RELLEX RELEASE NOTES

VERSION: 1.4 (SUMMER 2019)

RELEASE DATE: 28/09/2019

What's new?

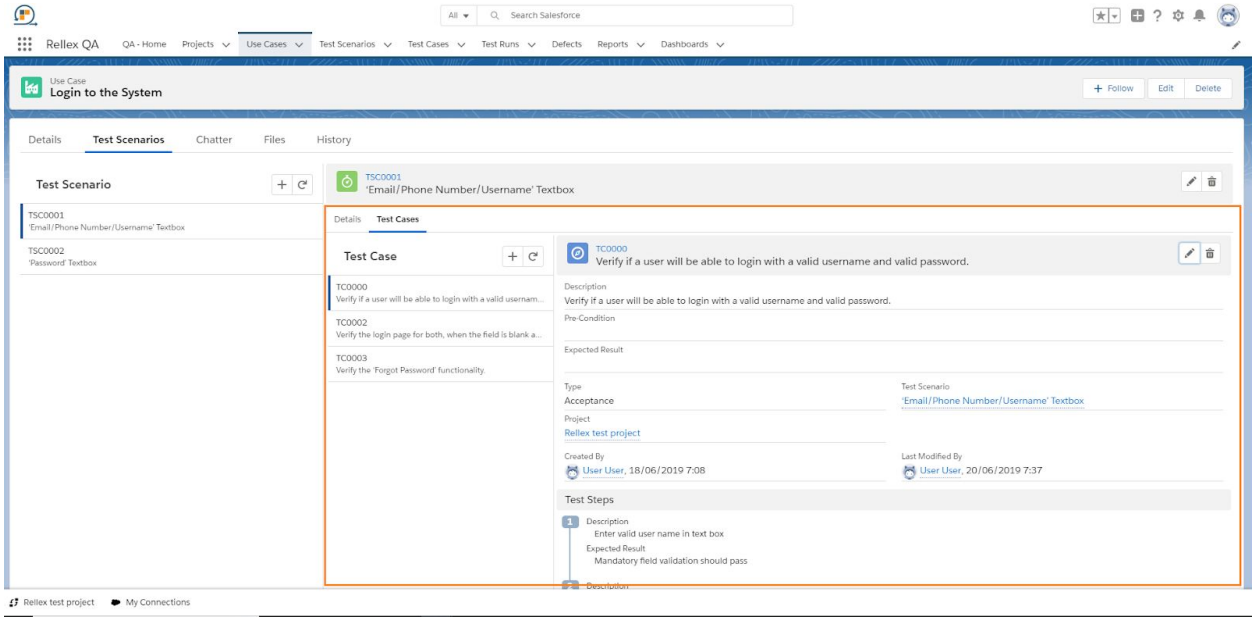
With this Summer '19 release we are bringing a whole new module to Rellex. We are happy to announce that Rellex will now contain a quality assurance module next to the existing project & release management modules! With Rellex QA you will be able to document your use cases, execute test runs and track defects!

But that's not all! We are adding many more features!

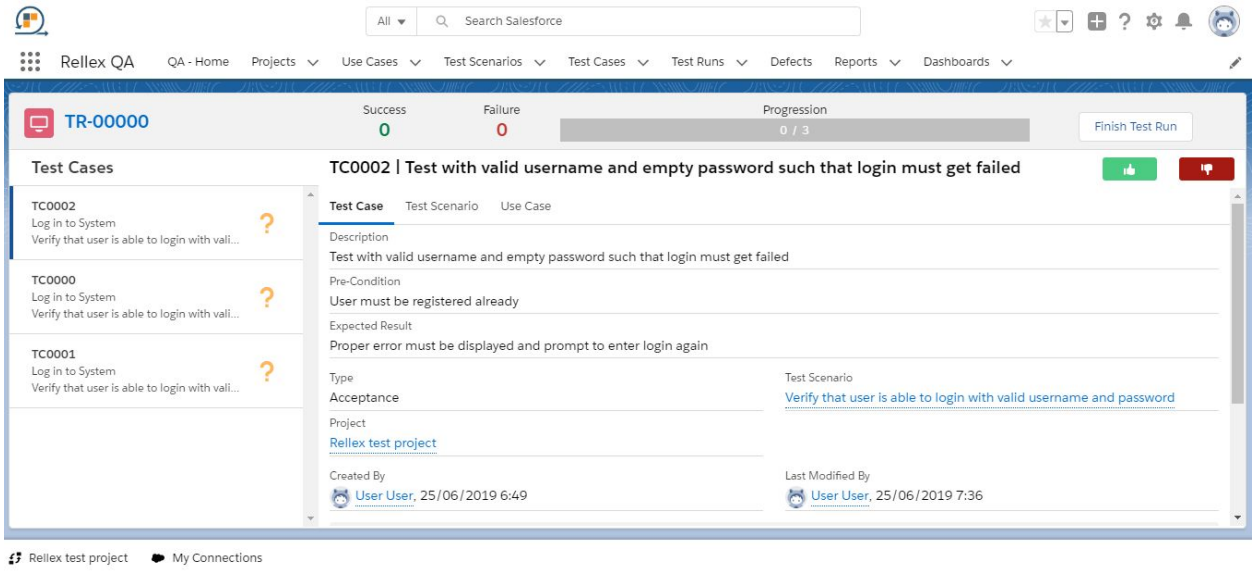
Quality Assurance

Today Rellex contains a project management module in which you can plan your releases and sprints, maintain your backlog and track the status of your user stories. It also contains a release management module that allows you to track and deploy metadata from one Salesforce environment to another. Now within this new release we are adding a quality assurance module on top of this.

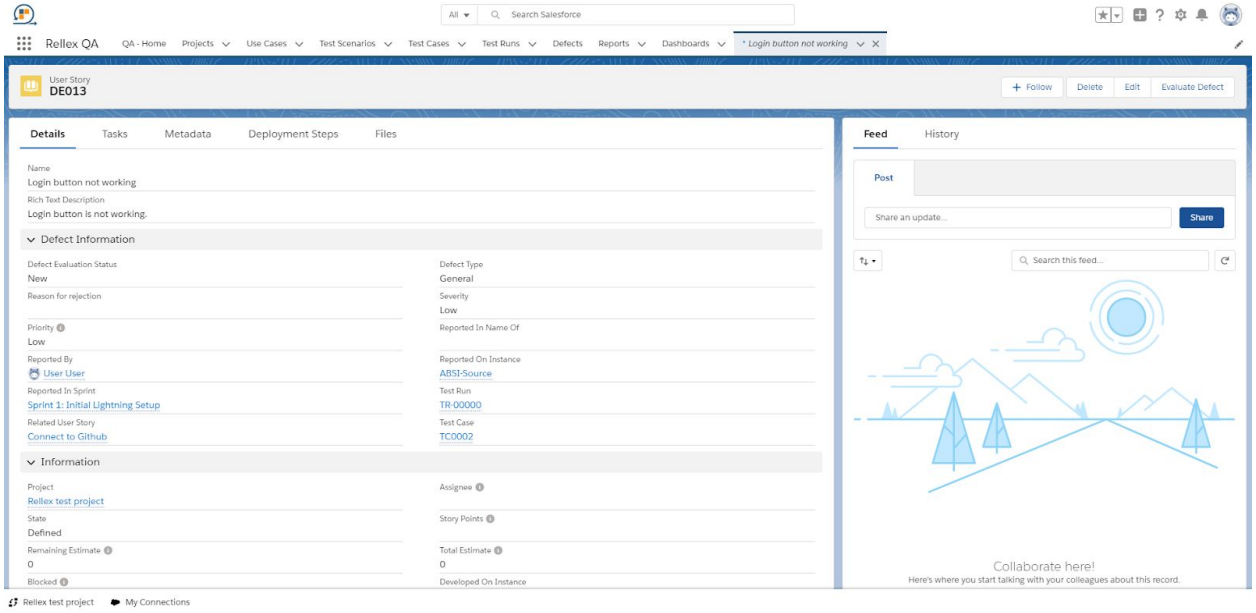
This new module consists of three major parts. The first part is the documentation. This allows you to document a use case, test scenarios and test cases, allowing you to track what has to be tested during each iteration.



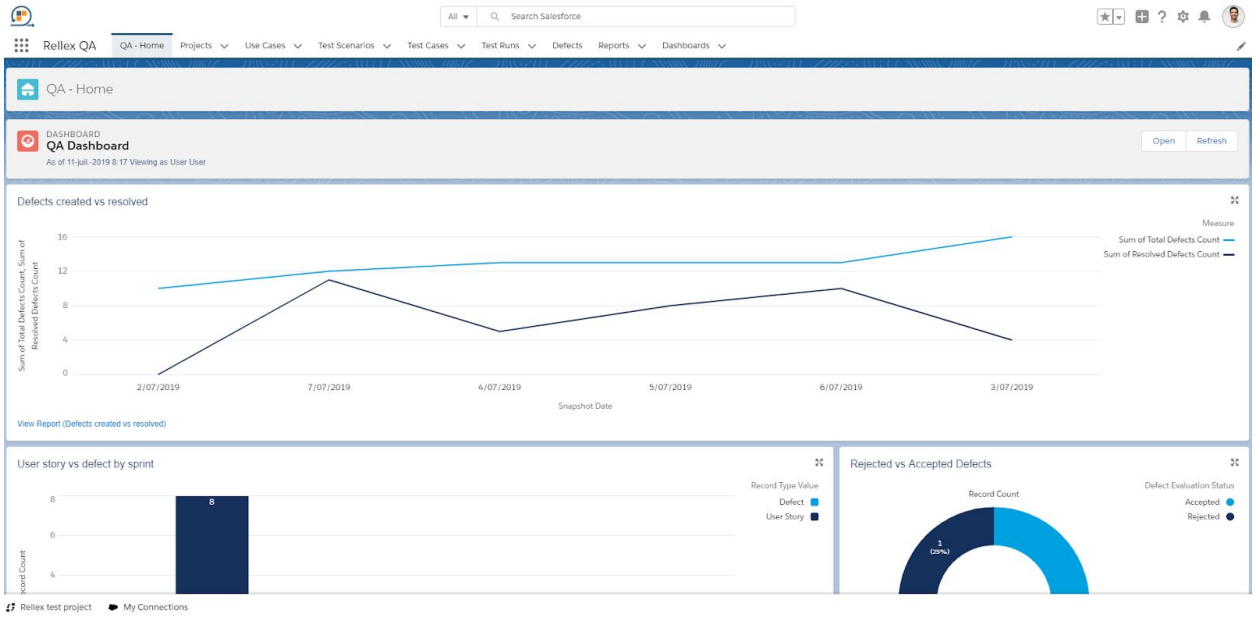
The second part is the execution. The execution part allows you to define a test run by selecting the use cases or test cases you wish to test. Once your test run is defined you can track the progress and capture the result of each test that is being executed.



The last part is logging the defects that are found during (or outside) a test execution. These defects flow back to the implementation team who will pick them up and handle them accordingly.

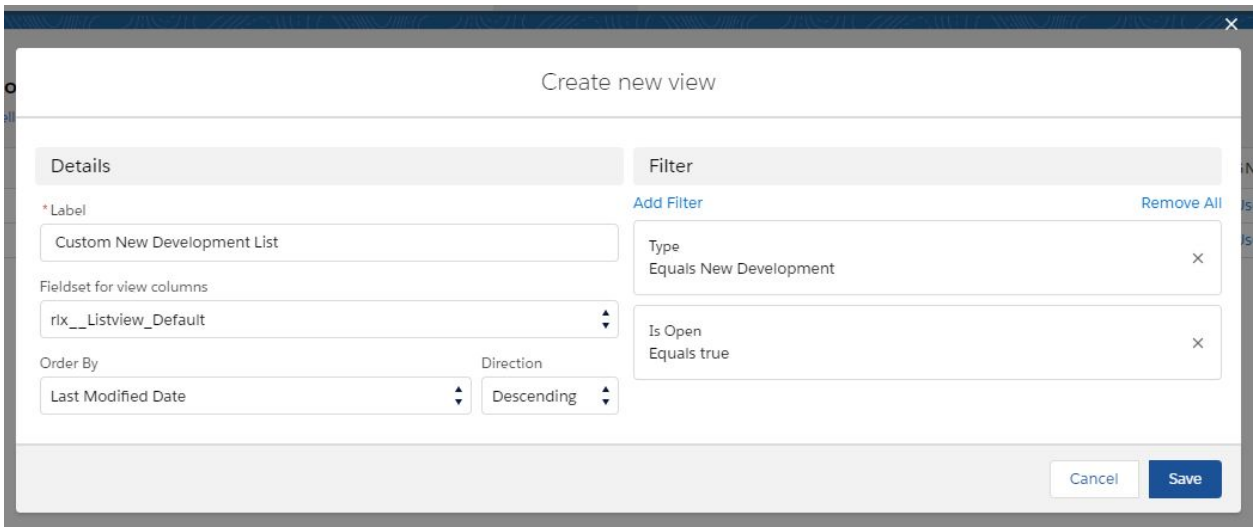


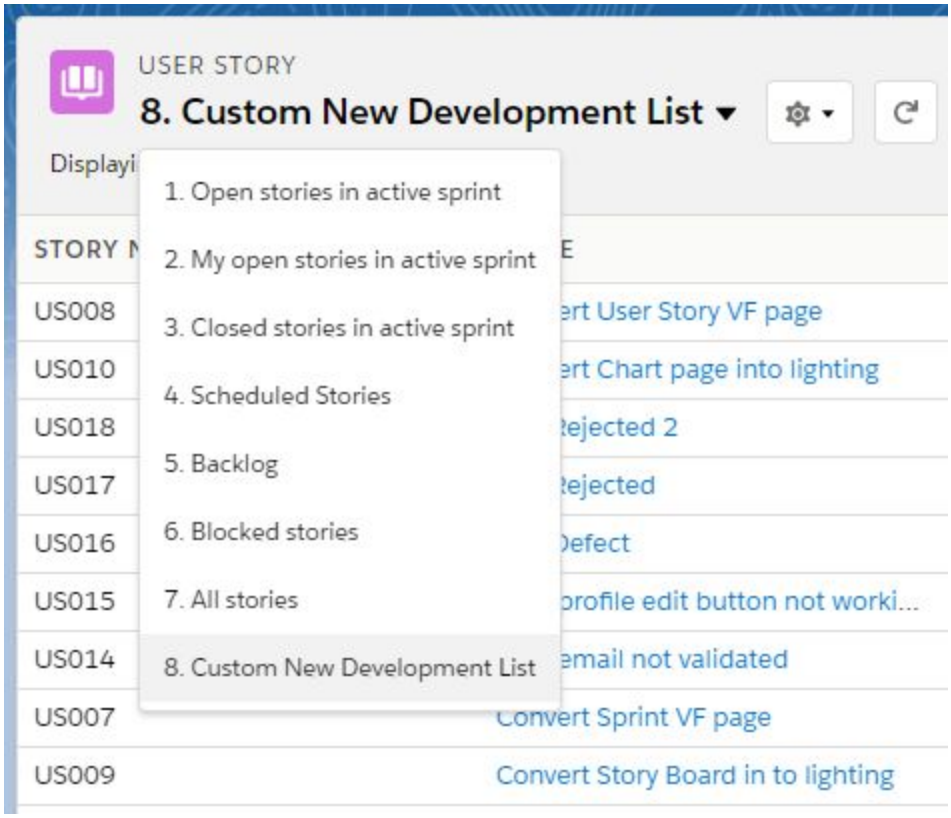
All this is topped off with out of the box reports and dashboards allowing you to track your team's performance in a visual way.



View - Customizable filters

It is now possible to create custom views on your data in Rellex. Next to the base views Rellex provides, you can now define your own filters to see the data you want to see. These filters are available for the object listviews as well as all the related lists.



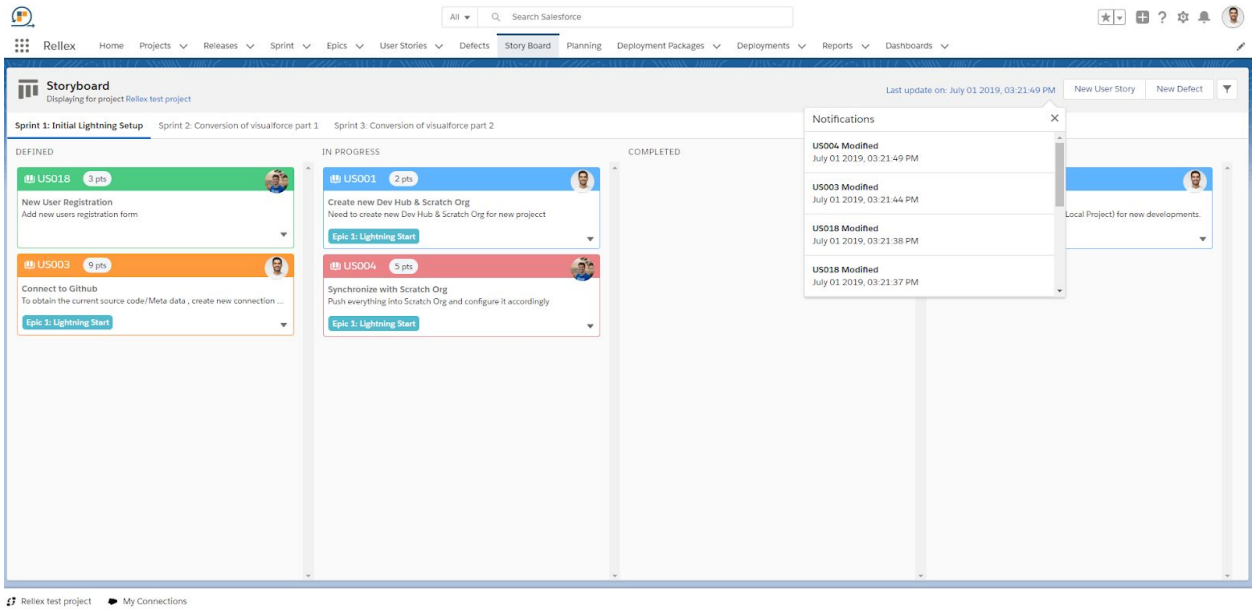


View - Last used

From now on Rellex remembers the last view that you were using. The next time you return to the same list, Rellex will select your last used filter by default. This applies for both listviews as well as related lists.

Boards - Real time updates

The Rellex boards are a great way to follow up on your sprint progress on a regular basis. It's the place where the implementation team will spend most of their time. In the past there was a risk that you were not looking at the latest version of the board if you stayed too long. To counter that and to eliminate the need of refreshing on a regular basis, we have added real time updates. This means that when another user updates the board status, this will automatically reflect for any other users looking at the same board. This ensures that you are never working on an older version of your board.



Please check the Rellex documentation on how to activate this new feature.

Boards - Project vs release backlog

The planning board is a great way to manage your backlog and to plan your sprints. It was however not always clear what was shown in the Unscheduled section. This section shows both the project backlog as well as the release backlog of the one you have selected. This is confusing if you purely want to schedule your current release.

To improve this, an additional filter option was added that allows you to only show the release backlog, only the project backlog or both. By default both are still shown.

Standard tabs & global search support

In the previous version of Rellex all the tabs that showed the listviews of their respective object were custom tabs. A big downside to this was that you were not able to search for the respective object in the Salesforce global search bar.

With this new release we have migrated the custom tabs to a standard tab. Doing this you are now able to make use of the tab functionalities in Salesforce like see recently viewed on the fly, directly create a new record and more importantly, you are now able to search! Please be aware that the standard search is not limited to your currently active project.

Tracking a deployment step status

Deployment steps are a great way to track manual steps that need to happen before or after a deployment to a specific environment. These deployment steps are shown directly in Rellex as a reminder. Up until now this was just a read only list of steps to take.

Now we have added the capability to capture if a deployment step was completed or not. This status is directly linked to the salesforce instance to which you are deploying. This ensures that you don't do a

deployment step twice to the same instance. It is also a great way of tracking your progress when it's a big list of steps you have to complete.

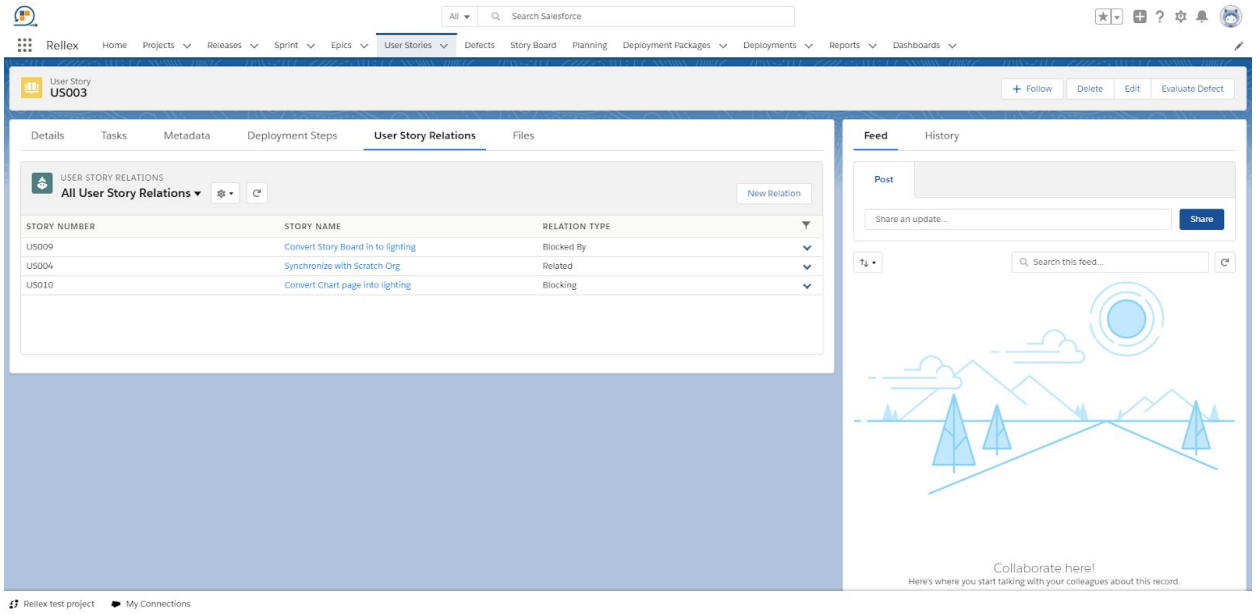
User story - Relationships

Rellex allows you to capture all your user stories, plan them and track their progress. But what if you have dependencies between your user stories? What if you can't start working on one story when the other is not yet ready? So far there was no way of capturing that information in Rellex. With the latest release we are happy to announce that from now on you are able to capture relationships between your user stories.

From now on you can create a link between user stories. The relationship can have three types:

- Blocking
- Blocked By
- Related

Creating a blocking/blocked by relationship will have additional effect on the affected user stories. User stories that are being blocked by at least one other user story will have a lock on them. This lock works in the same fashion as setting a manual block on the user story, no status changes are allowed at that point.



For more information please check the Rellex documentation.

User story - Proposed solution field

A new field was added to the User Story object named Proposed solution. This field can be used to describe the technical solution that has to be implemented to complete the user story.

User story - Evaluation

With the new "User Story evaluation" feature we allow a user to evaluate if a user story is ready for implementation or not. So far when a user story was created there was no way to identify if a user story was mature enough to start the implementation.

With this new feature we have added a new field to track if a user story is ready or not. When the feature is activated all user stories that are not ready for implementation can't move to an in progress status.

Metadata storage 2.0 - migration period

With this new release we are also marking the first step towards the future of metadata handling in Rellex. In order to support more advanced features we needed something more than was already present.

Therefore we took a radical change by implementing a new data model to keep track of metadata changes on a user story. To ease the transition between the new model and the existing model we are using the time between this release and the next release as a migration period.

The idea here is that by the next release all your old metadata information has been migrated to the new structure. In order to ease this process for you, the following is foreseen:

- A batch that automatically converts all your old data to the new data model.
- Trigger logic to copy the old metadata record to the new structure, this ensures that all newly created/updated metadata is also stored within the new data model.

What is expected of you?

Between this and the next release we expect that you have run the provided batch. The batch will move all the old 'stale' data, while the trigger ensures that all new / deleted data is also taken into account.

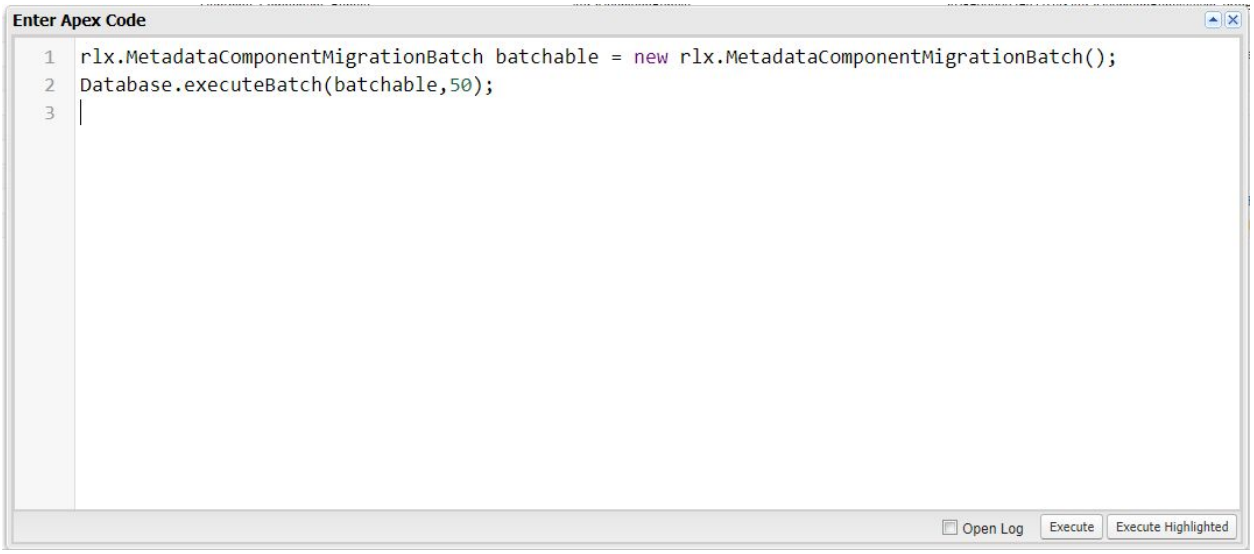
As of the next release all existing and new features will start using the new data model. If you did not run the batch before that time you might miss data linked to your user stories as Rellex no longer looks at the old model. Don't worry, your data will never be deleted so simply running the batch will always ensure the correct data is copied over, even after the switch.

Execute Apex Batch 'MetadataComponentMigrationBatch'

To migrate existing metadata into the new structure, the first option is run an apex batch. This batch will migrate all the existing metadata into the new metadata storage 2.0 format.

Open the Salesforce developer console and run following code as anonymous. (Make sure to keep batch size as 50)

```
rlx.MetadataComponentMigrationBatch batchable = new
rlx.MetadataComponentMigrationBatch();
Database.executeBatch(batchable,50);
```



Once you press execute the batch will start and you can use setup>environment>Jobs>Apex Jobs to follow up on the batch progress.

Bugs we crushed

- Fixed issue with inline edit on package user stories
- Fixed issue where updating a sprint in JIRA would result in an incorrect sprint status causing the sprint to disappear in Rellex.
- Fixed issue where related user stories were not displayed on the deployment error overview.
- Fixed issue where the name of a jira user story was too long and therefore the user story could not be created.

Manual Actions Required

With the release we are bringing new features. These new features come with new package components or upgrades to previous components. Sadly not every update can happen automatically. This is where your Rellex admin will have to make some small adjustments to get all the new features up and running.

Record type access

With this new release we are introducing record types on the user story object. This differentiates between user stories and defects.

Please make sure to assign the record types on your user profiles. Sadly we are unable to push this access to your org through the package upgrade. If the record access is not added to the profile user's might not be able to create user stories or defects in certain areas. We advice to set the default to User Story.

Layout assignment

This new release adds new record types on user story. This also introduces two new layouts, one for user story and one for defects. Make sure to assign the layouts on your respective user profiles to ensure the correct layout.

Fields

What should I do?

- Add these fields to your layouts. Package upgrades have no control over layouts.
- If you're using profiles or custom permission sets then make sure to add field level security for these new fields.

User Story

- Proposed Solution
- Ready for implementation

What should I do?

- Add the fields to the User story layout.
- Include all fields in custom profiles / permission sets