Representational State Transfer，簡稱REST，它是一種網路架構風格，他並不是一種標準。

以 API 而言，假設我們正在撰寫一組待辦事項的 API，  
可能會有以下方式來作為 API 的 url:

獲取使用者資料 /getAllUsers  
獲取使用者資料 /getUser/1  
新增使用者資料 /createUser  
更新使用者資料 /updateUser/1  
刪除使用者資料 /deleteUser/1

若是以 REST 風格來開發 RESTful API 的話:

獲取使用者資料 /GET /users  
獲取使用者資料 /GET /user/1  
新增使用者資料 /POST /user  
更新使用者資料 /PUT /user/1  
刪除使用者資料 /DELETE /user/1

兩者差異是在於 RESTful API 充分地使用了 HTTP protocol (GET/POST/PUT/DELETE)，達到:

1. 以直觀簡潔的資源 URI
2. 並且善用 HTTP Verb
3. 達到對資源的操作
4. 並使用 Web 所接受的資料類型: JSON, XML, YAML 等，最常見的是 JSON

額外(自己的筆記):  
API回傳資料:  
回傳資料時，也要注意幾件事情。首先由於 API 大多由 Ajax 的方式去呼叫，也就是說 Client 端（如網頁）是在背景再乎叫一次這個 API，得到資料後再顯示在前景，所以要回應不同的狀態讓 Client 知道狀況再去決定要幹麻，**設計 API 時不要轉址**。

舉個例子，註冊成功要轉到登入頁面應該是在 POST /user 後回傳成功，讓前端自己去決定失敗的話要顯示什麼訊息、成功的話要轉址到哪。畢竟設計 API 的時候你不知道前端工程師會不會想要顯示個五秒的動畫再震動之類的。