How to Report a Bug or Make a Feature Request

Overview

Spotting and reporting genuine bugs in Titanium is one of the most accessible ways to contribute, but there are a few challenges to writing a good bug report. Use this guide as reference to help ensure that your bug gets looked at promptly.

Introducing JIRA

Titanium bugs are tracked using the Atlassian's JIRA Issue Tracker, at http://jira.appcelerator.org.

Searching JIRA

Before reporting a bug, you should try to determine whether it's already been filed.

Tickets for outstanding issues often contain workarounds or other relevant information. For fixed bug, a ticket tells you when a fix for a bug was created and the build you need to obtain in order to take advantage of it. A useful feature is the ability to "watch" issues, which means you will be notified by email of any developments.

For information on searching JIRA, see Atlassian's official Searching for Issues guide. You'll see when you run a simple search on our server.

For more control over the results, the advanced search interface is very useful. See Atlassian's documentation for the JIRA Query Language (JQL), for information on entering complex search queries.

Projects overview

To help organize the information, every JIRA ticket is assigned to one of a small number of Appcelerator projects. The following table shows the project permissions that community members are granted by default.

JIRA Project Permissions for Community Members

Project	Key	View	Create	Comment
Aptana Studio	APSTUD	✓	☒	✓
Appcelerator - INBOX	AC	✓	✓	✓
Documentation	TIDOC	✓	☒	✓
Titanium SDK & CLI	TIMOB	✓	☒	✓
Alloy	ALOY	✓	☒	✓
Appcelerator Studio	TISTUD	✓	☒	✓

You can drill down via the Browse Projects page to obtain a summary of the recent achievements of each project.

Note: for Axway employees, you will be given individual access (as needed) that will grant you the basic rights such as viewing, creating, and commenting on select JIRA projects. If you need to be granted rights to certain projects, please contact our DevOps team.

Appcelerator - INBOX project

You will notice from the above table that only the Appcelerator Community project is open for ticket creation. All new tickets are created in the AC project so that our triage may check them for validity and completeness. Once a ticket is validated, it is moved to the appropriate project, ready to be reviewed and prioritized by the engineering teams.

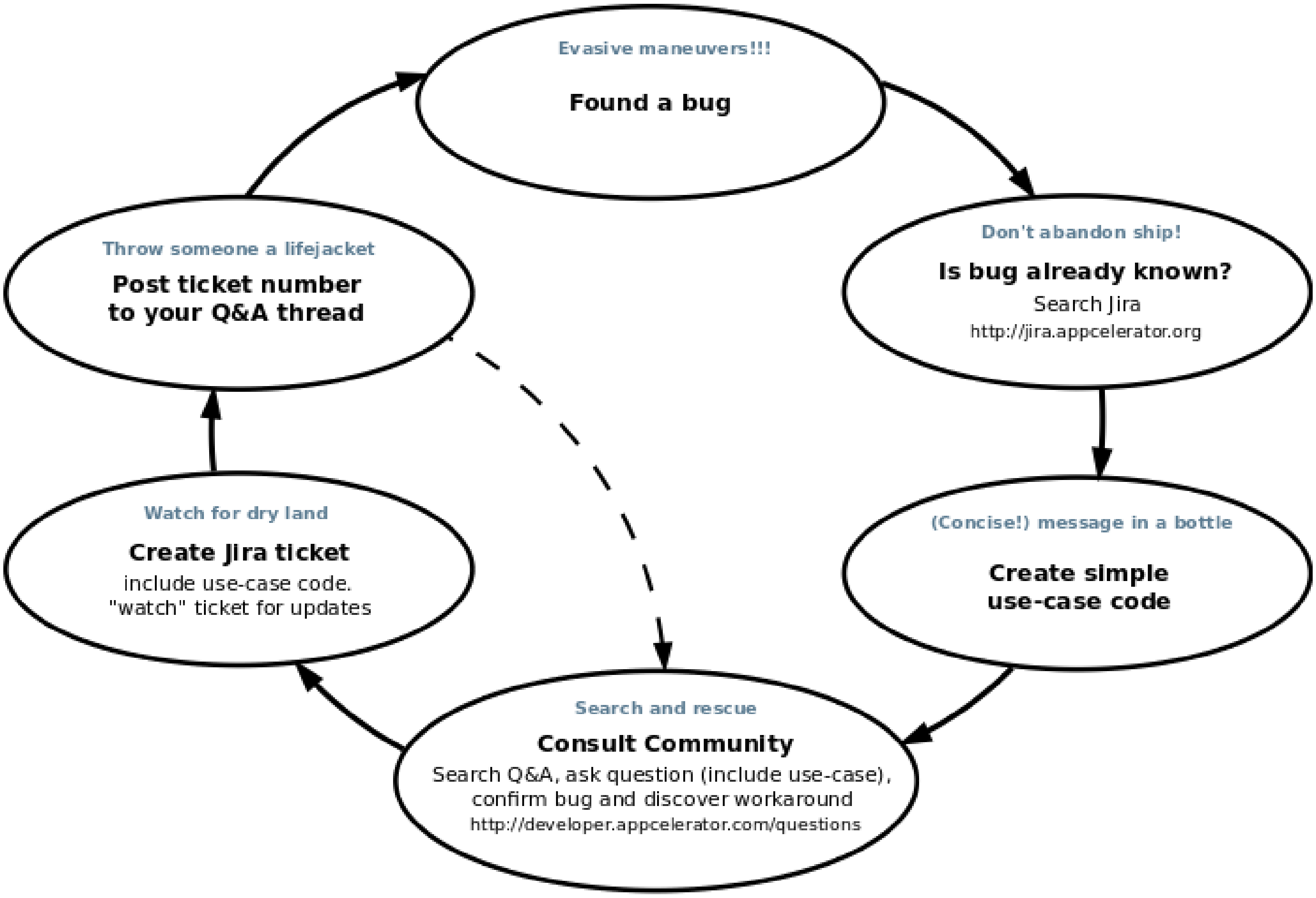
If you are new to Titanium, we understand if you are unfamiliar with the process of raising a well-defined ticket. Don't worry; we're here to help. Once you've gone through the process once or twice, the time it takes to write a good ticket drops considerably.

Submitting tickets

The bug-reporting lifecycle

Community-reported bugs are crucial to improving the Titanium platform. However, managing tickets can consume considerable resources, so inaccurate or duplicate submissions take time away from the crucial work of improving the code.

The most useful tickets are concise, complete and objective. If you are tempted to ask a question in a JIRA ticket, consider using Community Support instead.



To summarize the process you should follow if you encounter unexplained Titanium behavior:

1. **review the documentation**
2. **search JIRA** for similar known issues
3. **search the Q&A** for discussions about the issue between our community members
4. **create a test case** (it's a prerequisite for posting to the Q&A and JIRA). Follow the guidance in Creating a Test Case
5. **post a question in the Q&A** to garner the community's opinion about the issue and collaborate on a workaround. Include your test case with the rest of the information.
6. **create a JIRA ticket**, only after there is no doubt about the bug's existence, Create one ticket per bug, bearing the following in mind:
    - **keep to the point:** concise, complete and factual tickets are the most appreciated
    - **include all the information:** without a test case and all the information described in the JIRA Ticket Checklist, we cannot accept your ticket
    - **watch your ticket:** This makes us aware of your vote for the ticket's quick resolution
      NOTE: Your JIRA login is different than your AMPLIFY Appcelerator Services login. If you do not have a JIRA login with us, please select the link under the login boxes to contact the JIRA administrators and request a login. Logins typically provided within one business day.
7. **post the ticket's URL to your Q&A question.** Add this as a comment under the question that helped you most and mark it as "best" to

make it easy for everyone to find

Creating a test case

A "test case" is working code that reliably reproduces a specific bug. What sets the *quality* test cases apart from the rest is that they contain the *least possible* code while still demonstrating the bug. Take pride in your test case; make it an art!

Being in the habit of writing test cases as part of your troubleshooting process can significantly increase your coding efficiency. This is because it can differentiate user coding errors from bugs in the platform.

As a simple example, you observe in your app that the `layout:'vertical'` property of the Titanium.UI.Window object does not work as expected for its child views. In the process of producing a test case, you remove all the properties from both the window and its sub-views except for those that are absolutely necessary. With less code, it is often easier to notice a mistake and, in this case, you realize that the child view's top property had been inadvertently set, which caused the views to be positioned absolutely rather than relatively.

Syntax errors can be the cause of unusual behavior too. Therefore, there is no point escalating an issue to the community or JIRA until you have ensured it validates properly. Of course, our Axway Appcelerator Studio IDE can help you correct these types of problems, with its built-in Javascript validator. Otherwise. you could use an online tool such as JSLint.

In summary, to create a test case:

- **use a sequential set of Titanium API calls:** Don't worry about polluting the global scope; there is no need to use namespaces or custom functions. A test case should be too small for this to be significant
- **remove any superfluous objects:** this includes UI elements such as windows, views and controls
- **use default values where possible:** remove any properties that are not absolutely necessary
- **use a Javascript validator:** ensure that your code is free of syntax errors
- **ensure your code runs without any modification:** test your code by copying it into a blank `app.js` file and launching

A good read on this is Eric Lippert's How to debug small programs.

Creating a JIRA ticket

Assuming you have consulted the community via the Q&A, and you have your test case to hand, you are ready to create a ticket:

- open the JIRA Titanium Community project page. See the #Projects Overview section for more information about the projects.
  NOTE: Your JIRA login is different than your AMPLIFY Appcelerator Services login. If you do not have a JIRA login with us, please select the link under the login boxes to contact the JIRA administrators and request a login. Logins typically provided within one business day.
- select from the list of ticket types in the top right-hand corner, although usually only **Bug**, **Improvement** or **New Feature** will be applicable
- complete as many fields as you can. Use the JIRA Ticket Template as a starting point, and refer to the #JIRA Ticket Checklist below for the information that is required
- as the reporter, you will automatically receive any updates to the ticket. However, it's always a good idea to also watch all tickets where the resolution is crucial to your projects. This helps our team gauge the issue's impact on the community, from the total number watchers in the list

JIRA ticket checklist

The following table explains what information must be included in which fields in order for a ticket to be accepted.

> If you have been directed to this section after creating a ticket in JIRA, it likely means that you have omitted some information or not provided it in a suitable format. Please correct the ticket using the instructions below, to enable us to pass it to our engineering teams.

Field	**Required**	Information to include
Summary	✓	**A concise summary** that distinguishes it from other tickets, making it easy for the core team to determine at a glance the nature of the issue. Format: `<COMPONENT>: <SUB-COMPONENT> - Description` *(eg. "iOS: Facebook - Unable to login on iPod")*
Assignee	✓	Leave as default
Reporter	✓	Leave as default

Severity	✗	Optionally, one of the following descriptions may be chosen to reflect the issue's impact on your project: • **Blocker:** issue is blocking project • **Major:** crashing or incorrect behavior issue that may have a workaround or is not stopping ship • **Minor:** crash or incorrect behavior that does have a reasonable workaround or low impact • **Trivial:** functions correctly, but appearance/functionality needs improvement • **None:** default
Components	✓	Choose a component from the list
Environment	✓	*The tested environment (only include the versions of each component that exhibit the problem and have been physically tested)* • **Titanium SDK version:** number, build date and build hash, or output to the console output at application launch • **Javascript Engine:** V8 • **Platform & version:** iOS x.x, Android x.x, OS X x.x, Windows x, and/or Ubuntu x.x • **Device Details:** iOS simulator, Android emulator, or physical device model • **Host Operating System:** OS X x.x, Windows x, or Ubuntu x.x • **Titanium Studio version:** available in Help > About
Description	✓	A concise, factual, description containing the following sections (use the JIRA Ticket Template as a starting point): **Problem** *Explanation of the problem, including the actual behavior and the expected behavior* **Test case** *Steps to reproduce plus simple working code that demonstrates the issue and follows the Creating a Test Case guidelines, placed within a {code} markup block in the body of the ticket* **Community Discussion** *A link to the Q&A question in which you asked the community to verify the observed behavior* **Log** *Logged messages from the following locations, depending on the problem, placed within a {code} markup block in the body of the ticket:* • **Android and iOS Development:** Titanium Studio console output of exceptions • **Packaging:** contents of `build.log` (`Android: <PROJECT>/build.log`, iOS: `build/iphone/build/build.log`) • **Titanium Studio:** contents of menu item `Help > Titanium Studio > View Log File` (contents of `<workspace-dir>/log` file)
Attachment	✗	Attach images and assets, but try to **keep code in the description field**
Labels	✗	The only valid labels are `regression` for regressions and `parity` for differences between platforms. All others will be removed before the ticket is accepted.

We do not underestimate the time and conscientiousness involved in raising good tickets, and we very much appreciate it. Thank you for all your efforts!

If you find a ticket that describes a bug that is important to your project, **watch the ticket!** This will notify you by email of any developments, and the total number of watchers helps us prioritize our work.

Related topics

- How to Submit a Bug Report
- JIRA Ticket Template