



我被要求提供api name，怎麼辦？



你被要求提供 API name，根據你提供的資料，我幫你初步整理出一份建議的 API name 清單，依照功能命名，方便前後端理解與使用：

◆ Table1: `model_info` 相關 API

API 名稱	方法	說明
`getModelListByDept`	GET	根據 `dept` 回傳除 `content` 外的所有欄位

API 名稱	方法	說明
`getModelContentById`	GET	根據 `model_id` 回傳 `content` 欄位 (JSON)
`createModelInfo`	POST	建立一筆 model_info
`updateModelInfo`	PUT	更新指定 `model_id` 的資料
`deleteModelInfo`	DELETE	刪除指定 `model_id` 的資料

◆ Table2: `user_form_info` 相關 API

API 名稱	方法	說明
`createUserForm`	POST	建立一筆 user_form_info, 初始化 `wf_status=null`
`getFormInfoById`	GET	根據 `model_id` 回傳表單資訊
`updateFormStatusAfterATCR`	PATCH	當 ATCR 回傳簽核結果時, 更新 `wf_status=pending` 並產生新 `wf_id`
`checkAndUpdateWorkflowStatus`	PATCH	查詢 workflow 任務狀態後更新 `wf_status` (running、completed、failed 等)
`deleteUserForm`	DELETE	刪除指定 `form_id` 的資料

◆ Workflow 狀態查詢相關 API

API 名稱	方法	說明
<code>`getWorkflowStatusById`</code>	GET	根據 <code>`wf_id`</code> 查 workflow 狀態

命名說明建議：

- 以動詞開頭（get、create、update、delete）符合 RESTful API 命名風格。
- 中駝峰命名（camelCase），保持一致性。
- 若你們公司已有 API 命名規範（如 snake_case 或 kebab-case），可依照既有規則調整。

如果你要做成 API 文件或 Swagger 規格書，我也可以幫你格式化製作。需要的話告訴我即可。是否要我幫你整理成表格或 markdown 呢？



格式化製作、整理成表格跟 markdown



以下是你需求的 **API Name** 表格，已格式化為 Markdown 格式（適合放入文件、Wiki 或 Swagger 概要頁使用）：



API 名稱整理表（``model_info`` 與 ``user_form_info``）



◆ Table1: `model_info` 相關 API

API 名稱	方法	說明
`getModelListByDept`	GET	根據 `dept` 回傳除 `content` 外的所有欄位
`getModelContentById`	GET	根據 `model_id` 回傳 `content` 欄位 (JSON)
`createModelInfo`	POST	建立一筆 `model_info` 資料
`updateModelInfo`	PUT	更新指定 `model_id` 的資料
`deleteModelInfo`	DELETE	刪除指定 `model_id` 的資料

◆ Table2: `user_form_info` 相關 API

API 名稱	方法	說明
`createUserForm`	POST	建立一筆 `user_form_info`，初始化 `wf_status = null`
`getFormInfoByModelId`	GET	根據 `model_id` 回傳表單內容
`updateFormStatusAfterATCR`	PATCH	當 ATCR 回傳簽核結果後，更新 `wf_status = pending` 並建立新 `wf_id`
`checkAndUpdateWorkflowStatus`	PATCH	查詢 workflow 任務狀態後更新 `wf_status` (running、completed、failed 等)
`deleteUserForm`	DELETE	刪除指定 `form_id` 的表單資料

◆ Workflow 狀態查詢 API

API 名稱	方法	說明
`getWorkflowStatusById`	GET	根據 `wf_id` 查詢目前 workflow 狀態

是否需要我接著幫你產出 Swagger YAML 或 OpenAPI 格式？或是幫你整理成 PDF 檔、Notion 規格文件形式？都可以幫忙處理。



Swagger YAML 、 OpenAPI 格式、整理成 PDF 檔、Notion 規格文件



✅ 以下是你要的 PDF 版 API 規格文件，已格式化整理完畢：

👉 [點此下載 PDF](#)

接下來我會提供：

1. **Swagger/OpenAPI YAML 格式**
2. **Notion 表格欄位建議與 Markdown 匯入說明**

你想先看哪一個呢？還是兩個都要？ [↔]





好的，這裡是你要的兩個格式：

✓ 1. Swagger / OpenAPI YAML 規格

以下為符合 OpenAPI 3.0 的 YAML 格式，可匯入 Swagger UI 或 Swagger Editor 檢視：

```
openapi: 3.0.0
info:
  title: API 規格文件
  version: 1.0.0
paths:
  /model-info:
    get:
      summary: 根據 dept 回傳除 content 外的所有欄位
      operationId: getModelListByDept
      parameters:
        - name: dept
          in: query
          required: true
          schema:
            type: string
      responses:
        '200':
          description: 成功

    post:
      summary: 建立一筆 model_info
      operationId: createModelInfo
      requestBody:
        required: true
```

```

        content:
          application/json:
            schema:
              $ref: '#/components/schemas/ModelInfo'
      responses:
        '201':
          description: 建立成功

/model-info/{model_id}:
  get:
    summary: 根據 model_id 回傳 content 欄位 (JSON)
    operationId: getModelContentById
    parameters:
      - name: model_id
        in: path
        required: true
        schema:
          type: string
    responses:
      '200':
        description: 成功
  put:
    summary: 更新指定 model_id 的資料
    operationId: updateModelInfo
    parameters:
      - name: model_id
        in: path
        required: true
        schema:
          type: string
    requestBody:
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/ModelInfo'
    responses:
      '200':
        description: 更新成功
  delete:
    summary: 刪除指定 model_id 的資料
    operationId: deleteModelInfo

```

```

parameters:
  - name: model_id
    in: path
    required: true
    schema:
      type: string
responses:
  '204':
    description: 刪除成功

/user-form:
  post:
    summary: 建立一筆 user_form_info
    operationId: createUserForm
    requestBody:
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/UserFormInfo'
    responses:
      '201':
        description: 建立成功

/user-form/{model_id}:
  get:
    summary: 根據 model_id 回傳表單內容
    operationId: getFormInfoByModelId
    parameters:
      - name: model_id
        in: path
        required: true
        schema:
          type: string
    responses:
      '200':
        description: 成功

/user-form/update-status:
  patch:
    summary: 更新 workflow 狀態
    operationId: checkAndUpdateWorkflowStatus

```



```
requestBody:
  content:
    application/json:
      schema:
        type: object
        properties:
          form_id:
            type: string
          status:
            type: string
responses:
  '200':
    description: 狀態更新成功
```

/workflow/{wf_id}/status:

```
get:
  summary: 根據 wf_id 查詢 workflow 狀態
  operationId: getWorkflowStatusById
  parameters:
    - name: wf_id
      in: path
      required: true
      schema:
        type: string
  responses:
    '200':
      description: 成功
```

components:

schemas:

ModelInfo:

```
type: object
properties:
  dept: { type: string }
  owner: { type: string }
  signflow_id: { type: string }
  model_id: { type: string }
  create_date: { type: string, format: date-time }
  update_date: { type: string, format: date-time }
  content: { type: string }
  wf_name: { type: string }
```

```
UserFormInfo:
  type: object
  properties:
    dept: { type: string }
    owner: { type: string }
    signflow_id: { type: string }
    model_id: { type: string }
    form_id: { type: string }
    create_date: { type: string, format: date-time }
    content: { type: string }
    wf_name: { type: string }
    wf_version: { type: string }
    wf_id: { type: string }
    wf_status: { type: string }
```

“ 可貼到 <https://editor.swagger.io/> 線上編輯與檢查。”

2. Notion 表格欄位建議與 Markdown 匯入說明

你可以在 Notion 中新增資料庫 (Database Table)，欄位建議如下：

API 名稱	方法	所屬模組	說明	備註
getModelListByDept	GET	model_info	根據 dept 回傳除 content 外的所有欄位	-
getModelContentById	GET	model_info	根據 model_id 回傳 content 欄位 (JSON)	-
createModelInfo	POST	model_info	建立一筆 model_info	-
updateModelInfo	PUT	model_info	更新指定 model_id 的資料	-
deleteModelInfo	DELETE	model_info	刪除指定 model_id 的資料	-

API 名稱	方法	所屬模組	說明	備註
createUserForm	POST	user_form_info	建立一筆 user_form_info	wf_status = null
getFormInfoByModelId	GET	user_form_info	根據 model_id 回傳表單內容	-
updateFormStatusAfterATCR	PATCH	user_form_info	ATCR 結果後更新 wf_status 並建立新 wf_id	-
checkAndUpdateWorkflowStatus	PATCH	user_form_info	查詢 workflow 任務狀態後更新 wf_status	-
deleteUserForm	DELETE	user_form_info	刪除指定 form_id 的資料	-
getWorkflowStatusById	GET	workflow	根據 wf_id 查詢 workflow 狀態	-

匯入方式：

1. 複製上面表格貼到 Notion 頁面。
2. Notion 會提示是否建立資料庫表格，點選「Create Table」即可。
3. 可手動加入欄位型別（如：API 名稱 → Title，方法 → Select，說明 → Text）。

是否需要我把 YAML 和 Markdown 也幫你匯出成檔案？我可以幫你打包 zip 給你下載。