**一、RestFul簡介**

　　REST（Representational State Transfer 通常被翻譯為“表述性狀態傳輸”或者“表述性狀態轉移）是Roy Fielding提出的一個描述互聯系統架構風格的名詞。為什麼稱為REST？Web本質上由各種各樣的資源組成，資源由URI 唯一標識。流覽器（或者任何其它類似於流覽器的應用程式）將展示出該資源的一種表現方式，或者一種表現狀態。如果使用者在該頁面中定向到指向其它資源的連結，則將訪問該資源，並表現出它的狀態。這意味著用戶端應用程式隨著每個資源表現狀態的不同而發生狀態轉移，也即所謂REST。

　　簡單地來說REST它是一種使用URL來定位資源，使用HTTP請求描述操作的Web服務規範。REST主要包括以下幾方面：

　　（1） REST是一組架構約束條件和原則，而滿足這些約束條件和原則的應用程式就是RESTful。   
　　（2）REST的目標是構建可擴展的Web Service，它是一種更簡單的SOAP(Simple Object Access Protocol)協議以及以WSDL為基礎的WebService的替代。   
　　（3）REST採用的是HTTP協議並通過HTTP中的GET、POST、PUT、DELETE等動詞收發資料。   
　　（4） REST希望通過HTTP來完成對資料的元操作，即傳統的CRUD(Create、Read、Update、Delete)分別對應GET、POST、PUT、DELETE，這樣就統一了資料操作的介面，實現在不同平臺上提供一套相同的服務。   
　　（5） REST是一種面向服務的、分散式的API設計風格。

　　RESTful API的開發和使用，無非是用戶端向伺服器發請求（request），以及伺服器對用戶端請求的回應（response）。所以RESTful架構風格具有統一介面的特點，即：使用不同的http方法表達不同的行為：

* **GET（SELECT）：從伺服器取出資源（一項或多項）**
* **POST（CREATE）：在伺服器新建一個資源**
* **PUT（UPDATE）：在伺服器更新資源（用戶端提供完整資來源資料）**
* **PATCH（UPDATE）：在伺服器更新資源（用戶端提供需要修改的資來源資料）**
* **DELETE（DELETE）：從伺服器刪除資源**

**二、REST的約束條件和原則**

　　REST本質上是Web服務的一種規範，一種思想。它主要包括以下特性：

**1、資源(Resources)**

　　在REST中資源是整個架構或者說整個網路處理的核心，那麼什麼又是資源呢？在我們傳統的觀念中，資源是指伺服器上的一個檔，而在REST裡資源則是指一個URL。URL即統一資源定位，而我們都知道通過URL可以訪問互聯網上的資源，所以在REST裡這種對資源的指向性更加強烈，並且在這裡資源的範疇會被無限放大而並非局限在檔本身，例如以下實例：

1 http://api.cnblogs.com/info/source 表示獲取某人的成績

2 http://api.cnblogs.com/info/friends 表示獲取某人的好友列表

3 http://api.cnblogs.com/info/profile 表示獲取某人的詳細資訊

　　由此我們注意到REST在形式上更加趨向API設計，而我們獲取的資源則通過一定的形式進行統一而規範化的表達，因此REST實現了讓不同的平臺共用一套API這樣的願望，這是一件非常美好的事情，這個世界上的技術陣營舉不勝數，而它們為了各自的利益建立一套封閉、臃腫的體系框架，很多時候當我們不需要這樣的“全家桶”並且希望“跨平臺”的時候，REST將會是一個不錯的選擇。

**2、表現形式(Representational)**

　　在REST中表現形式作為我們對資源請求的一個結果的呈現，通過對HTTP協議的學習我們已經知道，伺服器會給用戶端返回什麼形式的資訊，這一點取決於伺服器回應報文中相關頭部欄位，而**對REST來講，它通常會採用XML或者JSON來告訴請求者請求的結果，因為JSON相比XML所含的冗餘資訊較少，所以目前更加傾向於或者說流行使用JSON作為請求結果的表現形式**。

**3、狀態變化(State Transfer)**

　　雖然我們一再強調HTTP協定是無狀態，這主要體現在HTTP請求與請求、HTTP回應與回應的上下文無關性上。在REST中，我們所說狀態變化更多是指HTTP中的GET、POST、PUT、DELETE等動詞實現。具體來講，看下面的簡單示例：

1. GET http://url/info 表示獲取全部的info
2. POST http://url/info 表示創建一個新的info
3. GET http://url/info/{id} 表示獲取一個指定id的info
4. PUT http://url/info/{id} 表示更新一個指定id的info
5. DELETE http://url/info/{id} 表示刪除一個指定id的info

　　除此之外，我們注意到REST基於HTTP協定，所以HTTP協定中的狀態碼對它來講同樣適用，例如最常用的200表示成功、500表示伺服器內部錯誤、404表示無法找到請求資源、400表示請求錯誤等等。

**三、如何構建RestFul風格的API**

　　如何構建REST風格的API？我們可以通過以下實例說明

* **URLRoot採用下面這樣的結構：**

1. <http://example.com/api/v1/>
2. http://api.example.com/v1/

* **API版本可以放在URL或者HTTP的Header裡**
* **URL使用名詞而非動詞：**

1. http://example.com/api/v1/getProducts 這是一個糟糕的設計
2. GET http://example.com/api/v1/products 這是一個優雅的設計

* **保證方法時安全的不會對資源狀態有所改變。例如：**

GET http://example.com/api/v1/deleteProduct?id=1 這是一個危險的信號

* **資源的位址推薦使用嵌套結構**

GET http://example.com/api/v1/friends/123456789/profile

* **使用正確的HTTP狀態碼表示訪問狀態**
* **返回含義明確的結果(一般推薦JSON)**