Troubleshoot an error

Table of Contents:

Error Log Files

- **DSpace log:** `[dspace]/log/dspace.log` (*usually*)
  - `[dspace]` is the location where DSpace is installed.
- **Tomcat log directory:** `[tomcat]/logs` (*usually*)
  - `[tomcat]` is the location where Tomcat is installed.

Resources

- DSpace Documentation - This would be the place to start for any installation or configuration problems
- Mailing Lists - Search the archives of our mailing lists for similar issues
- Directly search or browse the DSpace Issue Tracker for known/fixed bugs: https://github.com/DSpace/DSpace/issues and https://github.com/DSpace/dspace-angular/issues
- StackOverflow ("dspace" tag) - We now encourage technical questions to be asked on StackOverflow, so you may wish to search/browse there to see if anyone has encountered this problem before.
- Support - Where to go for support once you find the error message

General Instructions

DSpace 7.x or 8.x

Finding the detailed error message is important!

In DSpace 7.x or 8.x, there are two main places where detailed error messages may be found. It is *important to locate these detailed error messages* in order to debug any issues you are seeing. (The generic error messages that appear in the User Interface do not provide enough information to debug the problem.)

1. In your web browser's DevTools when running the User Interface - because the v7 User Interface is built with JavaScript (Angular.io), some errors are *only visible in your browser (and therefore will never appear in log files).* See "Finding the Error Message in the User Interface" below.
2. In the backend's log files - Some errors (especially 500 responses/errors) will result in the error message being logged to your dspace.log or Tomcat logs. See "Finding error messages in the REST API logs" below.

Once you locate the detailed error, see the section on "Report the Error and Describe How You Encountered It" below.

1. Finding the Error Message in the User Interface (Try this first!)

The User Interface is built with JavaScript (Angular.io). This means that some errors are *only visible in your browser (and therefore will never appear in log files).*

To find the errors in your browser, you will want to open your browser's Developer Tools *and then try to reproduce the error again.*

- In Chrome, go to "More Tools Developer Tools". Then try to reproduce the error back in your open browser tab.
  - The "Console" tab will provide a brief overview of errors that have occurred in this tab of your browser.
  - The "Network" tab will provide more information on *where* the error appeared (as the request that threw an error will appear in red). Clicking the red request will show you the exact Headers sent/returned and the error Response.
- In Firefox, go to "Web Developer Web Developer Tools". Then try to reproduce the error back in your open browser tab.
  - The "Console" tab will provide a brief overview of errors that have occurred in this tab of your browser.

    - The "Network" tab will provide more information on *where* the error appeared (as the request that threw an error will appear in red /purple). Clicking that request will show you the exact Headers sent/returned and the error Response.
- In Microsoft Edge, go to "More Tools Developer Tools". Then try to reproduce the error back in your open browser tab.
    - The "Console" tab will provide a brief overview of errors that have occurred in this tab of your browser.
    - The "Network" tab will provide more information on *where* the error appeared (as the request that threw an error will appear in red). Clicking the red request will show you the exact Headers sent/returned and the error Response.

If the User Interface error is a generic **500 response**, that means there's an error that occurred on your backend. Check your DSpace logs or Tomcat logs for information on the exact error. See the below section on "Finding error messages in the REST API logs".

2. Finding error messages in the REST API logs

Some (but not necessarily all) errors will result in logged errors on the backend. Similar to DSpace 6.x and below, the error message should be in either `[dspace]/log/dspace.log` OR `[tomcat]/logs/`.

Usually, the culprit error will appear in the logs with a line similar to one of the following:

- `yyyy-mm-dd time ERROR ...` OR,
- `yyyy-mm-dd time WARN ...`

3. Debug mode for the REST API (if necessary)

If you are not seeing the error appear in the REST API logs, and the error information is minimal in the User Interface, you can tell the REST API to always return the **entire error stacktrace** in any error response. This is done by temporarily adding the following to your local.cfg (on the backend):

```
# This tells the REST API to include a "stacktrace" param in error responses
# This param will include the full Java stacktrace of the error
server.error.include-stacktrace = always
```

Enabling this setting requires you to reboot your servlet engine (e.g. Tomcat).

Once this setting is enabled, you can reproduce the error from the User Interface. Now, the error message in your User Interface (in your browser's Developer Tools) may show the entire Java stacktrace that resulted in the error.

WARNING: Keep in mind that you should only enable this setting TEMPORARILY. Keeping this set to "always" is a security issue in Production scenarios, as it allows anyone to view the detailed stacktrace of any error they cause. These stacktraces can sometimes leak important information about your underlying system which can help hackers to find vulnerabilities.

4. Turning on Debug Logging for the REST API (if necessary)

If none of the above helped, or you are hitting a very strange backend error, you can change the DSpace logger settings to `DEBUG` which will sometimes provide you with more information about the error. To turn on debugging, visit the `[dspace]/config/log4j2.xml` file and do the following:

- To enable DEBUG logging in the `dspace.log` file, change the `loglevel.dspace` Property to `DEBUG` rather than `INFO`.
- **NOTE:** You'll need to restart Tomcat after enabling DEBUG mode in the `log4j2.xml` file.
- **WARNING:** Make sure to turn off debugging once you are finished. Leaving debugging turned on will cause the log files to grow very large very quickly!

5. Found an error message! Now what?

See "Report the Error and Describe How You Encountered It" section immediately after this.

Report the Error and Describe How You Encountered It

Before reporting the error

Before reporting any issue, it's best to check and see if the answer is already easily available.

- Search the DSpace Mailing Lists to see if someone else hit this before. If so, there may be a solution waiting for you.
    - Search the archives of the tech support list: https://groups.google.com/g/dspace-tech
    - Search the archives of the community list: https://groups.google.com/g/dspace-community
- If you recently installed (or reinstalled) DSpace, check to see if this is a Common Installation Issue listed at the bottom of the Install Guide

If neither of these resulted in a solution to your error, proceed to "Reporting the error" below.

Reporting the error

There are a few ways you may choose to report this error. However, we ask that you only **choose one**, depending on which you prefer.

1. Email a description of the error along with the error stacktrace (which you found above) to **dspace-tech@googlegroups.com**
    a. If you are not a member of this list, or want more information about DSpace lists, see Mailing Lists
2. OR, post a question on StackOverflow and **tag it with "dspace"**. See Guide to Posting Technical Questions to StackOverflow for more information.
3. OR, if you are *certain* you've found a bug in the codebase, please report it to GitHub Issues so that we can analyze it & schedule it to be resolved: https://github.com/DSpace/dspace/issues
4. Other support options detailed on the DSpace Support page

MAKE SURE to include the following information:

1. Your environment: Version of DSpace (most important), operating system, version of Tomcat, Java, Node, etc.
    a. NOTE: On the backend, simply running "`./dspace version`" from the command line will provide much of this information for you.
2. The exact steps you took to encounter the error, as this will help us in investigating the problem. Often, we need to understand how to replicate the error before we can determine what the underlying cause is.
3. The error stacktrace / message that you found in your log file or Browser DevTools

DSpace 6.x or below

Finding the Error Message in your Log Files

1. If the expected information simply never appears, leaving a blank browser screen, the problem is likely with either a servlet (for JSPUI) or an XSLT Theme (for XMLUI). The error message should be in either `[dspace]/log/dspace.log` OR `[tomcat]/logs/`.
    - For the XMLUI, you may also want to check the Cocoon logs to see if any extra information can be found there. Cocoon logs can be found in these locations:
        - DSpace 1.5.1 or previous: `[tomcat]/webapps/<name-of-webapp>/WEB-INF/logs/`
        - DSpace 1.5.2 or later: `[dspace]/log/cocoon.log`
2. If you get an "Internal Server Error" message, you will have to check the log files for a full error listing. If your email address is set up in dspace.cfg (`alert.recipient`) as the DSpace Administrator, you should receive an email with this full error listing. If not, move into the DSpace log directory (`[dspace]/log`) and view the end of the log file:
    - (Linux or Mac OSX) Go to the DSpace log directory via the command line and enter: `tail -100 dspace.log` Alternatively, you can open up the dspace.log file in your favorite text editor and look near the bottom of the file for the error message.
    - (Windows) Go to the DSpace log directory, and open dspace.log in your favorite text editor. Look near the bottom for the error message.
3. Usually, the culprit error is in the first five to ten lines of the error stack listing. The error stack starts with a line similar to one of the following:
    - `yyyy-mm-dd time ERROR ...` OR,
    - `yyyy-mm-dd time WARN ...`

Searching for Fixes to Common Errors

See "Report the Error and Describe How You Encountered It" section immediately above for details.

Turning on Debugging (optional)

- If you'd like to try and do some debugging yourself, you can change the DSpace logger settings to `DEBUG` which will sometimes provide you with more information about the error. To turn on debugging, visit the `[dspace]/config/log4j.properties` file and do the following:
    - To enable DEBUG logging in the `dspace.log` file, change the `log4j.rootCategory` and `log4j.logger.org.dspace` settings to `DEBUG` rather than `INFO`.
    - (*XMLUI Only*) To enable DEBUG logging in the `cocoon.log` file, change the `log4j.logger.org.apache.cocoon` setting to `DEBUG` rather than `INFO`.
    - **NOTE:** You'll need to restart Tomcat after enabling DEBUG mode in the `log4j.properties` file.
    - **WARNING:** Make sure to turn off debugging once you are finished. Leaving debugging turned on will cause the log files to grow very large very quickly!
- Depending on where you got your Java runtime environment, you may have the `jps` command. `jps -v` can show you the options actually used to run your Servlet container, which can be useful in debugging startup issues. A plain `jps` command will list process IDs of running JREs, which you can use with your favorite process monitoring tools. `jps` can only show processes which your user account is allowed to inspect, so you should run it as the user which runs the container, or as a superuser.