

Validacion de PRs



Cuba, Cinthia (LAS-V)

Tags



Status: Aprovado

1. Una vez que la persona encargada de evaluar los PRs de cada torre finalice, debe agregar la etiqueta "Estado: Aprobado" para que el equipo de Iris pueda evaluar ese PR.



Status: Alteração Pendente

2. Cuando se evalúe el PR y se requieran ajustes, se debe eliminar la etiqueta "Status: Aprobado" y se debe agregar la etiqueta "Status: Alteración Pendiente".



Status: Fazer Nova Revisão

3. Una vez finalizados los ajustes solicitados con "Status: Alteración Pendiente", se debe quitar esta etiqueta y añadir la de "Status: Hacer Nueva Revisión".



Assunto: Atualização

4. Cuando tengan la etiqueta de "Asunto: Actualización", no pueden haber agregaciones ni modificaciones de datasets.



Assunto: Bug

5. Cuando tengan la etiqueta de "Asunto: Bug", no pueden haber agregaciones ni modificaciones de datasets.

★ Asunto: Novo Dataset

6. cuando tengan la etiqueta "Asunto: Nuevo Dataset" deben verificar que la nomenclatura de todos los artefactos esté correcta ([mas detalles aquí](#)) y el datalakepath que figura en la documentación del YAML sea el mismo path que está siendo escrito en CZ.

Ejemplo:

```
datalakepath: Brazil/Sales/InnoTrade/BeesChefia/UserCreated
```

Triggers:

Nomenclatura de triggers:

A_[Contexto][Nombre del Trigger].json
[Contexto][Nombre del Trigger].json

Los triggers que comiencen con A_ solo se activaran en el entorno de producción.

Posibles estados del trigger:

- A_runtimeState = Stopped: Trigger activado sólo en producción.
- A_runtimeState = Started: Trigger activado en prod y dev (Atencion!!!, Tener cuidado y solicitar que se cambie el estado a Stopped si no es necesario en dev para que el trigger no se active en ambos entornos, gastando recursos innecesarios).

- Trigger sin A_ con runtimeState = Started: Trigger activado solo en entorno Dev.
- Trigger sin A_ con runtimeState = Stopped: El trigger no se activará ni en Dev ni en Prod.

Ejemplo de Triggers que se activan en producción:

Documentación .yaml

Toda notebook que escriba en CZ debe tener documentación, aún si se trata de una actualización en la notebook, es necesario dejar el link de la documentación en el PR.

- La nomenclatura debe reflejar el nombre del dataset.
- El campo "name" debe ser igual al ultimo path del datalakepath.
- El campo "datalakepath" NO debe comenzar con "/". Debe ser igual al valor de "country".
- Los campos "description", "teamowner", "dataowner" e "dataexpert" debe contener doble comillas cuando tenga algun caracter especial (incluyendo espacios).
 - **Caracteres especiales:** [@ ! # \$ % ^ & * () < > ? \ | } { ~ :]
- El nombre del archivo yaml debe estar escrito en snake_case.
- Chequear que la línea de "partitionbycolumn" no puede estar vacío o tener n/a. Si no tiene partición, no debe declararse en el yaml.

- Es necesario también que la identificación sea correcta, y que la extensión del archivo sea .yaml

Aclaración: Como aún no están definidos los dominios y subdominios de "Las" utilizaremos la siguiente nomenclatura:

Nomenclatura para nombres de archivos .yaml:

commercial_<sistema>_<pais_abreviado>_<nombre_dataset>.yaml

quedando:

**commercial_chess_py_resufactkpiipopchopp.yaml o,
commercial_chess_las_resufactkpiipopchopp.yaml, según el caso.**

DataFactory

Pipeline (PPL)

Los PPL deben estar dentro de la carpeta *datafactory/integracion/pipeline* + *nomenclatura clara* + *_PPL.json*

En caso de "timeout": "7.00:00:00", el retry debe estar con valor 0.

Configurar el "timeout" adecuadamente según el caso.

Databricks

NOTEBOOKS

Nomenclatura de notebooks:

Origen_Destino_Sistema_Dataset

- Evitar el uso de la librería Pandas, ya que la misma puede sobrecargar el driver del Cluster.
- Utilizar PyLiris para lectura y escritura de datasets en CZ, DW y Presto.
- Las notebooks que escriben en la CZ debe contener documentación .yaml y el path de escritura debe ser el mismo que el datalakepath de la documentación.

Ejemplo de la escritura en la notebook