¿Qué es dlt?

dlt es una biblioteca de Python de código abierto que carga datos de diversas fuentes, a menudo desordenadas, en conjuntos de datos bien estructurados y actualizados. Ofrece una interfaz ligera para extraer datos de APIs REST, bases de datos SQL, almacenamiento en la nube, estructuras de datos de Python y muchas más fuentes verificadas. Diseñada para ser fácil de usar, flexible y escalable, dlt automatiza el mantenimiento de pipelines con carga incremental, evolución de esquemas y contratos de datos.

Conceptos Fundamentales

Fuente (Source)

Una **fuente** es una agrupación lógica de recursos, es decir, endpoints de una única API. El enfoque más común es definirla en un módulo Python separado.

Las fuentes pueden definir opcionalmente un esquema con tablas, columnas, sugerencias de rendimiento y más. El módulo Python de la fuente típicamente contiene personalizaciones opcionales, transformaciones de datos, autenticación y código de paginación para una API particular.

Un **recurso** es una función (opcionalmente asíncrona) que genera datos.

Argumentos comúnmente utilizados

- **name**: El nombre de la tabla generada por este recurso. Por defecto es el nombre de la función decorada.
- **write_disposition**: Cómo deben cargarse los datos en el destino. Actualmente se admite: ***append, replace y merge***. Por defecto es append.

Características principales

- Agrupación lógica de endpoints de API
- Definición opcional de esquemas
- Personalización y transformación de datos
- Manejo de autenticación y paginación

Esquema (Schema)

dlt inferirá automáticamente el esquema para las tablas asociadas con los recursos a partir de los datos del recurso. Se puede modificar el proceso de generación utilizando sugerencias de tabla y columna.

Argumentos de recursos

- **table_name**: el nombre de la tabla, si es diferente del nombre del recurso.
- **primary_key** y **merge_key**: definen el nombre de las columnas (se permiten claves compuestas) que recibirán esas sugerencias. Se utilizan en la carga incremental y la carga por fusión.
- **columns**: permite definir una o más columnas, incluyendo los tipos de datos, nulabilidad y otras sugerencias.

Inferencia automática

dlt analiza los datos y determina automáticamente la estructura y tipos de datos apropiados para cada columna.

Claves primarias

Define identificadores únicos para registros, facilitando operaciones de actualización y fusión de datos.

Tipos de datos personalizados

Controla cómo se manejan estructuras complejas como JSON, listas y objetos anidados.

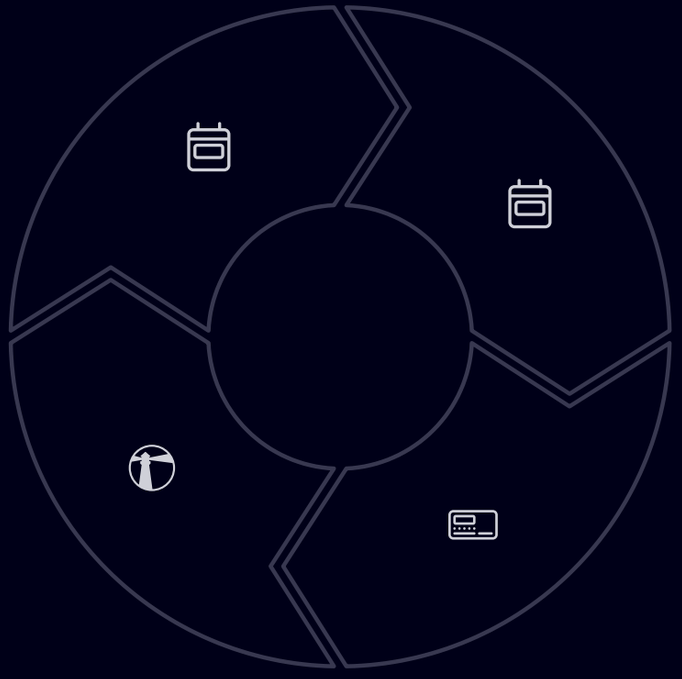
Pipeline

Un **pipeline** mueve datos desde tu código Python a un **destino**. El pipeline acepta fuentes o recursos dlt, así como generadores, generadores asíncronos, listas y cualquier iterable. Una vez que el pipeline se ejecuta, todos los recursos son evaluados y los datos se cargan en el destino.

Los pipelines son el componente central de dlt que orquesta todo el proceso de extracción, transformación y carga de datos. Proporcionan una interfaz unificada para mover datos desde cualquier fuente compatible hacia cualquier destino soportado.

Fuentes de datos

APIs, bases de datos, archivos o estructuras de datos de Python que contienen la información a cargar.



Procesamiento

Transformación, normalización e inferencia de esquemas durante la ejecución del pipeline.

Monitoreo

Seguimiento del estado, rendimiento y resultados de la ejecución del pipeline.

Destino

Almacenes de datos, bases de datos o servicios donde se cargan los datos procesados.

Los pipelines de dlt están diseñados para ser robustos y manejar automáticamente muchos de los desafíos comunes en la ingeniería de datos, como la evolución de esquemas, la carga incremental y la gestión de errores. Esto permite a los desarrolladores centrarse en la lógica de negocio en lugar de preocuparse por los detalles técnicos de la integración de datos.

Configuración del Pipeline

Para instanciar un pipeline, se llama a la función dlt.pipeline con varios argumentos que definen su comportamiento y destino. Este ejemplo muestra cómo un pipeline cargará una lista de objetos en una tabla DuckDB llamada "three":

```
import dlt
pipeline = dlt.pipeline(destination="duckdb", dataset_name="sequence")
info = pipeline.run([{'id':1}, {'id':2}, {'id':3}], table_name="three")
print(info)
```

pipeline_name

Nombre del pipeline que se utilizará para identificarlo en eventos de seguimiento y monitoreo, y para restaurar su estado y esquemas de datos en ejecuciones posteriores. Si no se proporciona, dlt creará un nombre de pipeline a partir del nombre del archivo del módulo Python que se está ejecutando actualmente.

destination

Nombre del destino al cual dlt cargará los datos. También puede proporcionarse al método run del pipeline. Determina dónde se almacenarán finalmente los datos procesados.

dataset_name

Nombre del conjunto de datos al que se cargarán los datos. Un conjunto de datos es un grupo lógico de tablas, es decir, un esquema en bases de datos relacionales o una carpeta que agrupa muchos archivos. Si no se proporciona, por defecto será {pipeline_name}_dataset en destinos que requieren conjuntos de datos.

La configuración adecuada del pipeline es crucial para garantizar que los datos se carguen correctamente en el destino deseado y con la estructura apropiada. Los parámetros proporcionados determinan no solo dónde se almacenarán los datos, sino también cómo se organizarán y nombrarán las tablas resultantes.

Carga de datos

Para cargar los datos, se llama al método run del pipeline y se pasan los datos en el argumento data. Este método es el punto de entrada principal para iniciar el proceso de extracción, transformación y carga de datos.

Argumentos principales

- **data** (el primer argumento): puede ser una fuente dlt, un recurso, una función generadora o cualquier Iterator o Iterable (es decir, una lista o el resultado de la función map). Este argumento contiene los datos que se cargarán en el destino.
- **write_disposition**: controla cómo escribir datos en una tabla. Por defecto es "append".
- **table_name**: se especifica en casos en que el nombre de la tabla no puede inferirse, por ejemplo, de los recursos o del nombre de la función generadora.

Modos de escritura

Modo	Descripción
append	Siempre agregará nuevos datos al final de la tabla.
replace	Reemplazará los datos existentes con nuevos datos.
merge	Deduplicará y fusionará datos basándose en las sugerencias primary_key y merge_key.

El método run devuelve información sobre la operación de carga, incluyendo estadísticas sobre la cantidad de datos procesados, tablas creadas o modificadas, y cualquier error o advertencia que haya ocurrido durante el proceso. Esta información es valiosa para monitorear y depurar pipelines de datos.

Entrada de datos

Proporcionar datos al pipeline a través del argumento data del método run.

Procesamiento

dlt procesa los datos según la configuración, infiere esquemas y normaliza estructuras.

Escritura

Los datos se escriben en el destino según el write_disposition especificado (append, replace o merge).

Resultados

El método run devuelve información detallada sobre la operación de carga realizada.