


# How to : fix issues in automation

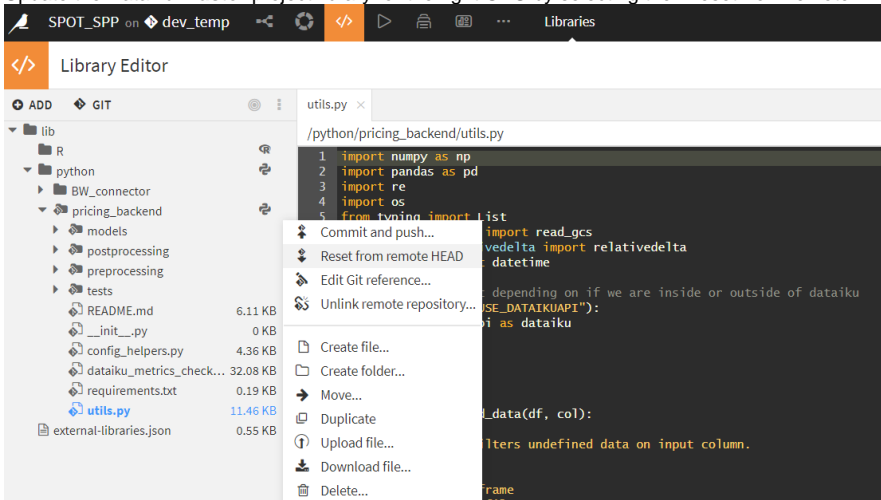
 Note: for some inputs on the common issues, please refer to [this article](#).

The code of the project is split between a common code-base and specific code in the Dataiku recipes, as highlighted in the [project architecture](#).

The actions to take are different depending where the issue is located.

## Option 1: Issue related to the code base (no change on the Dataiku project itself)

- Create a hotfix branch from the [code base repository master](#) branch.
- If you need it to test your development, create a branch from the Dataiku project in design following [these guidelines](#) and switch the library to your new code base hotfix branch.
- Make sure the hotfix does not break anything related to other GBU projects (you can create a branch based on your new code base in the [CS](#) or [SPP](#) project and run a "Full reload" scenario to make sure it is still working).
- Make sure the Gitlab pipeline with the unit tests passes successfully.
- Merge your hotfix branch for the code-base on **master** (you can still refer to [this page](#)).
- Update the Dataiku **master** project library for the right GBU by selecting the "Reset from remote HEAD" option.



- Launch a "Full reload" scenario on dev to make sure it runs properly
- Merge your hotfix branch for the code-base on **dev** (you can still refer to [this page](#)) to make sure it has the same status as master.
- Create a new bundle of the project and [deploy](#) it in automation.

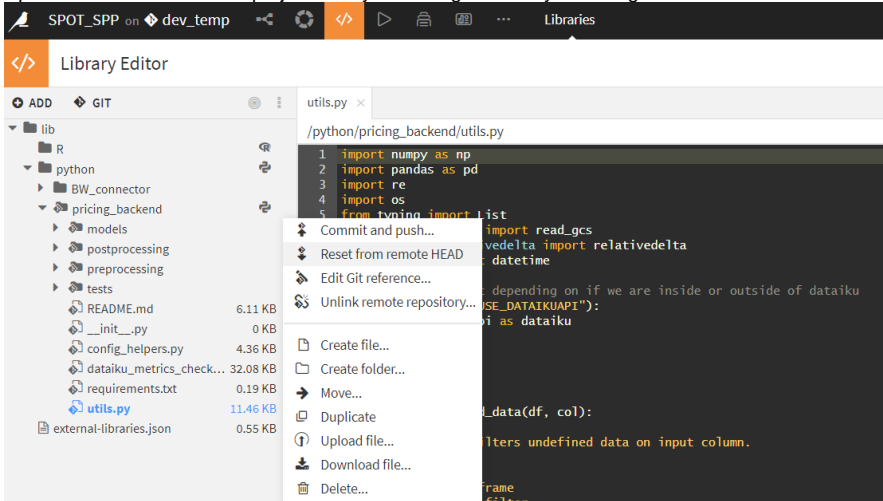
## Option 2: Issue related to the Dataiku project Flow (no change on the code base)

- Create a hotfix branch by duplicating the **SPOT\_SPP\_MASTER** Dataiku project (refer to [How to : start a new development](#) for the step but be careful that you are not starting from dev but master)
- Apply the hotfix and merge back to **master**, run a "Full reload" scenario to make sure everything works (**do not** rebase in master !)
- Rebase the dev to apply the hotfix changes.

## Option 3: changes on both the code base and the Dataiku project

- Create a hotfix branch by duplicating the **SPOT\_SPP\_MASTER** Dataiku project (refer to [How to : start a new development](#) for the step but be careful that you are not starting from dev but master)
- Create a hotfix branch from the [code base repository master](#) branch.
- Apply the changes to fix the issue
- Make sure the hotfix does not break anything related to other GBU projects (you can create a branch based on your new code base in the [CS](#) or [SPP](#) project and run a "Full reload" scenario to make sure it is still working).
- Make sure the Gitlab pipeline with the unit tests passes successfully.
- Merge your hotfix branch for the code-base on **master** (you can still refer to [this page](#)).
- Merge back the Dataiku branch to **master**, run a "Full reload" scenario to make sure everything works (**do not** rebase in master !)

- Update the Dataiku **master** project library for the right GBU by selecting the "Reset from remote HEAD" option.



- Launch a "Full reload" scenario on dev to make sure it runs properly.
- Merge your hotfix branch for the code-base on **dev** (you can still refer to [this page](#)) to make sure it has the same status as master.
- Rebase the hotfix Dataiku branch on **dev** to apply the hotfix changes.
- Create a new bundle of the project and **deploy** it in automation.