Templates & Macros

Темплэйты в Apache Airflow представляют собой механизм для параметризации DAG и задач внутри них с использованием Jinja2, мощного языка шаблонизации Python. Они позволяют встраивать переменные и выражения в атрибуты задач, что делает их более гибкими и настраиваемыми.

```
task = BashOperator(
    task_id='task2',
    bash_command='echo Processing file {{ ds }}.csv',
    dag=dag
)
```

Макросы представляют собой набор предопределенных функций в Airflow, которые облегчают доступ к метаданным и другим полезным данным в процессе выполнения задач. Они предоставляют удобный способ получения информации о времени выполнения, конфигурации и других аспектах DAG.

Как использовать?

Передача параметров в задачи
Динамическая генерация путей или имен
Управление параметрами и конфигурациями

```
task = PythonOperator(
    task_id='task3',
    python_callable=my_python_function,
    op_kwargs={'table_name': 'data_{{ ds_nodash }}'},
    dag=dag
)
```

```
# Динамическое использование даты выполнения в SQL-запросе
task = MySqlOperator(
    task_id='task2',
    sql=f'SELECT * FROM my_table WHERE date = "{{ ds }}"',
    mysql_conn_id='mysql_default',
    dag=dag
)
```

BD SCHOOL
ШКОЛА БОЛЬШИХ ДАННЫХ

Templates

- {{ jinja templating language }}
- Templates reference
  https://airflow.apache.org/docs/apache-airflow/stable/templates-ref.html#
- Удобное и практичное применение, часто best practice

templates

```
{{ var.value.foo }}
{{ ds }}
{{ var.json.foo.get() }}
```

without_templates

```
from airflow.models.variable import Variable
нужно написать отдельный метод, который
возвращает приходящие кварги
```

Templates

Ограничение использования templates в PythonOperator

```python
def my_function(**kwargs):
    ds = kwargs['ds']
    print(ds)
```

```python
def my_func(ds):
    print(f'Execution date: {ds}')

python_task = PythonOperator(
    task_id='my_task',
    python_callable=my_func,
    op_kwargs={'ds': '{{ ds }}'},
)
```

```python
echo_ds = BashOperator(
    task_id="echo ds",
    bash_command="echo {{ ds }}"
)
```

```python
def print_ds():
    print("print {{ ds }}")
```

Render templates

```
create_file_ds = BashOperator(
    task_id='create_file_ds',
    bash_command='touch /usr/tmp/lect4_{{ ds }}; touch /usr/tmp/lect4_{{ ds_nodash }}; ls /usr/tmp |grep lect4_*'
)
```

Task Instance: create_file_ds at 2023-09-22, 00:00:00

Task Instance Details | Rendered Template | Log | XCom

Rendered Template

bash_command

```
1  touch /usr/tmp/lect4_2023-09-22; touch /usr/tmp/lect4_20230922; ls /usr/tmp |grep lect4_*
```

Macros

Макросы в Airflow — это функции или переменные, которые могут быть использованы внутри шаблонов (templates) для динамического вычисления значений.

default macros

Variable	Description
`macros.datetime`	The standard lib's `datetime.datetime`
`macros.timedelta`	The standard lib's `datetime.timedelta`
`macros.dateutil`	A reference to the `dateutil` package
`macros.time`	The standard lib's `time`
`macros.uuid`	The standard lib's `uuid`
`macros.random`	The standard lib's `random.random`

some specific macros

airflow.macros.ds_add

airflow.macros.hive.max_partition

airflow.macros.ds_format

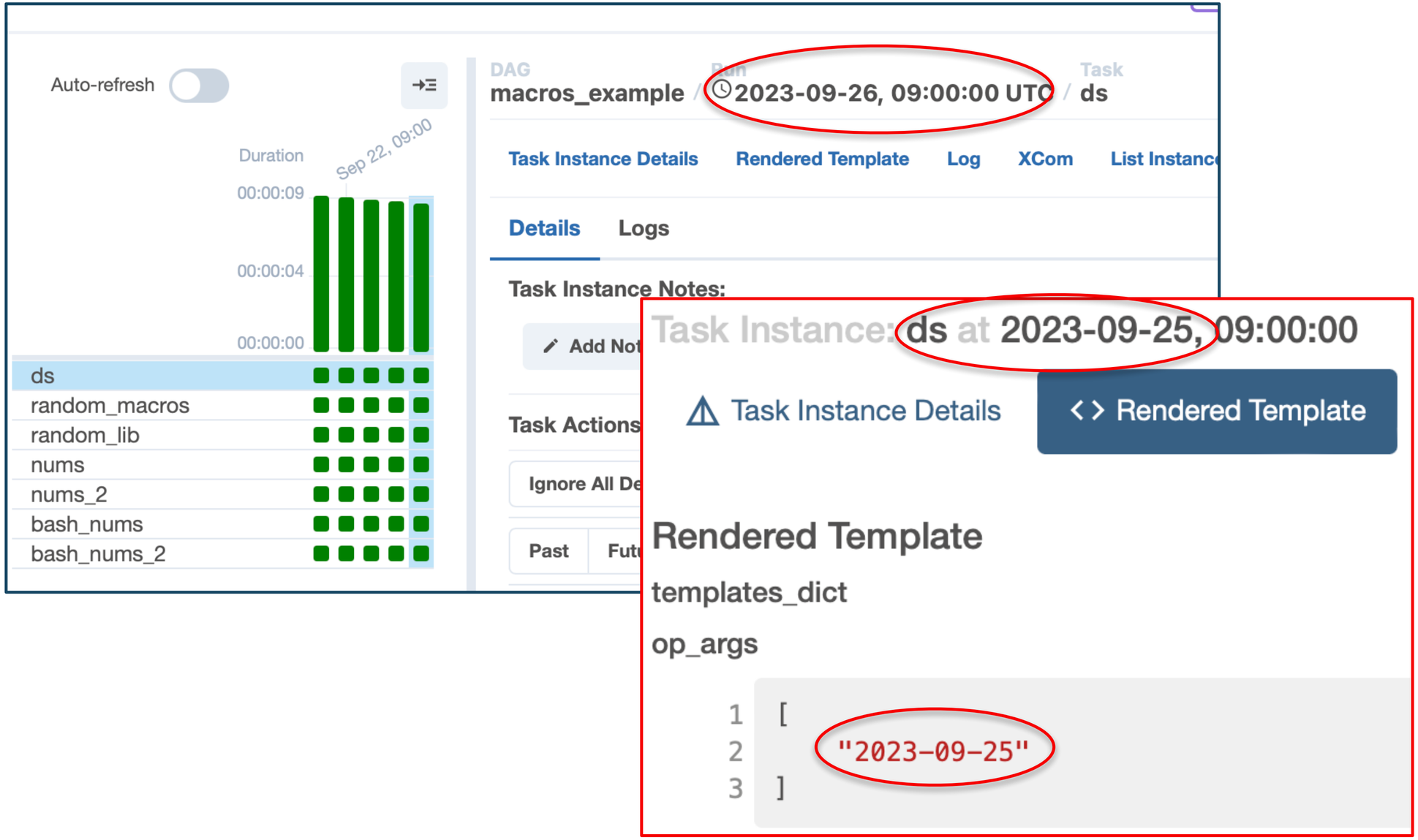
Macros

{{ }} - templates

{{ все, что внутри - macros **}}**

```
insert_data = PostgresOperator(
        task_id="insert_data",
        sql="""
        INSERT INTO example_table (id, value)
        VALUES ('{{ macros.uuid() }}', {{ macros.random() }})
        """
    )
```

Macros



Macros(link)

Разбор примера **macros_example**

Hooks

Хуки (Hooks) в Apache Airflow — это абстракции, которые обеспечивают удобный интерфейс для подключения и взаимодействия с внешними системами и сервисами, такими как базы данных, хранилища данных, API и другие системы. Хуки предоставляют набор методов для выполнения операций чтения, записи и других взаимодействий с этими системами.

```python
def psql_hook():
    postgres_hook = PostgresHook(postgres_conn_id='postgres_default')
    records = postgres_hook.get_records("SELECT name FROM lect_2;")

    for row in records:
        uppercase_name = row[0].upper()
        print(uppercase_name)
```

```python
psql_hook_op = PythonOperator(
    task_id="psql_hook",
    python_callable=psql_hook
)
```

```
Постгрес хук (Postgres Hook) – это как помощник шеф-повара,
который знает, как получить ингредиенты из кладовой.
Он знает, где находятся нужные продукты,
как открыть кладовку и как правильно взять нужные ингредиенты.

Пример:
Помощник идет в кладовую (базу данных PostgreSQL),
открывает нужную полку (подключается к базе данных)
и достает необходимые специи (данные).
```

Hooks

(link)Разбор примера **postgres_hook_example**

(link)Разбор примера **s3_create_file_and_connection_dag**

Postgres, macros, templates [«07_practice»]

Текст задания находится на:
/practice_md/07_practice.md