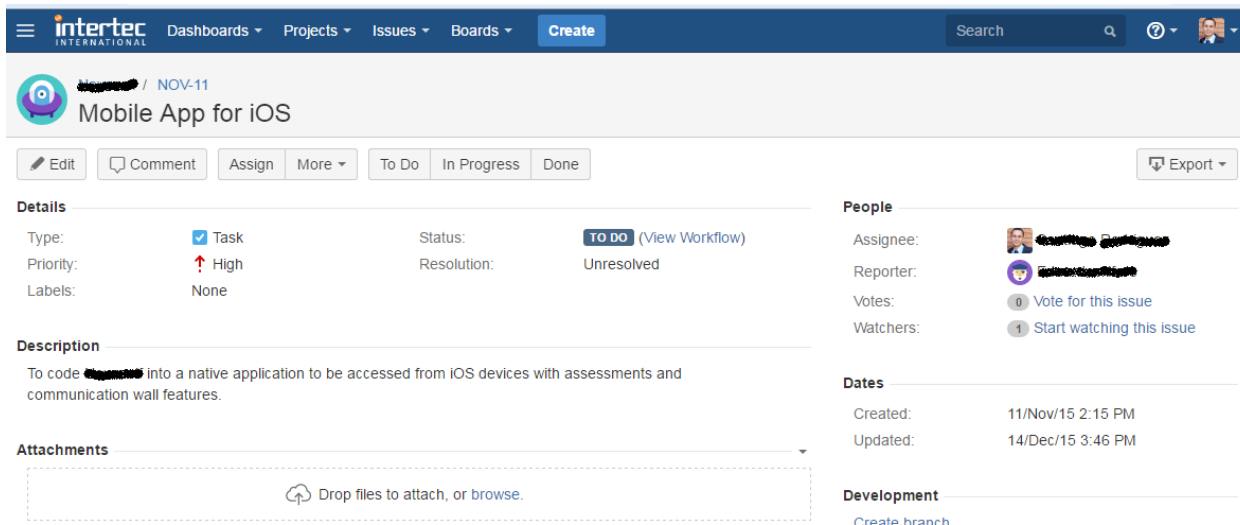
What is an Issue

Depending on how your organization is using JIRA, an issue could represent a software bug, a project task, a helpdesk ticket, a leave request form, etc.

JIRA can be used to track many different types of issues. The default types are listed below, but please note that your JIRA administrator may have customized this list to suit your organization.

Bug — A problem which impairs or prevents the functions of the product.
Improvement — An enhancement to an existing feature.
New Feature — A new feature of the product.
Task — A task that needs to be done.
Custom Issue — A custom issue type, as defined by your organization if required.

Getting Started
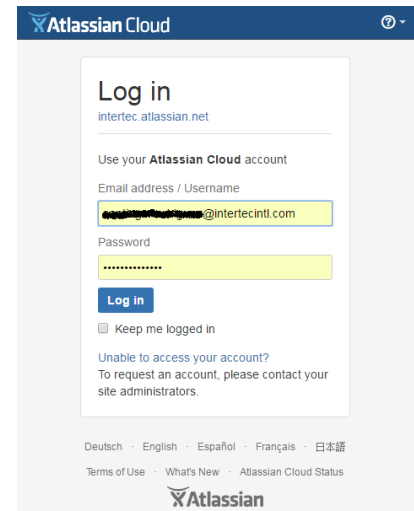
Logging in to JIRA

Enter the Jira page of your organization or client, click on the button login, enter your user and password.

Exploring the JIRA Workspace

The **Dashboard** is the first page you see (by default) after logging in to JIRA and basically it consists of:

- The navigation bar (at the top of the screen) is the same on every screen in JIRA. It contains links which give you quick access to many of JIRA's most useful functions.
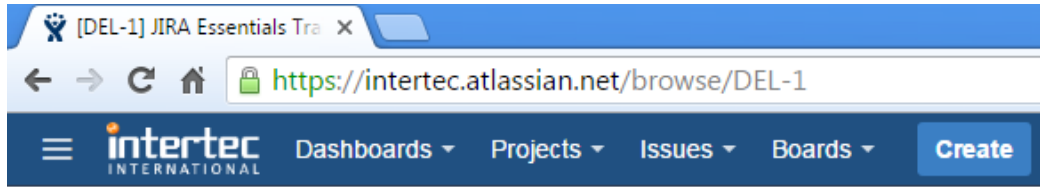
- The white area of the screen, below the top navigation bar, can be customized to display 'gadgets' showing many different types of information, depending on your areas of interest.

- Your JIRA dashboard page may look different from others, as the logo and colours may have been customized by your JIRA administrator also by the gadgets you add to your dashboard. The links in the navigation bar, however, will be the same.

Working with an Issue

Attaching a File

To attach a file to a JIRA issue:

1. Open the JIRA issue to which you wish to attach a file.
2. Select More > Attach Files.
3. The Attach Files dialog box is displayed.
4. Click Browse to search for your files.
5. Optional: Enter a comment about the files(s) you are attaching.
6. Click the Attach button. All selected files will be attached to the issue.
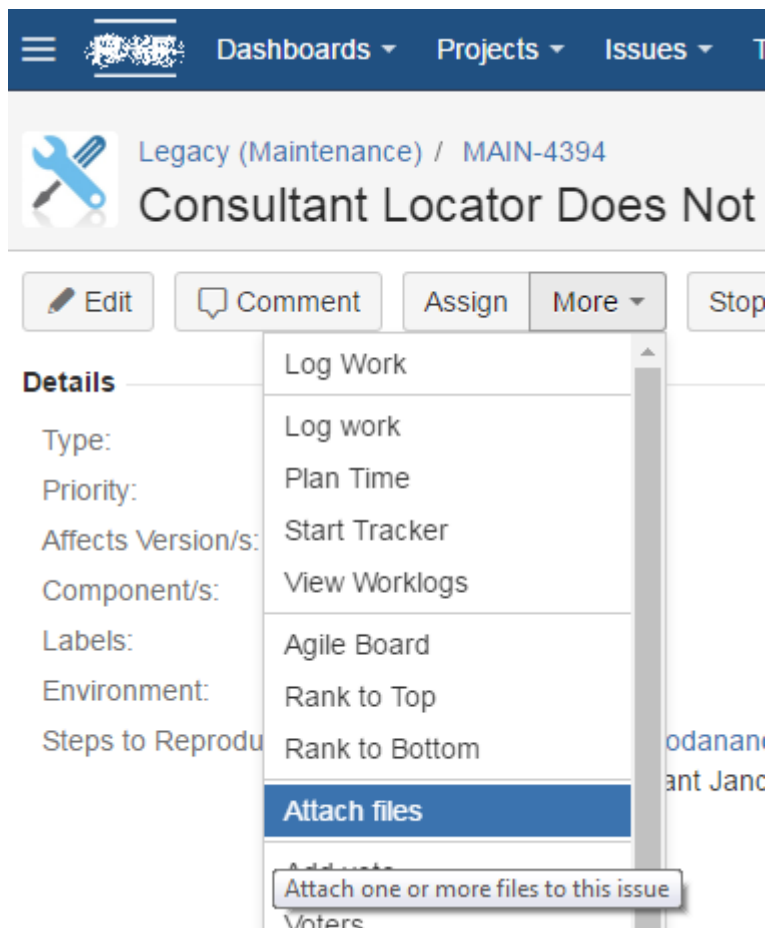
Attaching a Screenshot

1. Open the JIRA issue to which you wish to attach a file.
2. Select More > Attach Screenshot.
3. The Attach Screenshot dialog box opens. Note that if you take a screenshot at this point it will include the Attach Screenshot dialog box.
4. Paste the image from your clipboard using the relevant keyboard shortcut. (For Max OSX use CMD+V, for Windows use Ctrl+V)
5. Enter a file name for the screenshot you are attaching. (The file name defaults to screenshot-1.)
6. (info.) A valid file name cannot contain any of these characters: '\', '/','\"', '%', ':', '$', '?', '*'.
7. Click Upload to attach the captured image to your JIRA issue.
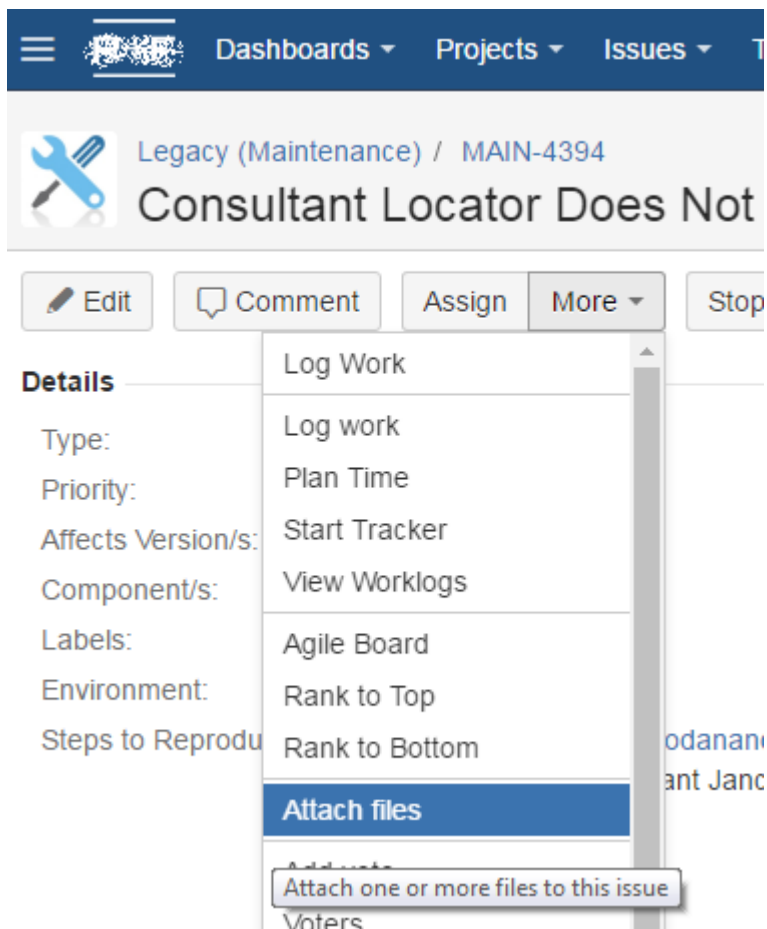
Cloning an Issue

Cloning, or copying, an issue lets you quickly create a duplicate of an issue within the same project. The clone issue is a replica of the original issue, containing the same information stored in the original issue.

A clone issue is a separate entity from the original issue. Operations on the original issue have no effect on the clone issue and vice versa. The only connection is a link – if created – between the original and the clone issue.

To Clone an Issue:

1. Open the JIRA issue you wish to clone.
2. Select **More > Clone.** The **Clone Issue** screen will appear.
3. You can edit the clone issue's **Summary** if you wish.
4. If the issue contains links to other issue(s), you can select whether or not to include the links in the new clone issue.
5. If the issue contains sub-tasks, you can select whether or not to create the sub-tasks in the new clone issue.
6. If the issue contains attachments, you can select whether or not to include the attachments in the new clone issue.
7. Click **Create**.

Comenting on an Issue

Adding comments to an issue is a useful way to record additional detail about an issue, and collaborate with team members. Comments are shown in the Comments tab of the Activity section when you view an issue.

Note:

1. When adding a comment, you can set the comment to be Viewable by members of a particular project role or user group only; or you can allow all users to view it.
2. For users to view a comment, they must have the Browse Project project permission to view the issue and for each comment, they must be a member of the Viewable by users (see Adding a Comment below).
3. You automatically become a watcher of the issues that you comment on. You can disable this via the Preferences > Autowatch option in your profile.
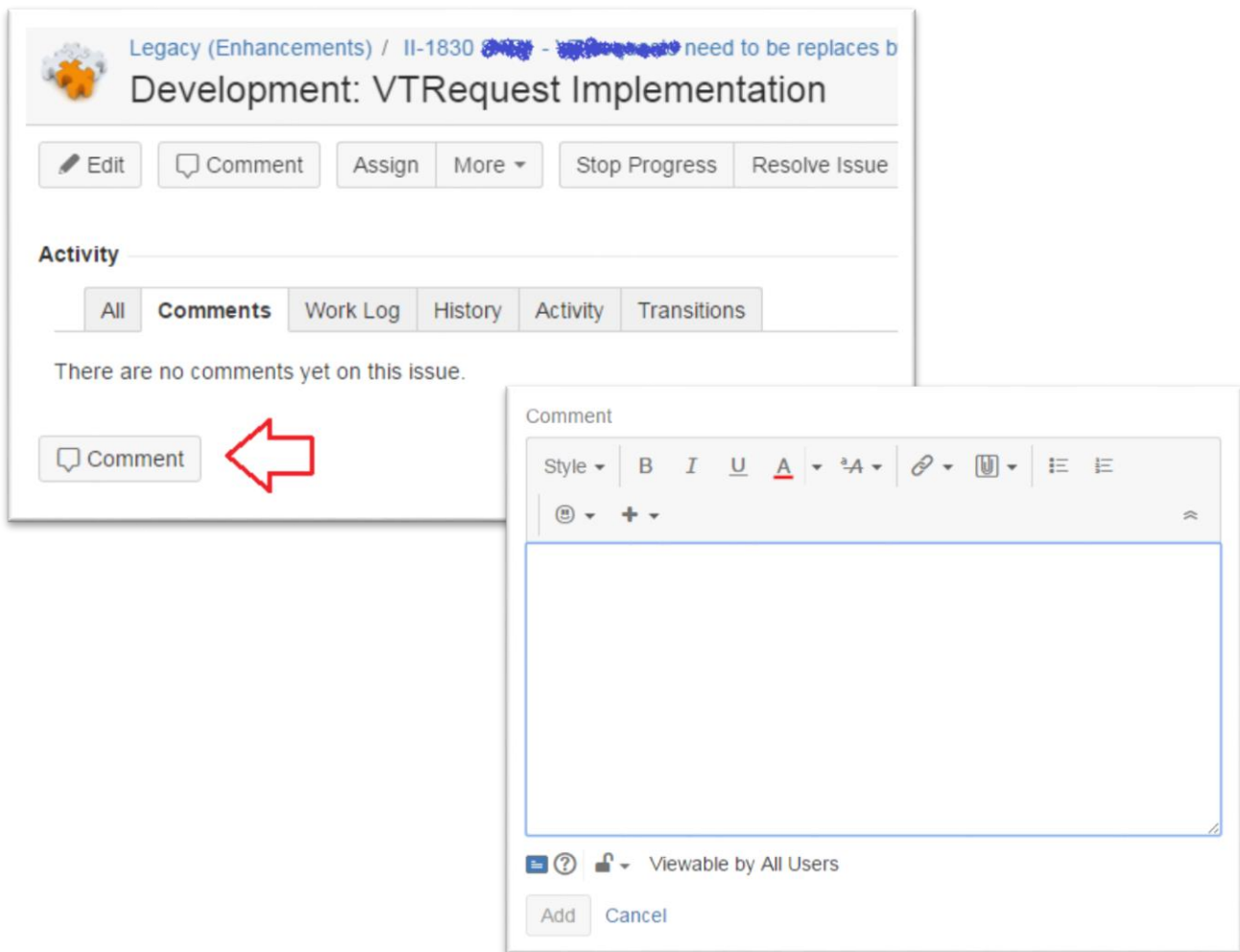
Comment standard for Jira issues

Normally there are four main steps before sending a ticket to UAT. The idea of these advices is improving the way we document the tickets adding colors and standard sections to help the stakeholders to understand the information in less time. Respect the header colors for each of the categories.

Analysis

This phase embraces all the tasks related to; validate description and steps to reproduce, try to reproduce the bug and research. Also add detailed steps about how did you reproduce it.

Development

Related to code Fixing, sql script implementation, attaching to the solution images, pdf documents, etc, config configuration, etc.

Code Review

A sort of any tasks required to ensure development was succeeded. Source code properly committed or merged, complete and consistent. Verify correctness of the code. Run an acceptable and minimum test to check if the expected result is according to the specifications.

QA
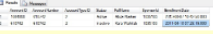
Verification of test cases and documentation of the results.

For those activities related to demonstrate a behavior, something fails, etc., it is imperative attaching screenshots, intermediate results (for example sql queries and the retrieved data), videos, transact-sql, etc. The idea is describing clearer those scenarios to everyone.

Samples of comments for each step are included below. The editor notation is besides each comment screenshot.

Analysis

The sections required for an analysis comment are: **Research and Conclusions**.

Look and Feel	Text Formatting Notation

ANALYSIS
 	
Research
First at all I noticed both accounts have the same AccountNumber '610742' and one of them has '610742' as account id too. The accounts corresponds to Alicia and Kara. Kara's account is inactive. See below the details:
Both accounts share the same SponsorId (1035103, Gretchen Lautenschlager).
 	
    select AccountID, AccountNumber, AccountTypeID,
    Status = Replace(Replace(Active, 0, 'Inactive'), 1, 'Active'),
    FullName = FirstName + ' ' + LastName,
    SponsorId, EnrollmentDate
    from Accounts
    where AccountNumber = '610742'
 	
    See image result
 	
-------------------- 	
In fact if we look for that OrderNumber in nscore4 we will recover its information under Kara's name.
 	
See it -->
 	
This information is retrieved by SearchOrder store procedure.
Conclusion
  ■ Data in OrderCustomers table should be updated using the Alicia's AccountId
	{panel:title=ANALYSIS|borderStyle=dashed|borderColor=#ccc|titleBGColor=#CCFFCC|bgColor=#FFFFCE}
	
	Research
	
	First at all I noticed both accounts have the same AccountNumber '610742' and one of them has '610742' as account id too...
	
	{quote}
	
	    select AccountID, AccountNumber, AccountTypeID,
	
	        Status = Replace(Replace(Active, 0, 'Inactive'), 1, 'Active'),
	
	        FullName = FirstName + ' ' + LastName,
	
	        SponsorId, EnrollmentDate
	
	    from Accounts
	
	    where AccountNumber = '610742'
	
	    See image result !result.jpg|thumbnail!
	
	{quote}
	
	-------------------- 	
	
	In fact if we look for that OrderNumber in nscore4 we will recover its information under Kara's name.
	
	See it --> !sample1.png|thumbnail!
	
	This information is retrieved by SearchOrder store procedure.
	
	Conclusion
	
	- Data in OrderCustomers table should be updated using the Alicia's AccountId
	
	{panel}

Development

Sections required for Development comments are: **Root Cause and Propose Fixes**.

Look and Feel	Text Formatting Notation
DEVELOPMENT **Root Cause** Duplicate Account numbers causes 'erratic' behavior in search account features. **Propose Fixes** 1) Update the OrderCustomer update OrderCustomers set AccountId = 1084868 where OrderId = 5221600	{panel:title=DEVELOPMENT|borderStyle=dashed|borderColor=#ccc|titleBGColor=#99CCFF|bgColor=#FFFFCE} *Root Cause* Duplicate Account numbers causes 'erratic' behavior in search account features. *Propose Fixes* 1) Update the OrderCustomer bq. update OrderCustomers set AccountId = 1084868 where OrderId = 5221600 {panel}

Code review

Sections required for comments about the code review result: Notes. See example below.

Look and Feel	Text Formatting Notation
CODE REVIEW **Notes** - Code looks complete and compiles in Release x.x.x. ✓ - However, class *Account*, ln 200, local variable 'string username' is being created inside the foreach with every item into the accounts list ✗ - It seems to be a missing image from the folder 'xxxxxx'. ✗ - Link does not redirect to the path according the bug description. ✗ - Tested in Staging using, Browsers: IE 8|9|10, FF23.0.1. ✗	{panel:title=CODE REVIEW|borderStyle=dashed|borderColor=#ccc|titleBGColor=#FFCC99|bgColor=#FFFFCE} *Notes* - Code looks complete and compiles in Release x.x.x. (/) - However, class _Account_, ln 200, local variable 'string username' is being created inside the foreach with every item into the accounts list (x) - It seems to be a missing image from the folder 'xxxxxx'. (x) - Link does not redirect to the path according the bug description. (x) - Tested in Staging using, Browsers: IE 8|9|10, FF23.0.1. (x) {panel}

QA Verification

Required section: **Steps to Verify this ticket**. See examples below.

Look and Feel	Text Formatting Notation
QA Verification **Steps to Verify this ticket:** - Correct Image is uploaded in the EmailImage folder? ✅ Yes, correct! - The project contains a reference to the image? ✅ Yes, correct! - SQL script updates PCNotice email using the new Image? ✅ Yes, correct! - PCNotice email body is updated in staging database? ✅ Yes, correct! - Email is generated correctly? ✅ Yes, correct! The mail was generadted and set successfully and it includes the september-october image now "Your journey to great skin continues – Your PC Perks Order is being prepared…" See Il_976-EmailShowsTheRightImage.jpg attached.	{panel:title=QA Verification|borderStyle=dashed|borderColor=#ccc|titleBGColor=#F7D6C1|bgColor=#FFFFCE} *Steps to Verify this ticket:* - Correct Image is uploaded in the EmailImage folder? (/) {color:green}Yes, correct!{color} - The project contains a reference to the image? (/) {color:green}Yes, correct!{color} - SQL script updates PCNotice email using the new Image? (/) {color:green}Yes, correct!{color} - PCNotice email body is updated in staging database? (/) {color:green}Yes, correct!{color} - Email is generated correctly? (/) {color:green}Yes, correct! {color} The mail was generadted and set successfully and it includes the september-october image now "Your journey to great skin continues – Your PC Perks Order is being prepared…" See [II_976-EmailShowsTheRightImage.jpg] attached. {panel}

Creating an Issue

To create a JIRA issue, you need the Create Issue project permission for the issue's relevant project. If you do not have this permission, please contact your JIRA administrator. To create a new JIRA issue:

1. Click Create at the top of the screen to open the Create Issue dialog box.

2. Select the relevant Project and Issue Type on the Create Issue dialog box.

3. Type a Summary for the issue and complete any appropriate fields — at least required ones which are marked by an asterisk. If you want to access fields that are not shown on this dialog box or you want to hide existing fields:
   a. Click the Configure Fields button at the top right of the screen.
   b. Click Custom and select the fields you want to show or hide by selecting or clearing the relevant check boxes, respectively, or click All to show all fields.

4. When you next create an issue, JIRA remembers your last choice of selected fields.

5. Optional: To create a series of similar issues – with the same Project and Issue Type – select the Create another check box at the bottom of the dialog.

6. When you are satisfied with the content of your issue, click the Create button.

Creating a Sub-Task

Sub-task issues are useful for splitting up a parent issue into a number of smaller tasks that can be assigned and tracked separately.

This can provide a better picture of the progress on the issue, and allows each person involved in resolving the issue to better understand what part of the process they are responsible for.

To create a sub-task:

1. Navigate to the issue you would like to be the parent issue of the sub-task you are about to create.

2. Select More > Create Sub-Task. You will see the Create Subtask screen.

3. Fill in the details as needed, and then click Create at the bottom of the page.
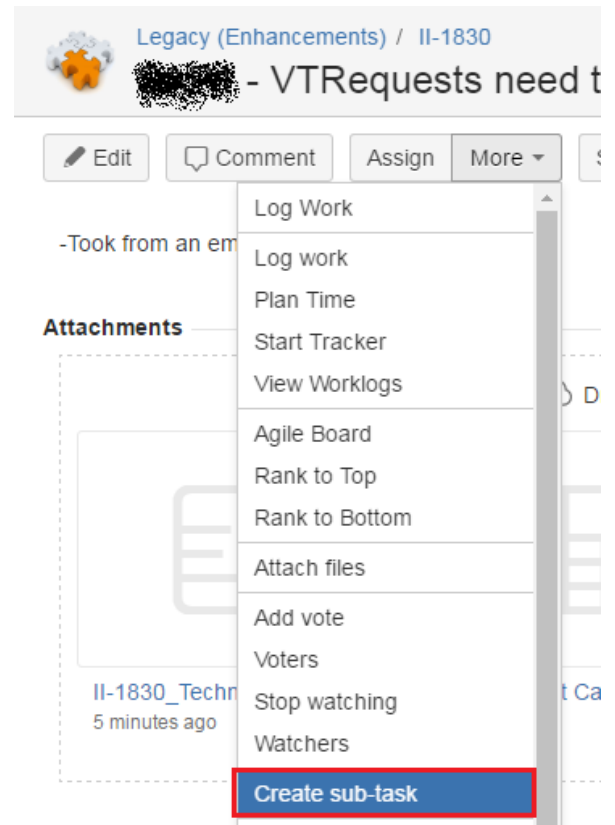
Linking Issues

Issue linking allows you to create an association between two existing issues on either the same or different JIRA servers. For example:

- An issue may *relate to* another.

- An issue may *duplicate* another.
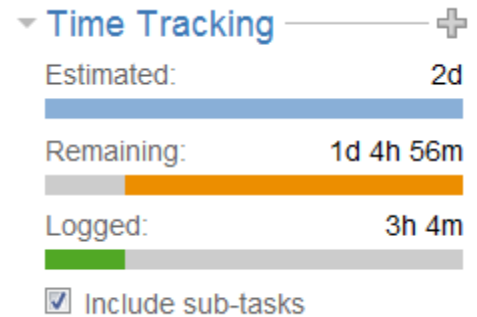
- An issue may *block* another.

Logging Work on an Issue

If an issue (or its sub-tasks) has had work logged and/or an **Original Estimate** value specified, three coloured bars will be displayed representing the following amounts of time:

- **Original Estimate** (blue) — the amount of time originally anticipated to resolve the issue. (This is indicated as**Estimated** when viewing an issue.)
-
- **Remaining Estimate** (orange) — the remaining amount of time currently anticipated to resolve the issue. (This is indicated as **Remaining** when viewing an issue.)

- **Time Spent** (green) — the amount of time logged working on the issue so far. (This is indicated as **Logged** when viewing an issue.)