AUDIENCE API V2.1

RD2 Weber Huang 2021-11-10

大綱

1. 專案說明
2. 專案流程
3. 專案工具
4. 專案環境建立
5. 專案使用方法
6. 錯誤代碼說明
7. 專案部屬需求

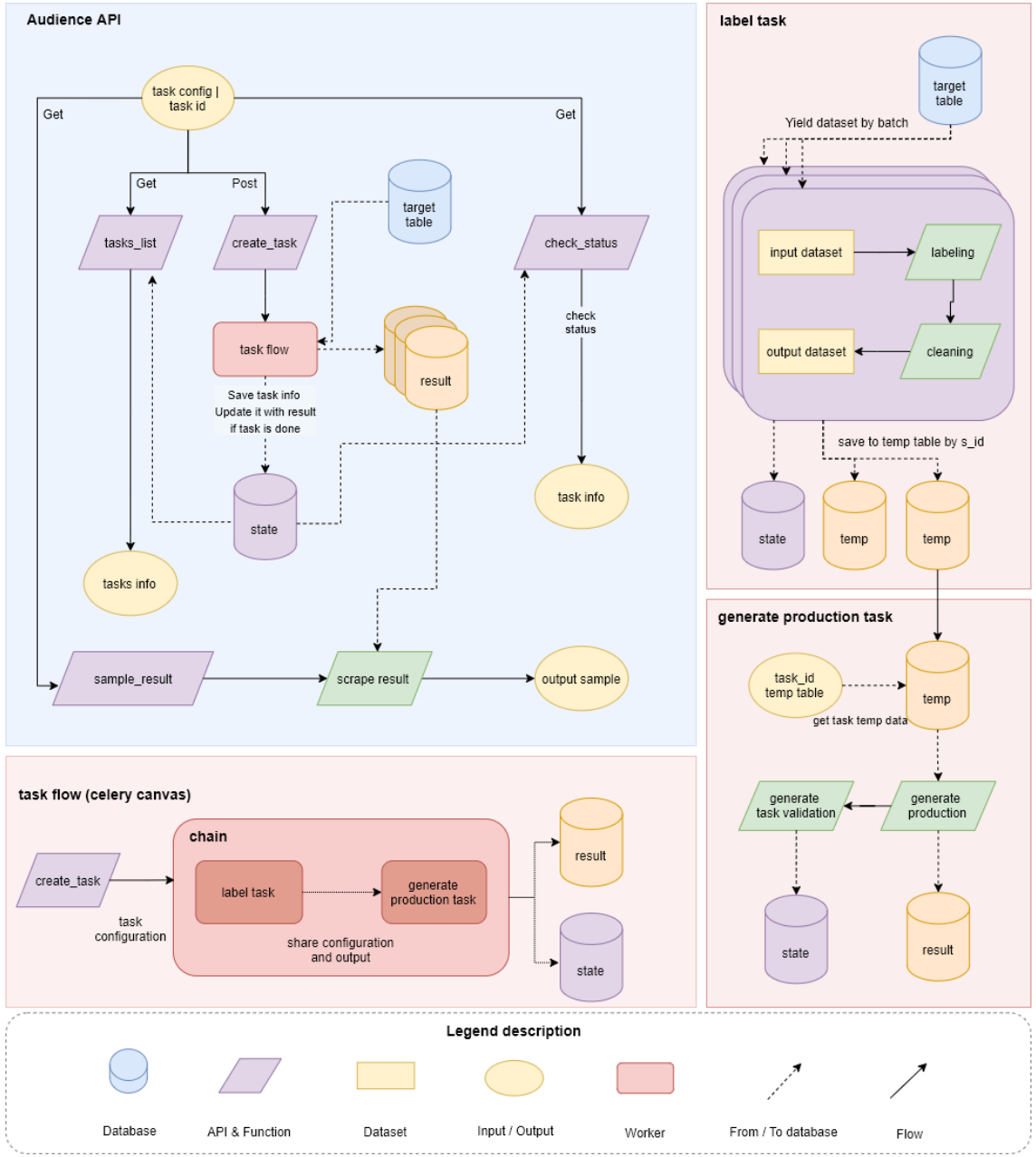
專案說明

此專案為協助 RD2 站台 (為了前後端分離) 進行貼標任務，支援使用者選擇貼標模型與規則，並且呼叫 API 回傳抽樣結果檢查貼概況。此專案共有四個 API 服務:

1. create_task : 依據使用者定義之情況建立、執行任務流程 (貼標 -> 上架)
2. task_list : 回傳近期五筆執行之任務與之相關資訊
3. check_status : 輸入任務ID，檢查任務進度與任務結果
4. sample_result : 輸入任務ID與任務結果，回傳抽樣之上架資料

專案流程

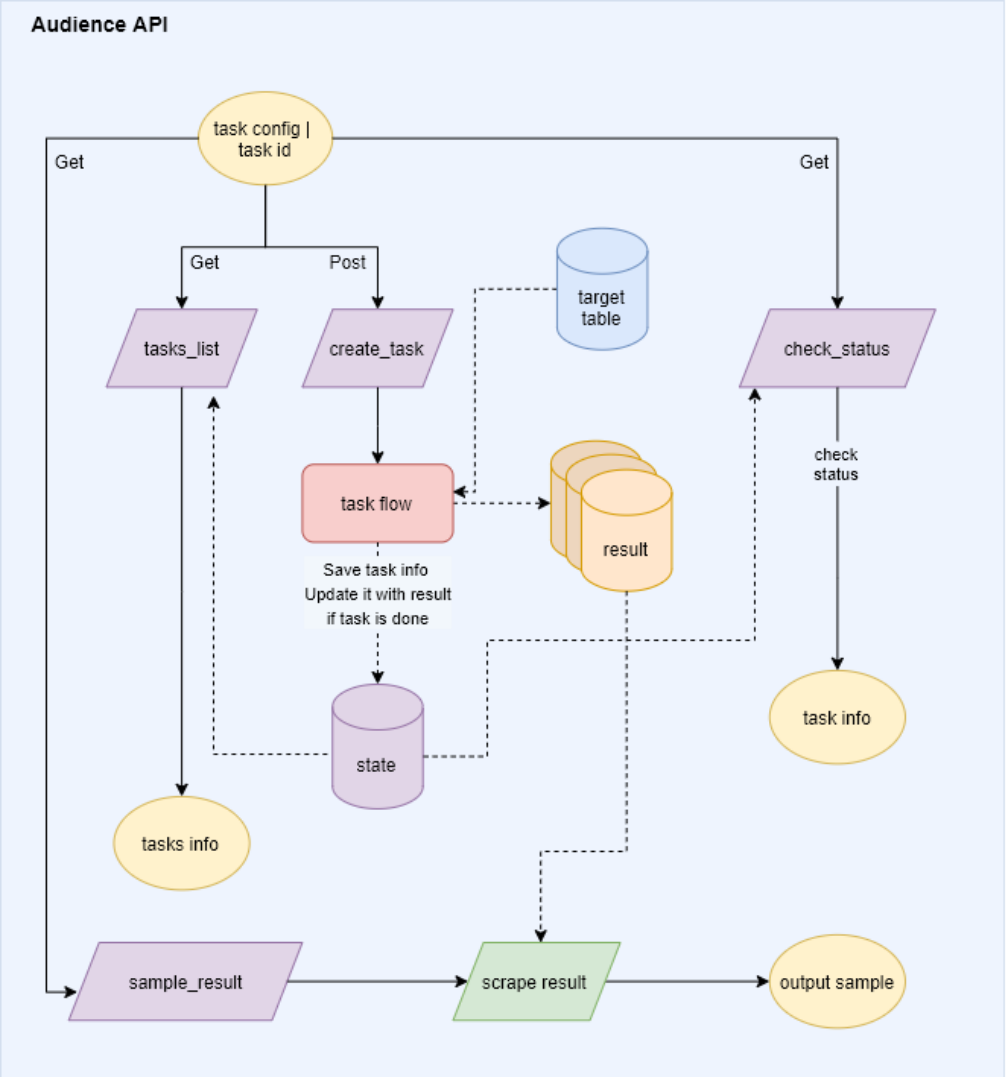
- API
- Task flow (celery canvas)
- Label task
- Generate production task



created by Weber Huang at 2021-10-14; last updated by Weber Huang at 2021-11-10

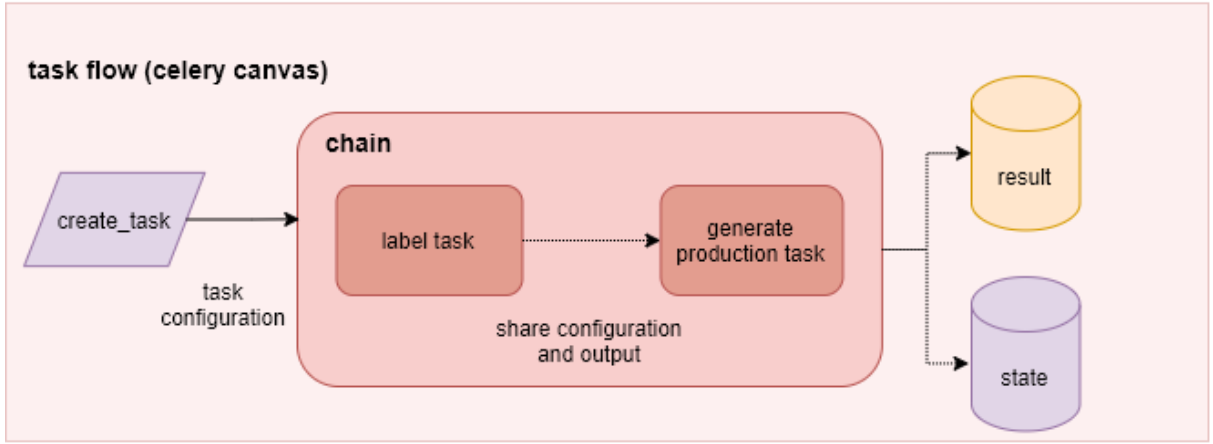
專案流程 (API)

- 使用者透過 create_task API POST 定義任務配置於 request body ，呼叫非同步任務執行爬取目標資料表內容進行貼標與上架流程。
- 執行中任務流程會將任務與結果相關資訊儲存於 state 追蹤資料表，上架結果資料會儲存於結果資料表。
- 使用者可以透過 task_list 訪問 state 回傳近期任務資訊；或呼叫 check_status輸入任務 ID 取得單筆任務資訊與結果資訊
- 當任務流程成功執行結束，使用者可以透過 sample_result 訪問結果資料表取得上架抽樣結果



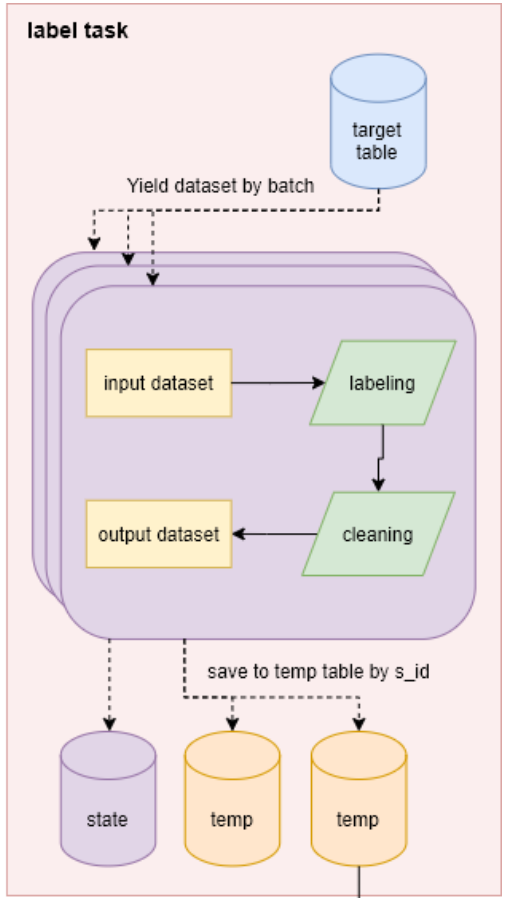
專案流程 (task flow)

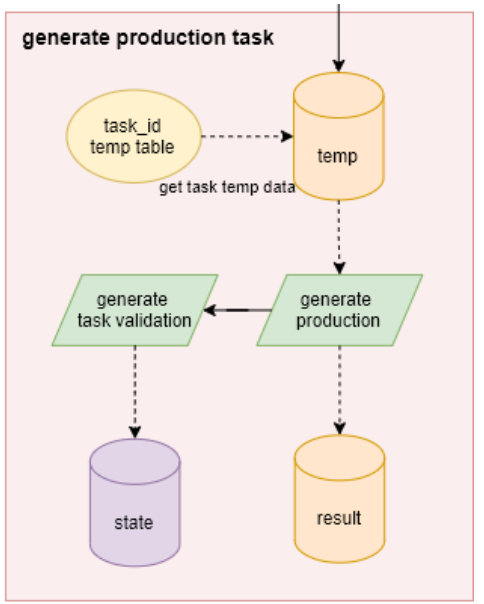
- 當使用者透過API建立 create_task 任務，create_task 會建立一組非同步任務流程。
- 該流程為先建立貼標任務，結束貼標後會將任務參數包含貼標結果共享給上架任務，並根據啟動上架任務
- 上架任務將貼標結果清理為不重複值並輸出至結果資料表，另外會儲存任務驗證資訊至 state 資料表。



專案流程 (tasks)

- 貼標任務接收到使用者定義之資料範圍，任務會透過生成器依據時間索引批次訪問資料庫。
- 處理貼標資料並批次儲存至暫存資料表。
- 過程中也會記錄貼標狀態至 state 資料表
- 上架任務根據貼標任務結果與共享參數，將暫存資料表中的貼標結果，清理並輸出至結果資料表。
- 上架任務也會記錄任務狀態與驗證資訊至 state 資料表。





專案工具

- 此專案使用以下環境與工具開發：
  - Windows 10
  - Docker
  - Redis
  - MariaDB
  - Python 3.8
  - Celery 5.1.2
  - FastAPI 0.68.1
- 此專案經由以下環境測試：
  - Windows 10 Python 3.8
  - Ubuntu 18.04.5 LTS Python 3.8

專案環境建立

- 建立流程：
  1. 先自行創建 Docker
  2. 建立專案環境匯入套件
  3. 設定環境變數
  4. 啟動 celery worker
  5. 啟動 API
- 詳細建立流程請參考 Audience API : Quick start

專案使用方法

- 使用者可以先透過 Swagger UI 網頁介面測試 API
- 或是經由 CURL 方法使用 API
- 詳細操作方法請參考 Audience API : Usage

錯誤代碼說明

- 此專案錯誤代碼分為：
  - HTTP錯誤代碼
  - 任務錯誤代碼
- 根據不同 API 任務，代碼會有不同的訊息
- 代碼詳細內容請參考 Audience API : Error code

專案部屬需求

- 系統：
  - Ubuntu 18.04.6 LTS
  - Windows 10 (不支援平行處理)
- Python : Python 3.8
- CPU : Intel(R) Core(TM) i5-8259U 同等或以上之處理器
- RAM : 16G 同等或以上

基本測試

- 資料筆數 : 2,376,186 rows
- 預測模型 : keyword base model
- 花費貼標時間 : 23.26 minutes
- 最大記憶體使用量 : 201.80 Mb