Hello city操作說明

**目錄**

[**目錄** i](#_Toc34817257)

[圖目錄 i](#_Toc34817258)

[表目錄 ii](#_Toc34817259)

[1. Android Studio專案加入Zenbo SDK模組 1](#_Toc34817260)

[(1) 點選左上角 [ File ] -> [ New ] -> [ Import Module ] 1](#_Toc34817261)

[(2) 選擇想要加入的 [ Module ] 2](#_Toc34817262)

[(3) build.gradle 加入 dependencies 設定 3](#_Toc34817263)

[2. 建立Repository 3](#_Toc34817264)

[(1) 前置作業 3](#_Toc34817265)

[a. Windows平台上必裝的Git軟體工具 3](#_Toc34817266)

[3. GitHub Repository 5](#_Toc34817267)

[(1) 在GitHub上建立空白的Git儲存庫 5](#_Toc34817268)

[a. 網站資訊： 5](#_Toc34817269)

[4. Gmail外部收信設定 8](#_Toc34817270)

[(1) 進入Gmail設定頁面 8](#_Toc34817271)

[(2) 新增郵件帳戶 9](#_Toc34817272)

[a. 設定「寄件」地址 11](#_Toc34817273)

[b. 完成Gmail收發信設定 13](#_Toc34817274)

[參考資料 13](#_Toc34817275)

# 圖目錄

[圖 1‑2、Import Module 1](#_Toc34817276)

[圖 1‑3、import ZenboSDK module 2](#_Toc34817277)

[圖 1‑4、Import RobotActivityLibrary module 2](#_Toc34817278)

[圖 1‑5、build.gradle加入dependencies設定。 3](#_Toc34817279)

[圖 2‑1、Git官網 4](#_Toc34817280)

[圖 2‑2、GUI Clients 4](#_Toc34817281)

[圖 3‑1、Gmail信箱 5](#_Toc34817282)

[圖 3‑2、中山大學網路郵局 7](#_Toc34817283)

[圖 4‑1、點選Gmail設定功能 8](#_Toc34817284)

[圖 4‑2、進入Gmail帳戶設定。 8](#_Toc34817285)

[圖 4‑3、新增郵件帳戶 9](#_Toc34817286)

[圖 4‑4、輸入「郵件地址」 9](#_Toc34817287)

[圖 4‑5、設定POP伺服器連線資訊 10](#_Toc34817288)

[圖 4‑6、透過「本信箱」傳送郵件。 11](#_Toc34817289)

[圖 4‑7、設定名稱 11](#_Toc34817290)

[圖 4‑8、輸入「帳號」與「密碼」。 12](#_Toc34817291)

[圖 4‑9、輸入驗證碼。 12](#_Toc34817292)

[圖 4‑10、完成Gmail收發信設定。 13](#_Toc34817293)

# 表目錄

[表 4‑1、Cell(列, 欄) 10](#_Toc34817294)

# Android Studio專案加入Zenbo SDK模組

## 點選左上角 [ File ] -> [ New ] -> [ Import Module ]

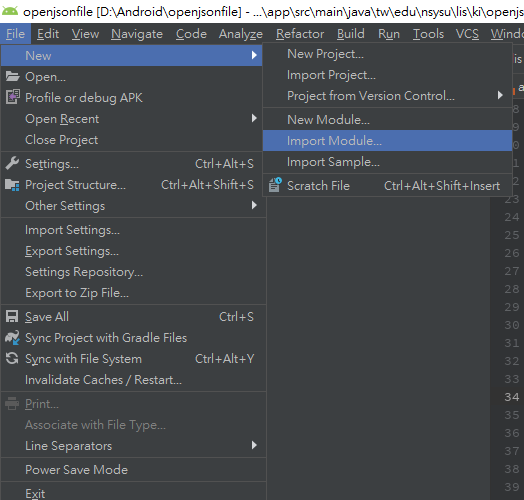


圖 1‑1、Import Module

## 選擇想要加入的 [ Module ]

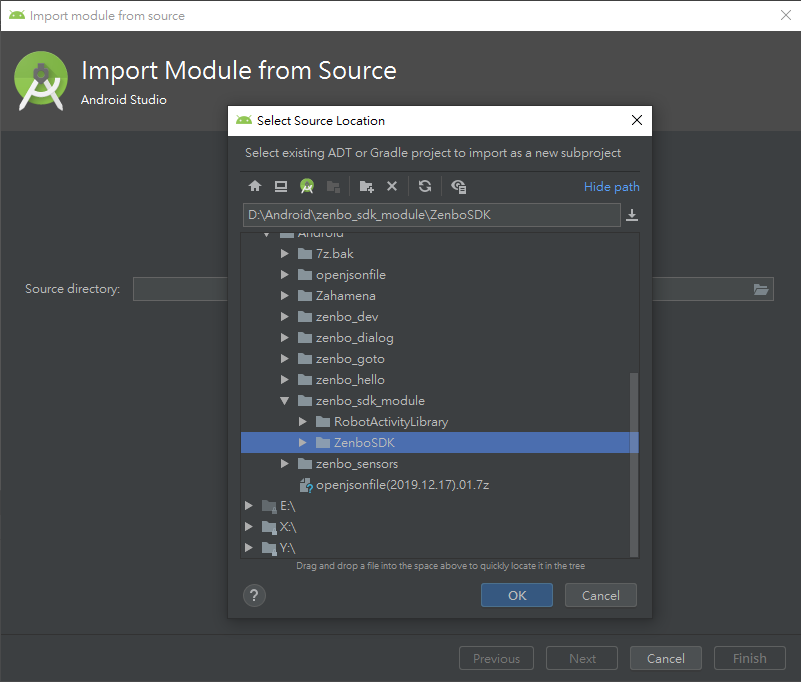


圖 1‑2、import ZenboSDK module

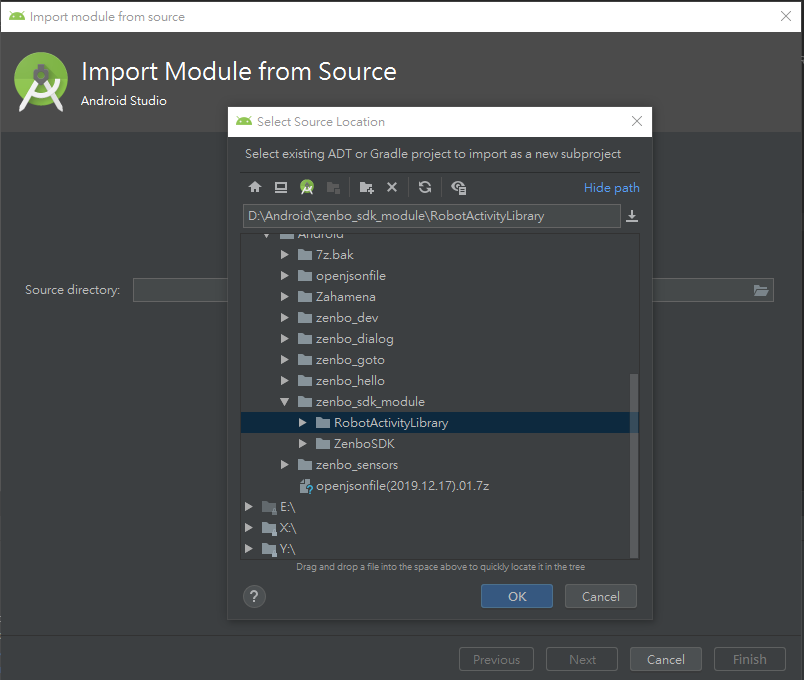


圖 1‑3、Import RobotActivityLibrary module

## build.gradle 加入 dependencies 設定

在app下的build.gradle加入dependencies 設定：

dependencies {

implementation project(':RobotActivityLibrary')

api files('../ZenboSDK/ZenboSDK.jar')

}

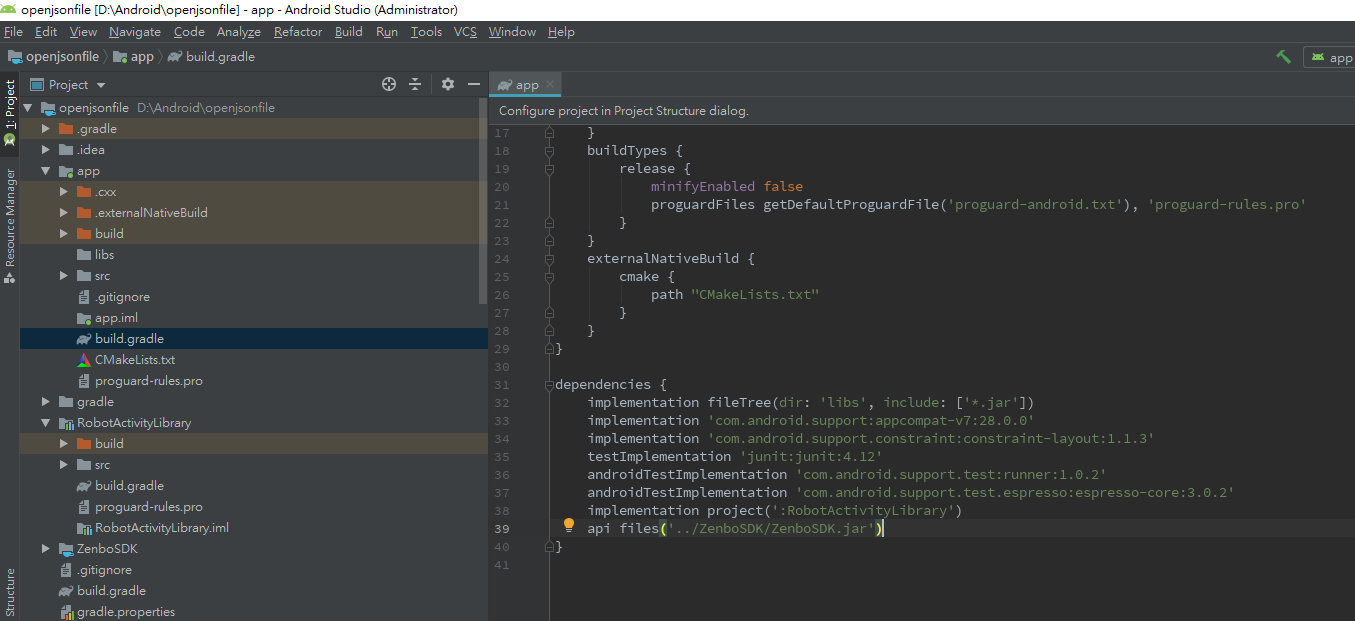


圖 1‑4、build.gradle加入dependencies設定。

# Zenbo App Management

App Managemenet相關設定都在網頁上匾及與設定。

## 網址：

<https://zenbo.asus.com/developer/console/>

## 新增App資訊：

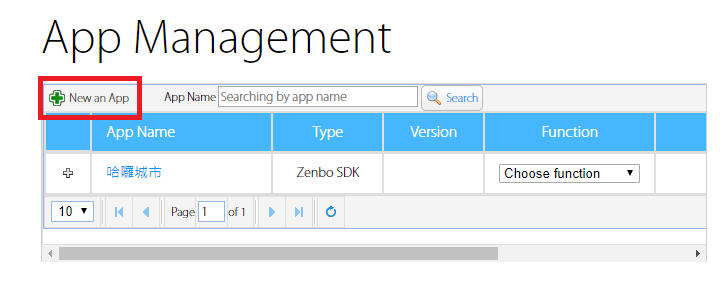


圖 2‑1、新增App資訊。

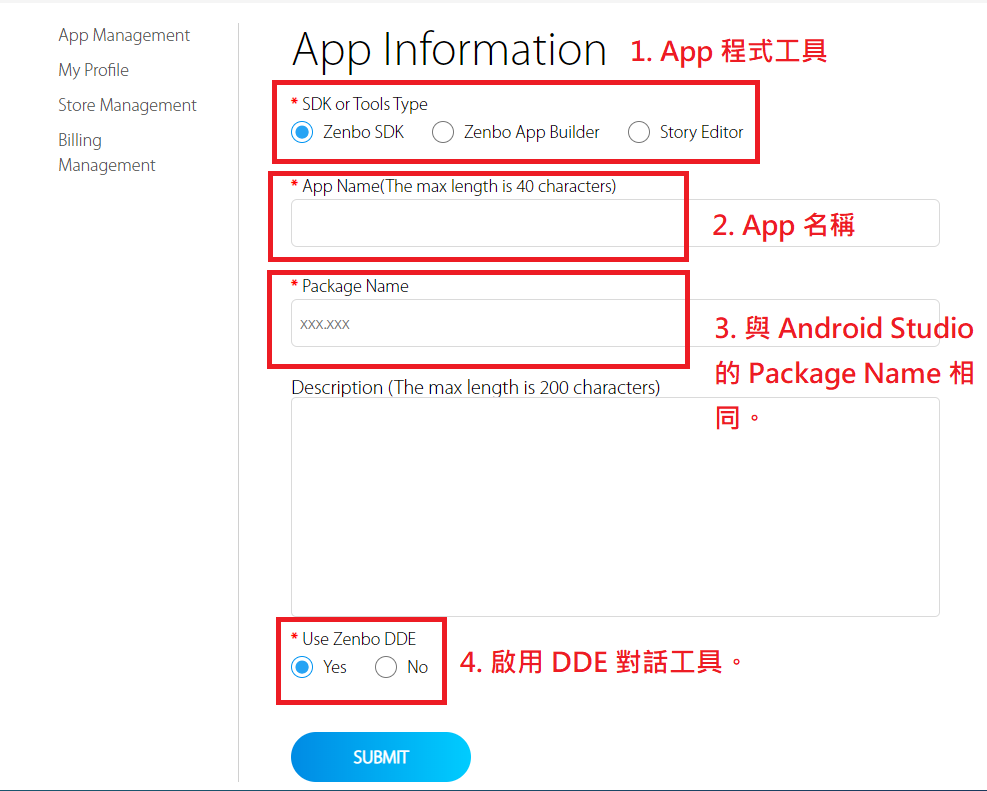


圖 2‑2、新增App Console步驟。

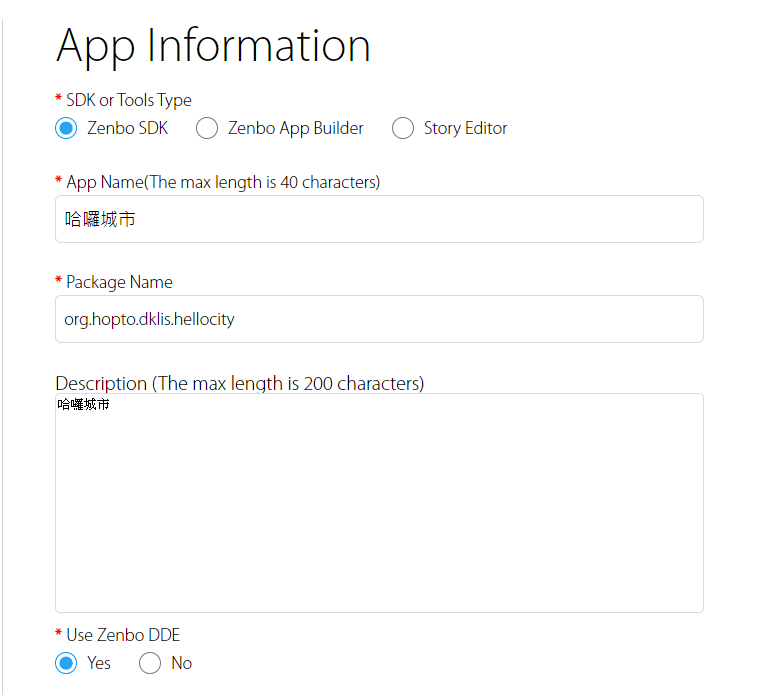


圖 2‑3、完成新增App Console (1)

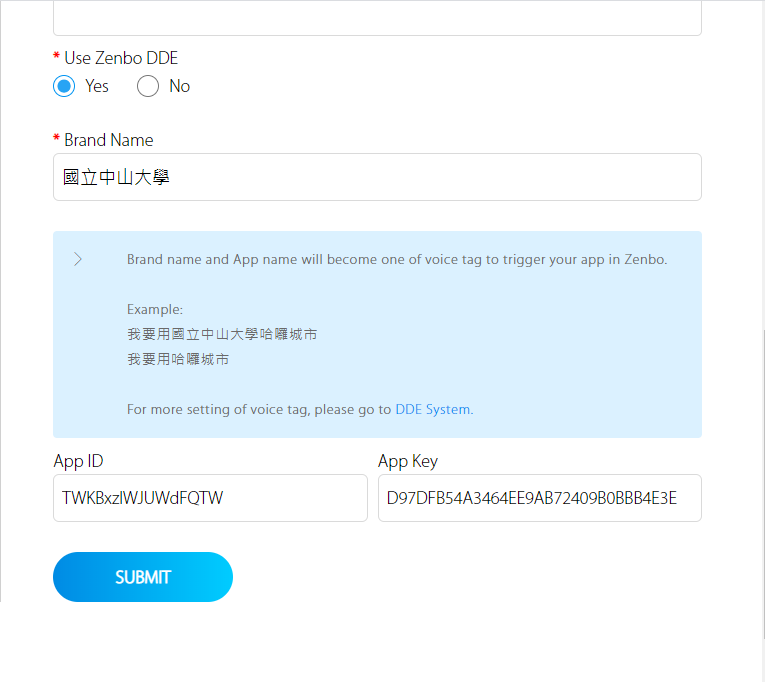


圖 2‑4、完成新增App Console (2)

# DDE Editor實作

同樣的DDE Editor都是在網頁上測試與編寫。

## 網址：

<https://zenbo.asus.com/developer/tools/ds-editor.jsp>

## 建立專案

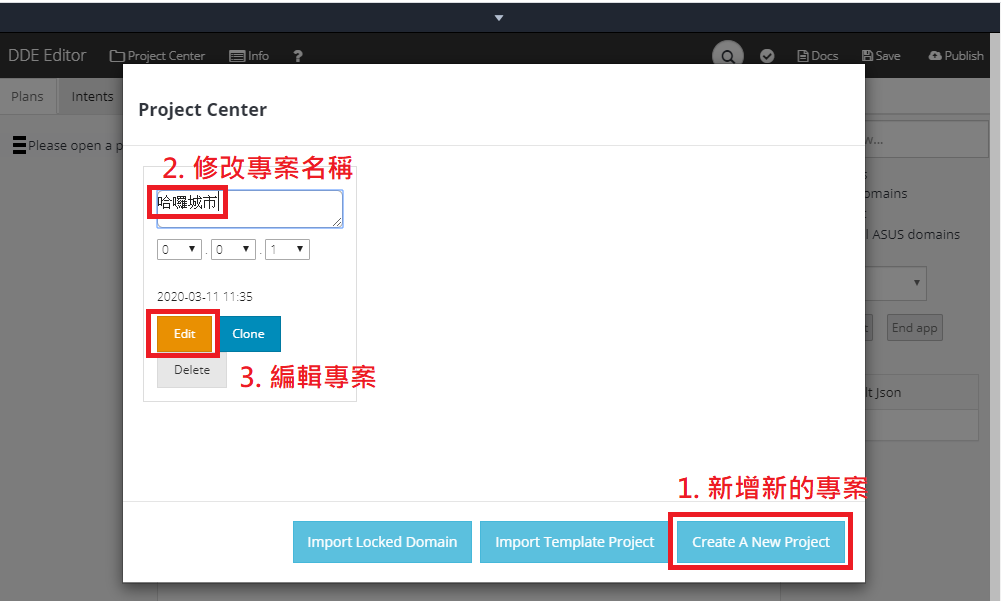


圖 3‑1、建立專案。

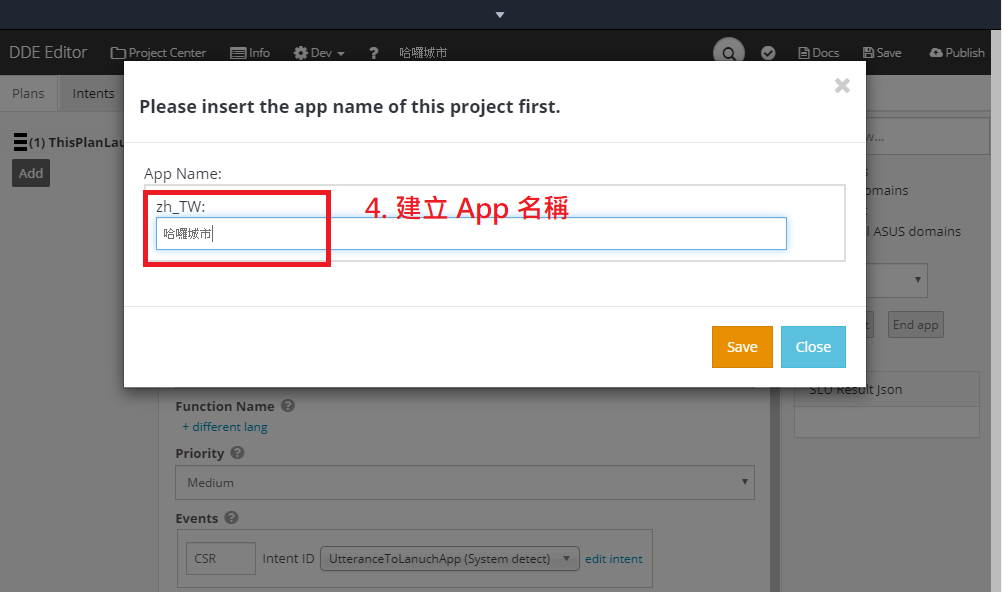


圖 3‑2、建立App名稱

## 必要設定

### 註冊「DDE Domain ID」資訊

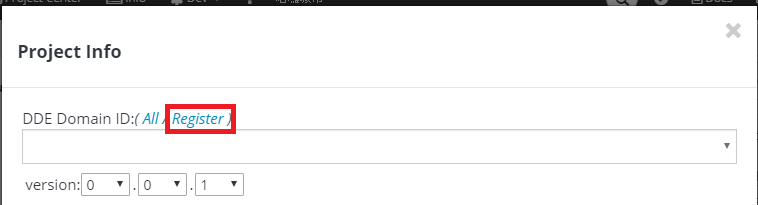


圖 3‑3、註冊「DDE Domain ID」資訊。



圖 3‑4、註冊「Domain Name」資訊。

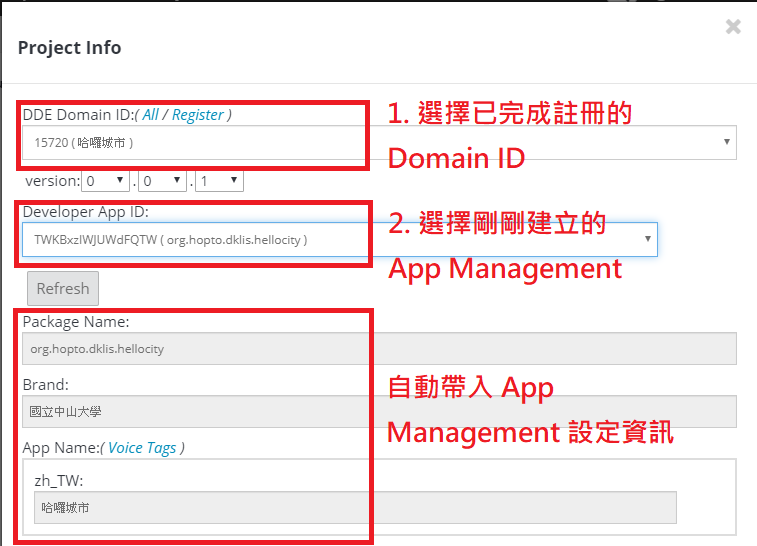


圖 3‑5、設定Domain ID與App Management資訊。

## Concept、Intent、Plan功能介紹

在語音對話過程是由多個句子組合而成，而句子的意圖以Plan來控制邏輯關係,，意圖的集合句型置於Intent之中，而Concept則是各個句型中所包含的詞彙集合。

所以設計「語音對話」建議先從「小」到「大」來設計對話方式，先從「Concept」🡪「Intent」🡪「Plan」方式設計。

### Plan

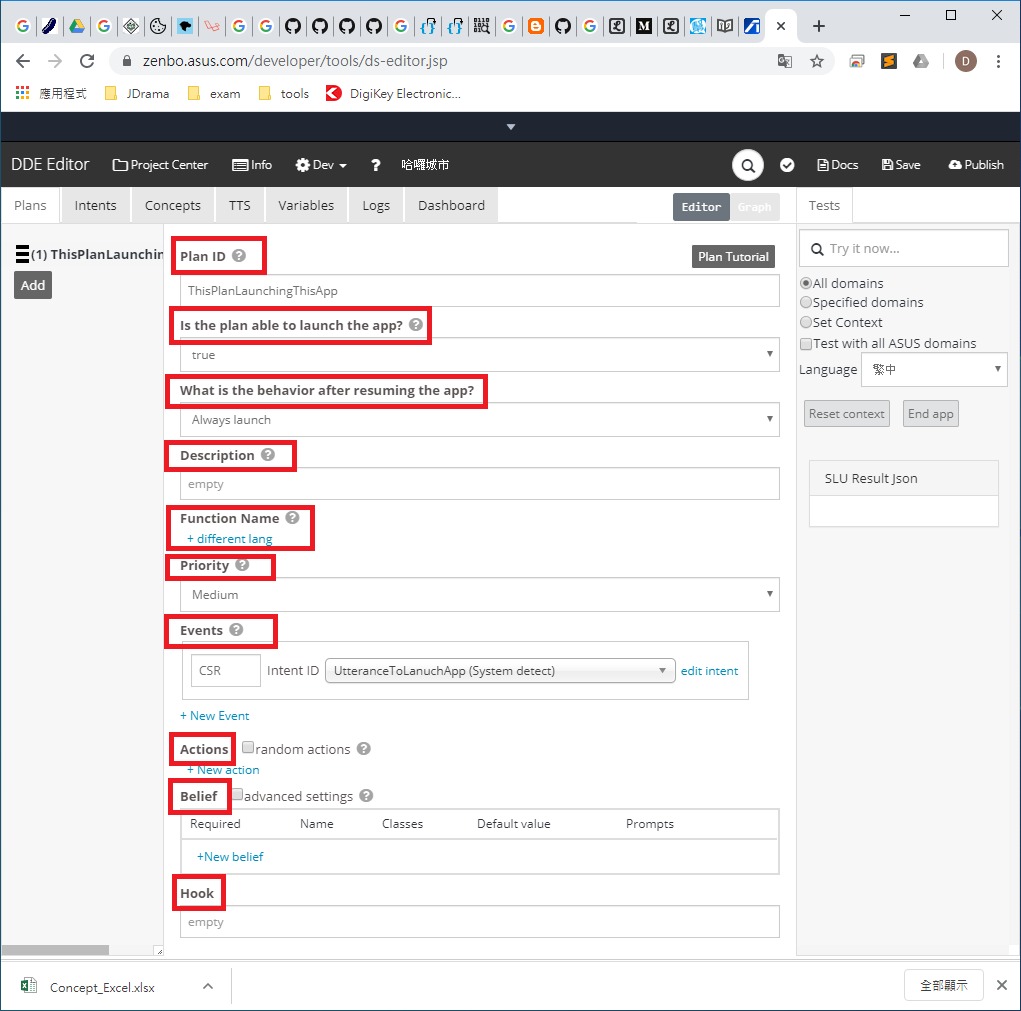


圖 3‑6、Plan說明。

表 3‑1、Plan項目說明

|  |  |
| --- | --- |
| 項目 | 說明 |
| Plans | Plan ID is the name of a plan. Every plan requires a unique plan ID. |
| Is the plan able to launch the app? | Set the plan as cross plan to launch the app or in-app plan.  True🡪 plan to launch the app.  False🡪 in-app plan. |
| What is the behavior after resuming the app? | To resume background apps, there are three ways, which are “Always launch”, “Always resume”, and “Always Resume but Set Plan Context”. If choosing the last one to resume the app, the Context’s settings will contain Plan’s Output Context. |
| Input Context | For example, set a plan’s Input Context as "A". It will be linked to other plan, which its Output Context is also set as "A". |
| Description | Description About the plan that you edit. |

# 參考資料

|  |  |
| --- | --- |
| [1] | 黃保翕, “30 天精通 Git 版本控管 (24)：使用 GitHub 遠端儲存庫 - 入門篇,” 電週文化事業, 24 10 2013. [線上]. Available: https://ithelp.ithome.com.tw/articles/10139849. |
| [2] | github, “github,” github, 2020. [線上]. Available: https://github.com/. |
| [3] | git, “git,” git, 2020. [線上]. Available: https://git-scm.com/. |