How does Exalate handle sync failures?

Last Modified on 02/17/2026 9:44 am EST

When a synchronization fails, **Exalate logs the error** and displays it in the unified console dashboard, offering detailed insights into what went wrong. This helps you quickly identify the issue and fix it without needing to sift through system logs.

Error logging and visibility in the unified console

The **error view** in the dashboard provides:

- The **connection** where the failure occurred
- The **specific work item** involved
- The **error message** and **line in your script** where the problem happened

This visibility makes troubleshooting more straightforward and saves time in resolving sync failures.

Queuing failed synchronization attempts

Failed sync attempts are not lost—they remain **queued** in the system until the underlying issue is fixed. The affected work items stay in a **pending state**, ensuring no data is lost. Once the cause of the failure (like a script error or connectivity problem) is resolved, you can **retry synchronization** for the affected items.

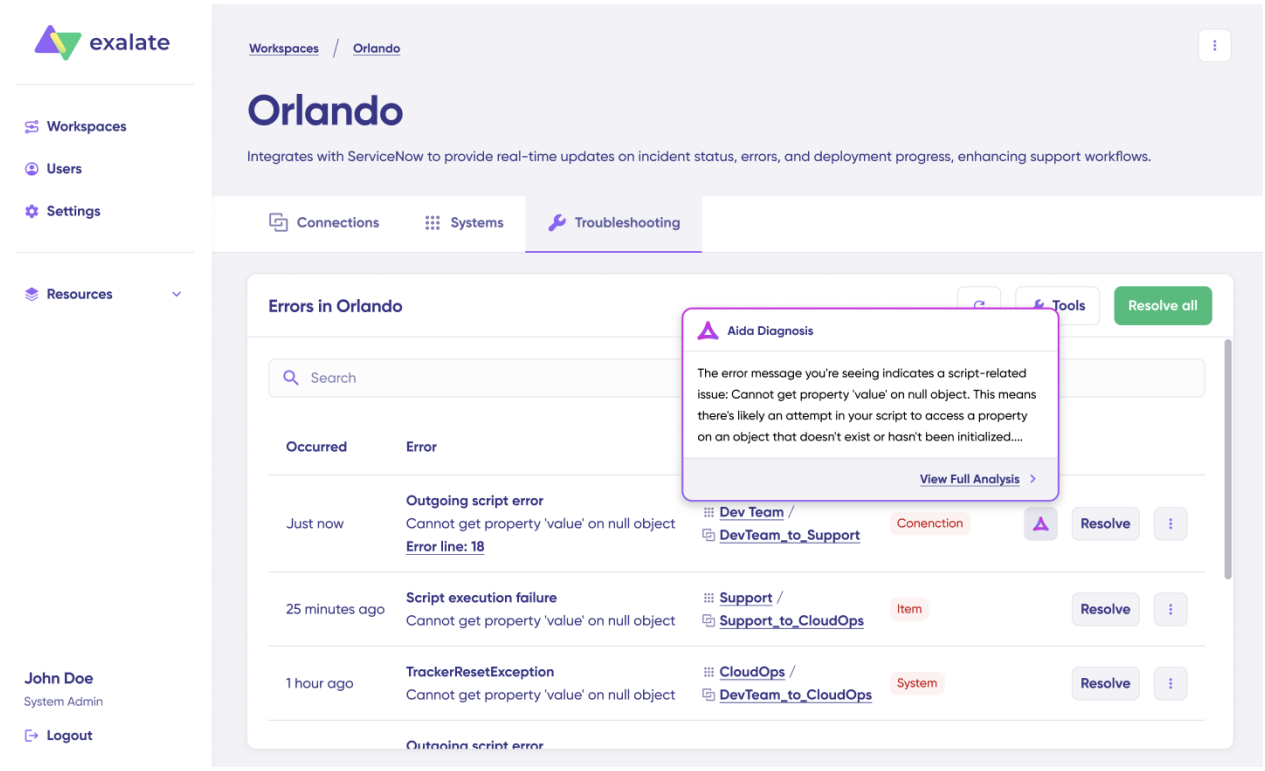
This queuing mechanism ensures **no data loss**, even during temporary problems preventing successful sync.

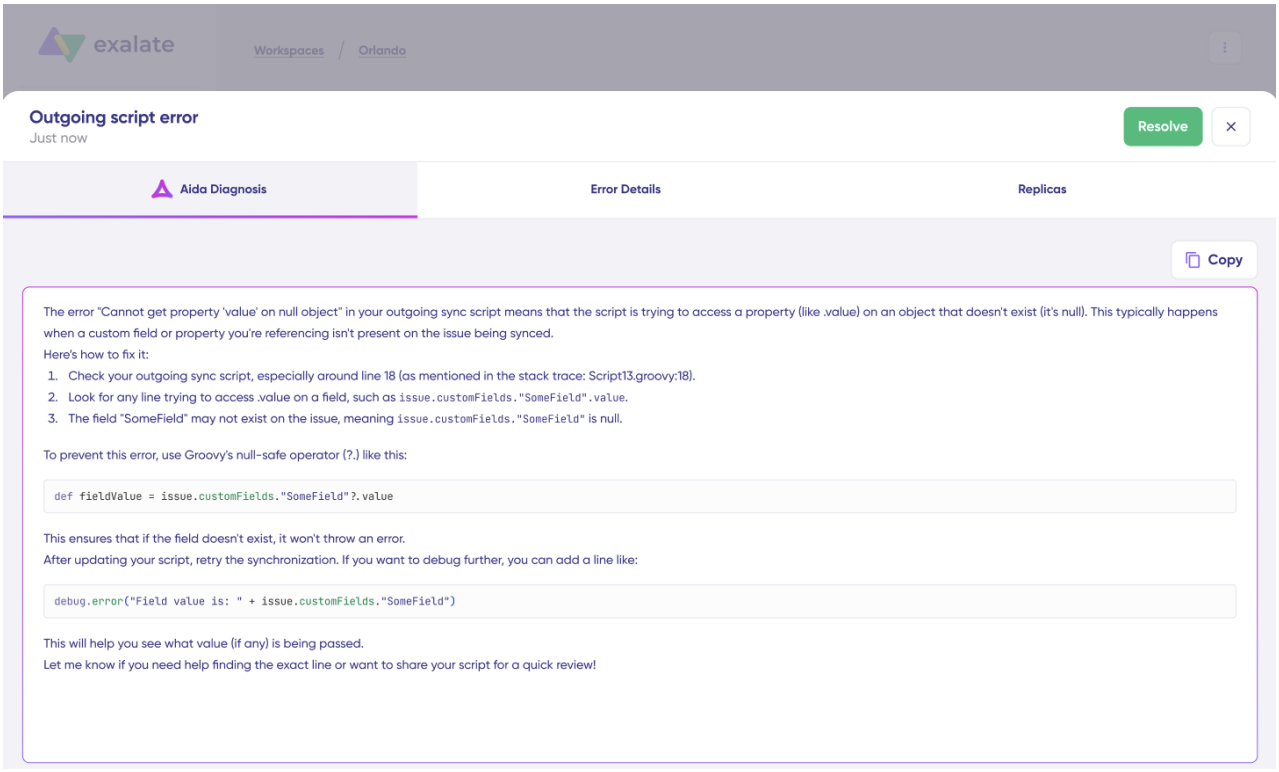
Aida's assistance for troubleshooting sync failures

In the **New Exalate experience**, **Aida** provides **context-aware assistance** for troubleshooting sync failures. When an error occurs, Aida helps by:

- Explaining the problem in an accessible language
- Suggesting **potential solutions** based on the error type and your configuration

For example, if a script tries to access a property that doesn't exist, Aida identifies the missing field and guides you on how to handle it properly.





Common causes of sync failures

Sync failures can be caused by several factors:

- **Script errors**, such as **null pointer exceptions** or **syntax mistakes**
- **Permission issues**, where Exalate lacks access to the required fields

- **Network connectivity problems**
- **Validation failures**, where incoming data doesn't meet the receiving system's requirements

The error messages provide insight into which category the failure falls under, helping you target your troubleshooting efforts effectively.

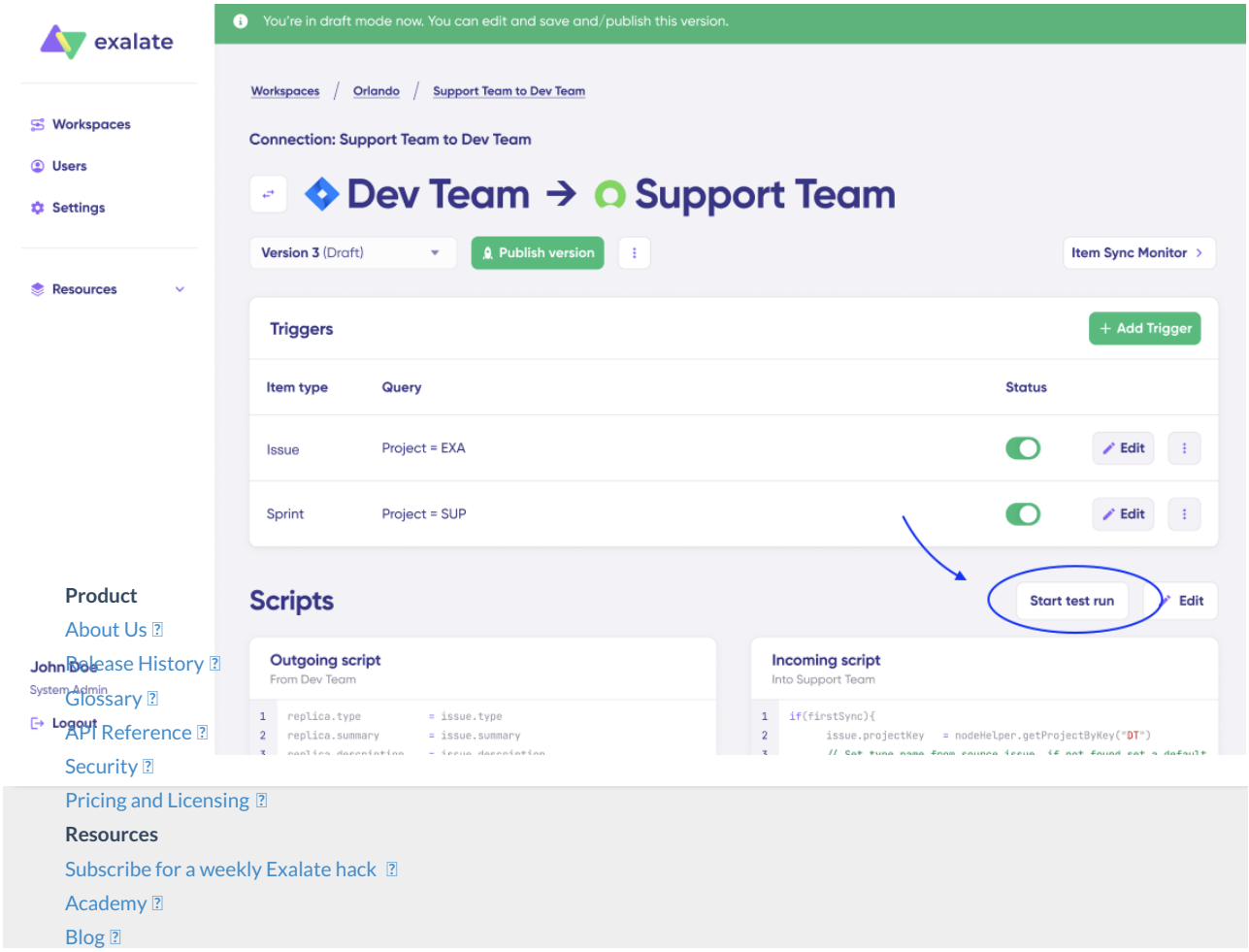
Ensuring data integrity during failures

Exalate ensures that other work items continue to sync successfully, even if specific items fail. **One problematic work item** (due to a script error or other issue) **won't block** the entire connection from functioning, which means most of your integration continues operating while you address the failure.

Testing fixes with draft versions and Test Run

For persistent errors that affect multiple items, you can create a **draft version** of your configuration and test fixes using the **Test Run** feature. This allows you to try changes in a safe environment and prevent further failures while you work on a solution.

This approach to handling sync failures ensures **continuous synchronization**, effective troubleshooting, and **data integrity** during issues. With Exalate's built-in features, you can quickly resolve issues and keep the integration running smoothly.



YouTube Channel

Ebooks

Still need help?

Join our Community

Visit our Service Desk

Find a Partner