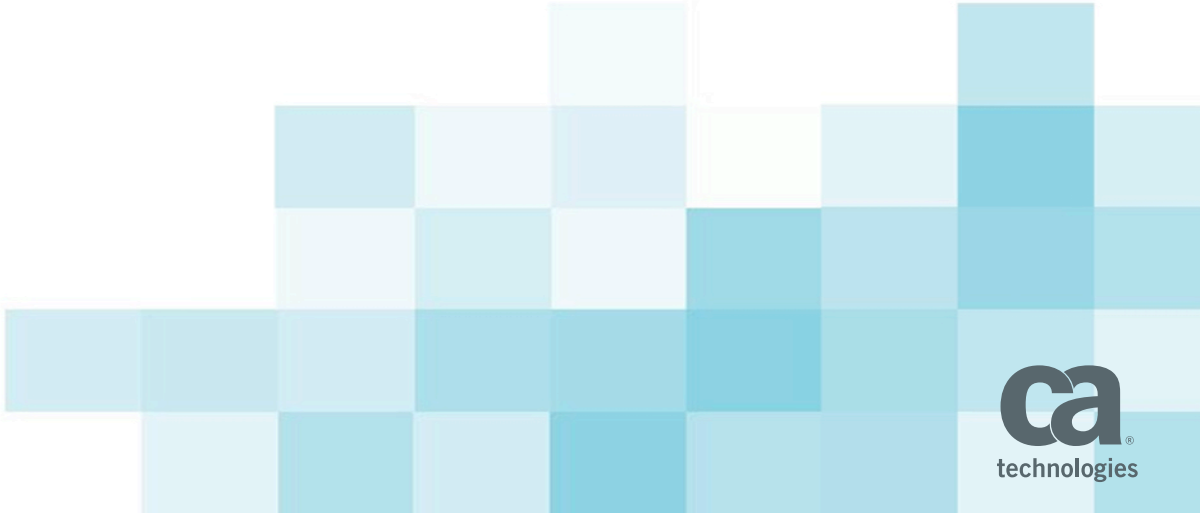
Github – Agile Central Integration Demo Hands-on

Kristy Corkan
Agile Central Technical Services

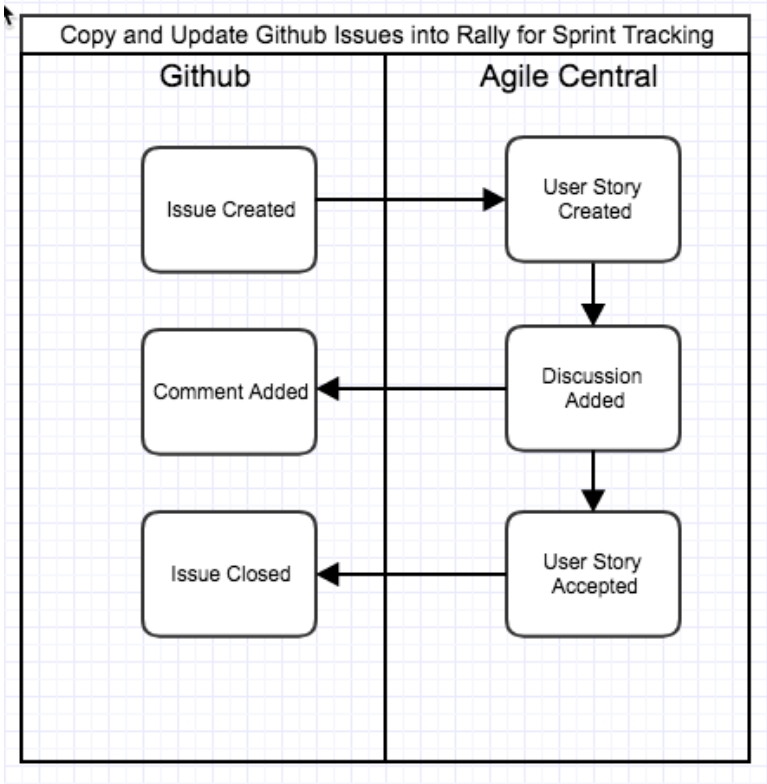
October 17, 2017



ca technologies

Github to Rally Integration

- Our team plans and tracks work in Rally and our customers create issues against our repositories.
- We would like to automatically create User Stories in Rally when an issue is submitted in Github so that we can plan the work into our iterations.
- We would like to automatically update issues as the User Story is completed in Rally so that the customer can follow the progress of the Issue in Github.



Prerequisites: System Setup

Github – Agile Central Integration Demo

1. Select a Rally Project to demo with (must be one that you can create test data in and have editor privileges to). Get the project reference from the project's details page (e.g **/project/12345**)

2. Create an API Key in Rally with permissions to edit in the selected Rally Test Project (goto https://rally1.rallydev.com/login to create an API key) for authentication

3. Select a Github Repo to demo with that they have access to

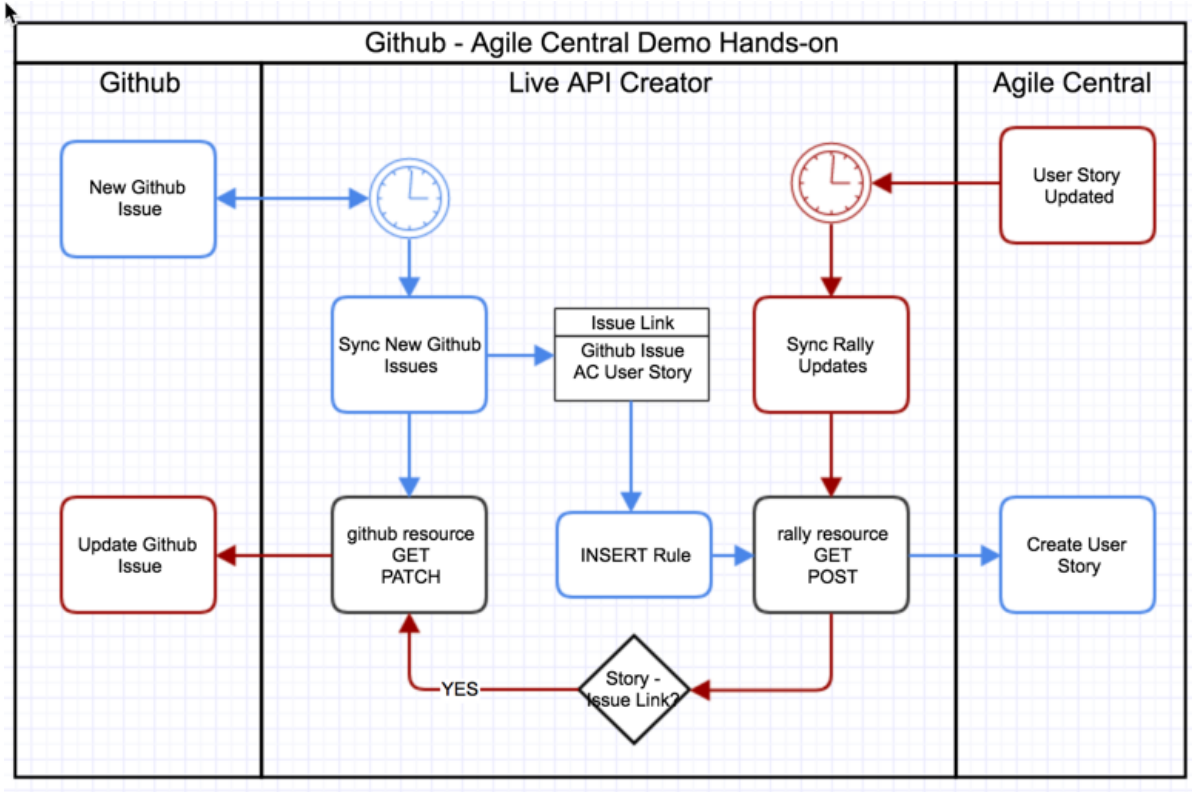
4. Create your encoded ***username:password*** string to use for Basic authentication:
   ```
   > echo -n username:secretpassword | base64
   ```

5. Note the key parts of the base address of your repo: https://github.com/***organizationOrUser/repositoryName***

Approaches

- Webhook triggers are ideal for this scenario, since they are supported by both Github and Agile Central, *though they may not be feasible for some customers running LAC on premise behind a firewall*.
- We will use the Timer feature to monitor changes in each system.

LAC Components and Flow “Stickies”



Build Steps

- ✓ Enable moment.js in Libraries
- ✓ Create Link Table with fields
  - ✓ ac_id (string),
  - ✓ github_id (string),
  - ✓ github_data (text)
- ✓ Create & Test General Resources
  - ✓ Github GET/PATCH (Reference: https://developer.github.com/v3/issues/)
  - ✓ Rally GET/POST (Reference: https://rally1.rallydev.com/slm/doc/webservice/)

Build Steps (cont)

- ✓ Create linkedObject Table resource
- ✓ Create Request Event for linked Object
- ✓ Create Github Sync “Controller”
- ✓ Create INSERT Rule
- ✓ Create Rally Sync ”Controller”
- ✓ Create the timers

Challenges to consider

- Logging and troubleshooting: how will a customer know/troubleshoot when a sync throws an exception?
- Paging of data
- Configuration: how to make the the demo/implementation easily configurable to change systems, update tokens and extend?
- More complex value syncing, deletions
- Each system in an integration flow must support all synced object types and relationship between those types.


ca
technologies