

Rally Lookback API Getting Started Guide

[ORIGINAL DOCUMENT](#) (must have rallydev.com email to access)

[Rally Lookback API Getting Started Guide](#)

[Purpose and caveats](#)

[Documentation](#)

[How to access](#)

[Querying via GET](#)

[Querying via POST](#)

[Limitations:](#)

[Endpoints](#)

[Additional functionality as of 2012-03-11](#)

[Backward breaking changes as of 2012-03-11](#)

Purpose and caveats

The **purpose** of this early-access preview is two-fold:

1. **To provide Rally customers with a chance to start building Apps** upon this new powerful API. Several of our customers in this early-access were holding off on upgrading their existing Rally Apps because it is much easier to do certain things with this new API. We know it will take a little while for you to roll new functionality out to your Rally users and wanted to give you a head-start on that.
2. **To give us early feedback from actual usage.** This is a new data model and stack (MongoDB) for Rally and we want a chance to work out the bumps with our closest partners before rolling it out to everyone. This early-access preview is very much in the spirit of Agile's "learning by doing" approach.

Due to the nature of this being an early-access preview (some might use the term Alpha), there are a few **caveats**, you want to keep in mind:

1. **We know of a few things that don't work as expected** (see Limitations section below) **and we're still finding more.** Most of the unexpected behavior we are finding now is related to edge cases with the way Rally reports changes to the Analytics engine. We are dealing with most of them as they come up but some of them require changes to Rally that could take a while to resolve.
2. **You will see debug information (stack traces) that will eventually be hidden behind an error response.**
3. **We will take the service down from time to time** to load a new version and modify the data model. We will follow the same service level agreement that we follow for Rally's sandbox services, which states that we will not schedule any planned outage between 9am and 6pm mountain time on workdays.
4. **We are still adding functionality and we may still make backward breaking changes** if we find

features that don't hold up under load or introduce some risk.

Documentation

The hub for documentation about this effort is on GitHub at https://github.com/RallySoftware/rally_analytics

How to access

Querying via GET

```
https://rally1.rallydev.com/analytics/1.27/41529001/artifact/snapshot/query.js?find={ObjectID:1443125795}
```

where **41529001** is the ObjectID of the workspace and **1443125795** is the ObjectID of some work item (User Story, Defect, etc.) in your workspace.

All endpoints respond to both GET and POST.

Querying via POST

You can use a Chrome plugin named XHR Poster or some other "Poster" plugin for your browser to try POSTs

url:

```
https://rally1.rallydev.com/analytics/1.27/41529001/artifact/snapshot/query.js
```

content type: application/json

content:

```
{
  "find": { "ObjectID": 1443125795 },
  "fields": true
}
```

Use the 'POST' button.

Limitations:

- Data only for early-access participants. Other subscriptions in the works. We will ramp up availability based upon the feedback from early-access users.
- Double login. Your login to Rally proper is not known by the Lookback API. Your browser will prompt you to login again even if access from inside of Rally.
- When developing, you may not be able to test the example charts outside of Rally because neither cross-origin resource sharing (CORS) nor JSONP are implemented. The code for CORS is done and should be out shortly. In the mean time, you can either test inside of Rally (not ideal) or you can

launch a browser with cross-origin protection disabled. You can launch a copy of Chrome with the flag `--disable-web-security` to accomplish this. Other browsers probably have similar options.

- Boolean and numeric fields are not set in the first snapshot for newly created Artifacts. This happened because they automatically default to false and 0 in our SQL database so Rally was not sending revision history transactions about them. We should have this fixed within a couple of weeks.
- Newly created custom fields will not be query-able by string name until the next midnight after creation. We hope to integrate this look-up into the continuous ETL, but for now, newly created custom fields will not be available until midnight. Similarly, changes to allowed values will not show up until midnight.
- No backward crawling ETL. This has several impacts
 - First, we only have data back to November 19.
 - Also, snapshots don't look exactly the same. For instance, there is a `_RevisionNumber` field for the 1st snapshot but the `_Revision`, and `_User` (for the revision)
- Work item type prefixes (DE, S, US, TA, etc.) are currently hard coded when expanding queries against `FormattedID`, so if you've changed them from the defaults (other than changing US to S), it will not work as expected. You can work around this by expanding into a query against `_UnformattedID` and `_Type`.
- No easy syntax for querying iterations and releases. Right now you'll need to query Rally's standard WSAPI first to figure out the `ObjectID` of all iterations and releases that apply. Each Project has its own copy of the iteration/release data so if you need all the iteration/release data below a specific Project, you'll need the `ObjectIDs` of all of those iterations/releases. We don't like this so we're exploring ways to hide that messiness on the analytics side.
- No disambiguation of Tag Names to `ObjectIDs`. For now, you'll need to provide the `ObjectID` for any queries against the Tags collection.
- The example code for CFD and Burn charts does not yet handle well with artifacts that are deleted. We know what we have to do to fix this. We just have yet to implement the fix. The Cycle Time Scatter/Histogram example handles deletion (and even restoration) of artifacts as expected.
- SSO nor LDAP will be supported.
- Browser caching may make it hard to see the latest data without a refresh. We're looking at ways to avoid this in the future.

Endpoint

- `/analytics/1.27/1234/artifact/snapshot/query.js?find:{_Type: "Defect", "KanbanState": "Accepted"}`

Additional functionality as of 2012-03-11

- Live server data should now be available. Make a change in your Rally subscription and within a few seconds to a few minutes, the change should be represented in the Lookback API.
- You can provide the string value for all fields where that makes sense. You still need to provide the foreign key `ObjectID` for fields that relate to another Artifact (eg Parent on

HierarchicalRequirement), but now you can provide the string value for all other fields like "Accepted" for ScheduleState. Previously, you had to provide the ObjectID of the "Accepted" value (yes, every value has its own ObjectID).

- `find={_Type:"Defect", KanbanState:"Accepted"}`
- Non-custom fields support inequality comparisons (\$lt, \$gt, \$gte, etc) using the order of the values you've specified in Rally. So, "Accepted" is greater than "In Progress". Note, inequality comparisons still do not work as expected for custom fields but we're close to fixing this.

Backward breaking changes as of 2012-03-11

- \$where is now blocked. The documentation always listed \$where clauses as unsupported. Now we are actually blocking its use.
- _PreviousValues keys must be quoted (proper JSON), or the query will fail to parse.
 - `find={"_PreviousValues.KanbanState":"Accepted"}`
- Fields parameters must be proper JSON strings and quoted.
 - `find={ObjectID:12345}&fields=["KanbanState","_PreviousValues.KanbanState"]`
- _IsLeaf is no longer supported. In the documentation, we give examples of how to accomplish the same thing using other means with the same fast performance as _IsLeaf.
- JSONP is not supported at this time. There were issues with using it in conjunction with basic authentication. We are exploring possibilities for cross-browser access including [CORS](#) and additional forms of authentication.