**知識網格元件編寫**

Ver. 0.2 2018.12.12

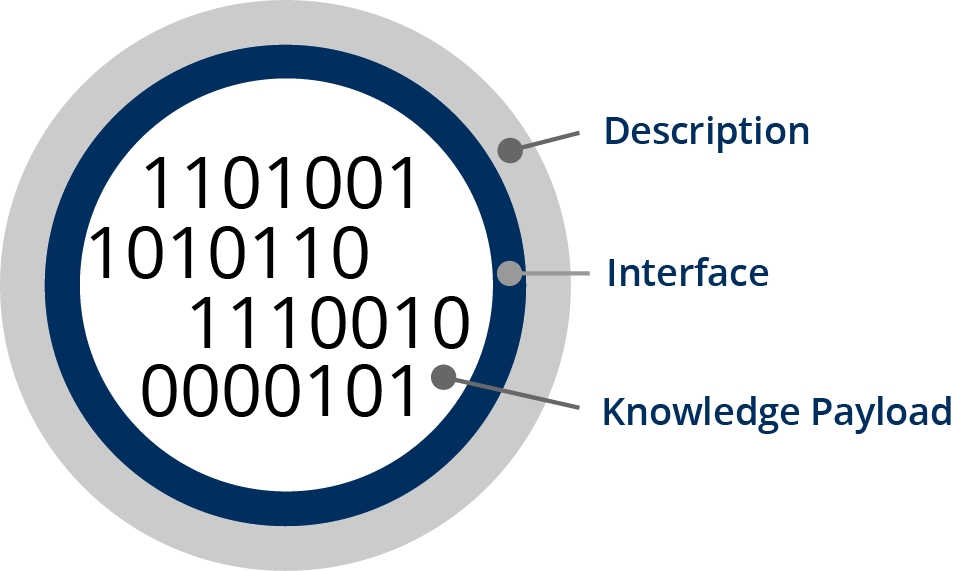
wucy1 @ github

說明：本文基於KGrid Activator 1.0.3版撰寫，新版的知識網格在知識元件的結構上與舊版(0.x版)有極大的差異，舊版的知識元件為單一檔案，新版的知識元件為一具有特定結構的資料夾，元件內程式碼(payload)的編譯工作依然是由adapter執行，但activator本體會內含javascript及proxy的adapter以便編譯元件內由javascript及python寫成的主程式碼(payload)。

作業環境：以下說明是基於在windows 10上執行知識網格的activator，但是在Ｍac或Linux上應該也可以直接運作。

如何編寫知識網格：

一、知識元件的結構概念

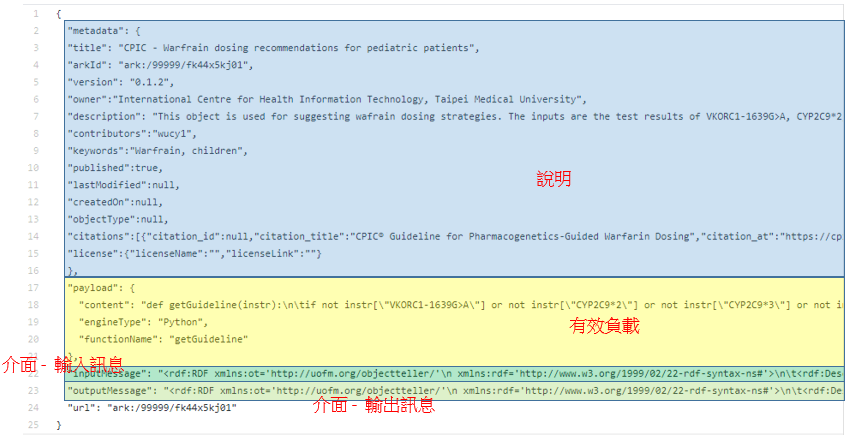


圖一、知識元件的組成(註一)

知識元件以模組化、計算機可處理的形式來儲存知識，由知識有效負載(payload/編碼成任何計算機語言或可計算格式)、詳細的描述(description)及介面(interface/含輸出入訊息)三部分組成(見圖一)；將知識可計算化涉及將健康知識表示成生產規則、網路圖、結構化判斷、方程式以及機率的技術，一旦這些技術應用於將知識可執行化與可計算化，便產生知識有效負載，被包裹於知識元件之中；知識元件有效負載的例子包括預測模型、可計算化的指引、計量問卷、可計算化的表型以及臨床決策支援規則等；知識元件以模組化、可計算機處理的形式表示知識資源，將其外部化並提供服務；關於知識元件結構的概念如上所述，但是在具體的編寫方式上，在新舊版本的知識網格中有很大的不同。

二、舊版知識元件格式

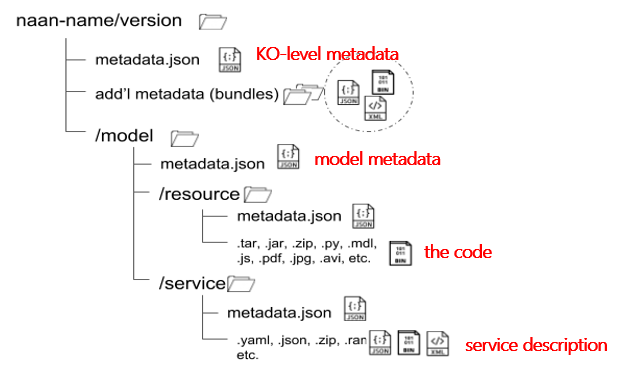
舊版的元件撰寫方式，是將上述三個部分整個以JSON格式打包到一個檔案中，以"CPIC - Warfrain dosing recommendations for pediatric patients"元件為例，各部分呈現如圖二：



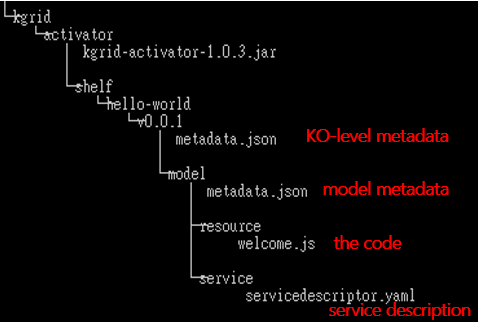
圖二、舊版知識元件格式

三、新版知識元件架構

新版的元件撰寫方式，是將上述三個部分，分別置於不同的檔案及目錄，整個元件的架構，如圖三所示；一個知識元件的要素有四項，即元件層級的metadata(KO-level metadata)、模型層級metadata(model metadata)、服務描述(service description)以及主程式碼(the code)；圖四所示為一包含四要素的最小知識元件範例。



圖三、新版知識網格元件架構(註一)



圖四、新版知識網格元件範例

四、元件層級的metadata(KO-level metadata)

以下各點以官方所附範例元件hello-world v0.0.1版為例說明，打開hello-world/v0.0.1目錄下的metadata.json，可以看到如下資訊：



圖五：元件層級的metadata.json(註二)

其中model跟service是最重要的兩個參數，model指定含有主程式的目錄，service則用來指定服務描述檔案。主程式跟服務描述檔案的位置基本上不需要按圖四的架構安排，只需要放在model參數指定的目錄下，然後在元件層級的metadata.json內指定服務描述檔案的位置，另外在Model層級的metadata.json指定主程式位置(見第五點)。

五、模型層級metadata(model metadata)

在model目錄下，要有另一個metadata.json檔案，以提供活化器所需要的資訊，包含函式名稱(functionName)、主程式位置(resource)、adpater類型(adapterType)及詳細資訊(adapters)，以便部署主程式做為REST(表現層狀態轉換/Representational State Transfer)服務的端點(endpoint)。

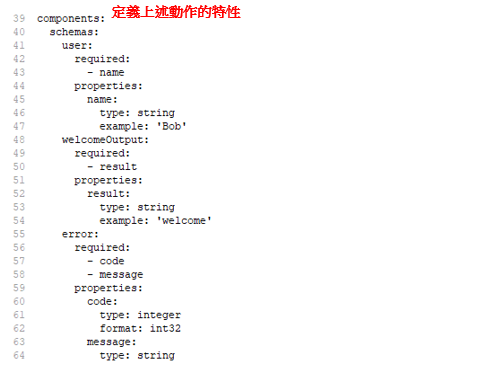


圖六、Model層級的metadata.json(註二)

六、服務描述(service description)

範例元件所提供的服務則由model/service/servicedescriptor.yaml這個檔案來進一步定義，這個檔案的功能涵蓋圖一所提到的介面。檔案各部分內容概略功能標記如圖七，各項參數具體功能請參考註三。





圖七、服務描述檔內容

七、主程式碼(the code)

範例元件的主程式碼welcome.js，包含一welcome函式，功能為接收一json格式的inputs，擷取其中的name參數，然後返回歡迎訊息。然後再由activator根據服務描述檔內的定義，以json格式回傳資訊。



圖八、主程式碼內容

八、知識元件的執行

範例元件可以在終端介面以curl指令測試：

curl -X POST -H "Content-Type:application/json" -d "{\"name\": \"Fred Flintstone\"}" <http://localhost:8080/hello/world/v0.0.1/welcome>

以上指令意義為：

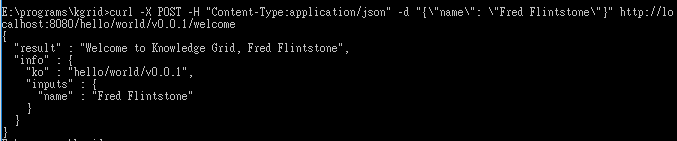
1. 呼叫服務端點<http://localhost:8080/hello/world/v0.0.1/welcome>

以POST方式，透過Activator呼叫，每個元件的服務端點由前述服務描述檔定義

1. 向此端點以json格式送出查詢資料

此資料的格式跟必要欄位，也是由服務描述檔定義

1. 若查詢資料有誤，會依服務描述檔的定義傳回錯誤訊息，若正確，則會將資料傳送給主程式檔，等待主程式處理後，傳回訊息如下：



圖九、範例元件呼叫結果

註一：<http://kgrid.org/tools.html#object>

註二：<https://kgrid.org/getting-started/developing-kos/>

註三：<https://swagger.io/docs/specification/about/>

誌謝：

知識網格中文技術手冊之編寫係由以下單位及計畫支持 -

臺北醫學大學健康資訊科技國際研究中心

國家衛生研究院-「亞太生醫矽谷精準醫療旗艦計畫-臺北醫學大學合作案」

臺北醫學大學-市立萬芳醫院專題研究計畫-「打造亞太第一家學習型健康系統醫院」