Creating an Issue in Jira

To create a new issue:

1. Click the '**Create Issue**' link at the upper-right corner of the screen, just below the Quick Search.
2. The '**Choose the project and issue type**' popup will be displayed. Select **DataSite 7**, the issue type, and then click the '**Create**' button.
   a. Production Issue – Issues/defects found in Production
   b. Development Issue – Issues/defects found during development
   c. Enhancement – An improvement to an existing feature
   d. Technical Debt – Used by the Development and QA Teams to record or track tasks that need to be done to resolve an issue or to complete work for a feature.
3. The '**Enter the details of the issue**' screen will be displayed. Type a summary of the issue and complete any other required fields, which are italicized and highlighted by an asterisk.
   a. ***Summary:** provide a "brief, but insightful" summary describing the issue. A good Summary should not be more than 50 – 60 characters. It should quickly and uniquely identify a bug report as well as explain the problem.
   b. **Component/s:** Select the related component(s) that are affected by the issue.
   c. **Priority:** Select the appropriate priority for the issue. If you are unsure of the correct priority use the default setting of "Minor"
      1. Blocker - Blocks development and/or testing work, production could not run.
      2. Critical - Crashes, loss of data, severe memory leak.
      3. Major - Major loss of functionality in a component.
      4. Minor - Minor loss of functionality or there is an easy workaround for the problem.
      5. Trivial – Cosmetic problems like misspelled word or misaligned text/graphics,
   d. **Release Priority:** used to determine the priority in which items need to be fixed in the release. This field will set by the BA Team or by the Defect Review Team (DRT). QA and Dev may also set this field if the issue occurs during release.
      1. Required – Must be resolved for release. Required functionality or showstopper issue.
      2. High – Most likely will be resolved for release. Something that is really difficult to workaround or does not have a workaround. Frustrating to just about all clients.
      3. Medium – Should be resolved for release. Something that you'd like fixed in the release. There is a workaround, could be resolved with release notes, or the issue is something that will not be commonly seen by clients. Frustrating to most clients.
      4. Low – May be resolved for release. Something that is important but we can live without for the current release. There is an easy workaround for the issue.
      5. TBD – Might be resolved for the release. If we get time then we should look at it.
   e. **Browser:** Select the default "Any". If the issue is Browser specific, select the browser(s) in which the issue occurs.
   f. **Operating System:** Select the operating system that the issue occurs on. The default is set to "Windows".
   g. **Language:** Select the default "Any". If you know that the issue is Language specific, select the language(s) in which the issue occurs.
   h. * **Description:** Detailed restatement of the summary.

      Be specific – the quicker the engineer or the Defect Review Team can isolate the issue to a specific problem, the more likely it will be expediently fixed.

Include the following information if available:

Build Version: if the defect was found on dev

Project: include name, number, and ID

Role: User, SiteAdmin or Merrill

Test URL: if the defect was found on Dev

i. ***Steps to Recreate:** Note whether the issue is reproducible or intermittent. If you are able to reproduce the issue, record the steps to recreate it. Try to limit these steps to a minimum set required to reproduce the problem. Include any special setup steps. The Steps to Recreate should be the minimized, easy-to-follow steps that will trigger the bug.
j. **Current Results:** Describe what the application did after performing the Steps to Recreate.
k. ***Expected Results:** Describe what the application should have done, were the bug not present.
l. **Assignee** - choose Defect Review Team if this is a current issue found in Production. If the issue is related to the feature that is currently in Development set the issue to unassigned
m. **Affected Version/s** - Select the version(s) where the issue was observed.
n. **Fix Version/s** - choose the default "Unknown" unless you know the version where the issue is expected to be fixed. If the issue was found in the current version that you are testing and is not a production issue set it to the current testing version.
o. **Attachment** - use this tool to attach any relevant files to the issue; e.g., (screenshots, error code text, diff, logs, etc.).

   1. You should use attachments rather than comments, for large chunks of ASCII data, such as trace, debugging output files, log files, email conversations.

   2. You should trim screenshots. There's no need to show the whole screen if you are pointing out a single pixel problem.

p. **Notes from Business Analyst** – Business Analyst will fill this out if needed.
q. **Perforce** – Developer will set this.
r. **Development Notes** – Developer will fill this out if needed.
s. **Testing Instructions** – Developer will fill this out if needed.
t. **Dev Est.** – Developer should enter an estimate on how long it will take to fix the issue.
u. **QA Est.** – QA Analyst should enter an estimate on how long it will take to test the issue.
v. **Related to Feature/Proposal** – Business Analyst will add link to user story or requirements document.
w. **ISMS** – Used to identify items that directly affect certain areas of Information Management Security System (ISMS).
x. **Story Points** - story points are units of relative size used in estimating requirements as an alternative to units of time. Story Points are determined by the Development and QA Teams.

4. After completing the form, click the **Create** button to submit the issue.