

Документација за Flink апликацијата

Оваа документација го опишува функционирањето на две главни датотеки во апликацијата: `flink_classes.py` и `streaming_transforms.py`. Апликацијата користи **Apache Flink** за обработка на податоци од **Kafka** во реално време. Главен фокус е датотеката `streaming_transforms.py`.

Инструкции за репликација на кодот / апликацијата се достапни на овој [линк](#).

1. `flink_classes.py`

Оваа датотека содржи класи и функции кои служат како основни building blocks за апликацијата.

Главни класи:

- **Record**: Структура на податок со атрибути `key`, `value`, и `timestamp`.
- **WindowStatistics**: Информации за прозорец, како минимална, максимална, просечна вредност и број на елементи.
- **WindowInfo**: Резимирани информации за прозорец.

Функции:

- **CreateRecordFromJSON**: Го конвертира JSON во објект `Record`.
- **WindowStatisticsToJSONMapper** и **WindowInfoToJSONMapper**: Ги конвертираат податоците во JSON формат.
- **ProcessWindowStatistics**: Генерира статистика за секој временски прозорец.
- **TimestampGetter**: Го вади временскиот печат (`timestamp`) од секој запис.

Помошни функции:

- **create_kafka_source()**: Креира Kafka извор.
 - **create_kafka_sink()**: Креира Kafka излез.
 - **create_watermark_strategy()**: Конфигурира стратегија за watermarks.
 - **create_env()**: Иницијализира Flink `StreamExecutionEnvironment`.
-

2. streaming_transforms.py

Главниот фајл кој ги дефинира трансформациите на податоците и го поврзува Flink со Kafka.

Конфигурација

- **Kafka Topics:**
 - **TOPIC:** Влезен topic - `sensors`.
 - **OUTPUT_TOPIC_1:** Излезен topic за статистики од прозорец - `results1`.
 - **OUTPUT_TOPIC_2:** Излезен topic за информации за прозорци - `results2`.
- **Параметри:**
 - **WINDOW_TIME:** Големина на прозорец.
 - **OUT_OF_ORDERNESS:** Максимално задоцнување.
 - **BOOTSTRAP_SERVERS:** Адреса на Kafka серверот.

Главни чекори:

1. **Иницијализација:**
 - Се вчитуваат сите `.jar` библиотеки од папката `jars`.
 - Се креира Flink environment со `create_env()`.
2. **Извор:**
 - Се додава Kafka извор со `create_kafka_source()`.
3. **Трансформации:**
 - **Обработка на записи:**
 - Претворање JSON во објекти `Record` преку `CreateRecordFromJSON`.
 - Додавање timestamps и watermarks со `assign_timestamps_and_watermarks()`.
 - Групирање според `key` и прозорец со `TumblingEventTimeWindows`.
 - Процесирање на прозорецот со `ProcessWindowStatistics`.
 - **Конверзии:**
 - Екстракција на резимиран информации со `WindowInfoExtractor`.
 - Конверзија на податоци во JSON со `WindowStatisticsToJSONMapper` и `WindowInfoToJSONMapper`.
4. **Излез:**
 - Мапираните податоци (`stats_stream_mapped` и `info_stream_mapped`) се испраќаат до Kafka излезните теми `OUTPUT_TOPIC_1` и `OUTPUT_TOPIC_2`.
5. **Извршување:**
 - Апликацијата се извршува преку `env.execute("Kafka Data Stream")`.

Белешки:

- **Логирање:** Апликацијата користи `logging` за следење на активностите.
- **Променливи:** Сите променливи како `WINDOW_TIME` и `TOPIC` лесно се конфигурираат во `streaming_transforms.py`.