

Web API 通訊組件

Shih-Hao Liu

Nov. 11, 2020

Serialization

- What ?

- Ans :

序列化是將物件狀態轉換為可保持或可傳輸的形式的過程。

序列化的補集是反序列化，後者將流轉換為物件。

這兩個過程一起保證資料易於儲存和傳輸。

Serialization

- When ?
- Ans :

物件序列化可以實現分散式物件。主要應用例如：RMI要利用物件序列化執行遠端主機上的服務，就像在本地機上執行物件時一樣。

物件序列化不僅保留一個物件的資料，而且遞迴儲存物件引用的每個物件的資料。可以將整個物件層次寫入位元組流中，可以儲存在檔案中或在網路連線上傳遞。利用物件序列化可以進行物件的"深複製"，即複製物件本身及引用的物件本身。序列化一個物件可能得到整個物件序列。

Serialization

- Benefit ?

- Ans :

- 資料需要序列化以後才能在服務端和客戶端之間傳輸。

不序列化也可以傳輸，但是無法跨平臺，安全性也無法保障在程序內通訊，也可以做到更高的安全性，物件不再通過呼叫棧互動，而是通過代理或通道。

Serialization

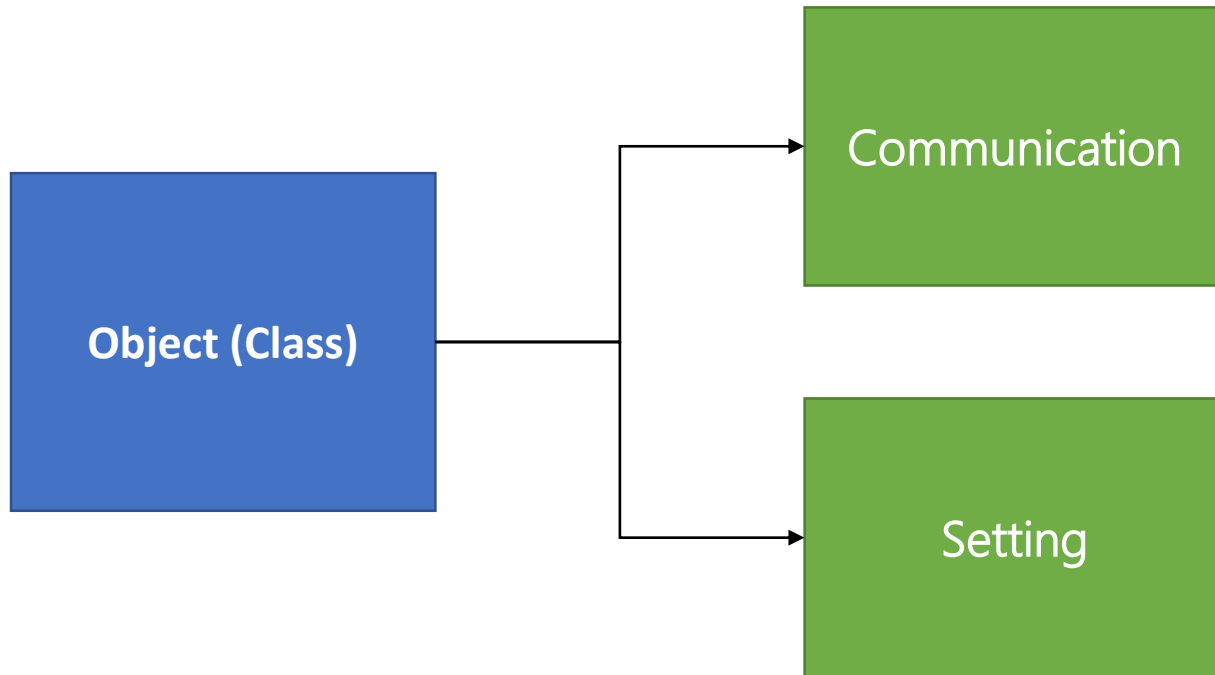
- 序列化就是一種用來處理物件流的機制，所謂物件流也就是將物件的內容進行流化。

我們可以對流化後的物件進行讀寫操作，也可將流化後的物件傳輸於網路之間(注：要想將物件傳輸於網路必須進行流化)！在對物件流進行讀寫操作時會引發一些問題，而序列化機制正是用來解決這些問題的！

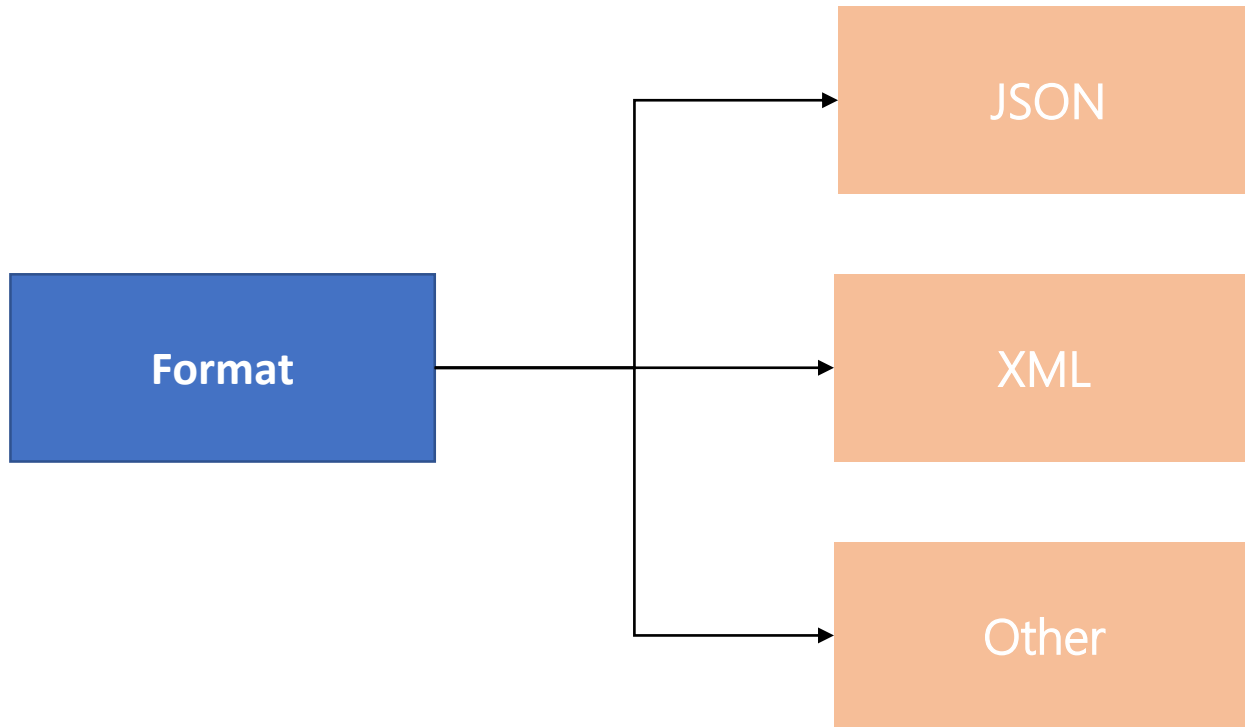
- 描述資料的傳輸格式，這樣可以方便自己組織資料傳輸格式，以至於避免一些麻煩及錯誤。

如果是跨平臺的序列化，則傳送方序列化後，接收方可以用任何其支援的平臺反序列化成相應的版本，比如Java序列化後，用.net、python等反序列化。

Serialization



Serialization



JSON

- 數值(Value) : short, int, long, float, double...
- 字串(String) : 以雙引號 "" 括起來。
- 布林值(Bool) : true or false。
- 陣列(Array) : 使用方括號[,]括起來。元素之間用逗號,分割。
例如 : [value, value]
- 物件(Object) : 物件以花括號{開始, 並以}結束。鍵-值對之間使用逗號分隔。鍵與值之間用冒號:分割。
- null類型 : 值寫為null。

JSON

```
{
  "Name" : "John",
  "age" : 25,
  "address" :
  {
    "streetAddress": "21 2nd Street, New York ",
    "postalCode": "10021 "
  },
  "phoneNumber":
  [
    { "type" : "home" , "number" : "212 555-1234" },
    { " type " : " fax " , "number" : "646 555-4567" }
  ]
}
```

Web Service Interface

- 網路位置
- URL(內網)[測試用] : <http://{IP}:{PORT}/AGV/Test>
- URL(內網) : <http://{IP}:{PORT}/AGV/SendAgvCmd>
- IP : 192.168.101.234
- PORT : 50100
- URL(外網)[測試用] : <http://{IP}:{PORT}/WebTest>
- IP : 59.124.226.9
- PORT : 6592

- WIFI
- NAME : CAR_AP
- PASSWORD : 11111111

Web Service Interface

- 設定內容
- Content-Type : "application/json"
- Method : "POST"
- 以C#平台為範例

```
string url = "http://localhost:50100/AGV/SendAgvCmd";  
HttpWebRequest req = (HttpWebRequest)WebRequest.Create(url);  
req.ContentType = "application/json";  
req.Method = "POST";
```

Web Service Interface

- 輸入參數為

```
{  
  " Name " : "AGV名稱",  
  " Cmd " : "命令代碼",  
  " Param " : [參數1, 參數2, ...]  
}
```

- 輸出參數為

```
{  
  " StatusCode " : 結果碼,  
  " Config " : { 狀態內容 },  
}
```

Web Service Interface

執行命令	命令代碼說明	輸入參數	參數說明
500	取得AGV狀態資料		
30100	Servo ON(鎖定馬達模式)		
30101	Servo OFF(馬達解鎖模式)		
30108	右轉90度		
30109	右轉180度		
30110	左轉90度		
30111	左轉180度		
30208	右轉90度(轉盤不動)		
30209	右轉180度(轉盤不動)		
30210	左轉90度(轉盤不動)		
30211	左轉180度(轉盤不動)		

Web Service Interface

執行命令	命令代碼說明	輸入參數	參數說明
30214	轉盤右轉90度		
30215	轉盤右轉180度		
30216	轉盤左轉90度		
30217	轉盤左轉180度		
30218	轉盤上升(含自動對位)		
30219	轉盤下降(含自動對位)		
30310	緊急停止		
30312	腳本暫停		
30313	腳本繼續		
30314	腳本停止		

Web Service Interface

執行命令	命令代碼說明	輸入參數	參數說明
30112	邊走邊修(直走)	1000~50000	mm
30230	前進充電站	300~1000	mm
30231	離開充電站	300~1000	mm
30121	執行腳本	請參考後續表格	

Web Service Interface

腳本代碼	腳本代碼說明
50	右轉90度
60	右轉180度
70	左轉90度
80	左轉180度
150	右轉90度(轉盤不動)
160	右轉180度(轉盤不動)
170	左轉90度(轉盤不動)
180	左轉180度(轉盤不動)
300	腳本暫停
330	轉盤上升(含自動對位)
340	轉盤下降(含自動對位)

Web Service Interface

腳本代碼	腳本代碼說明
350	轉盤右轉90度
360	轉盤右轉180度
370	轉盤左轉90度
380	轉盤左轉180度
500+A	等待 A 秒 ; $0 < A < 100$
6000+B	前進 B(mm) 抵達充電站 ; $0 < B < 1000$
7000+C	前進 C(mm) 離開充電站 ; $0 < C < 1000$
10000+D*100	直行 D 公尺 ; $0 < D < 50$

Web Service Interface

AGV名稱	
ITRI_3-1	教室內
ITRI_3-2	教室外
ITRI_3-3	教室內

Web Service Interface

結果碼	代碼詳細敘述
0	正常。
900	錯誤代碼；AGV忙碌中。
902	錯誤代碼；AGV名稱不存在。
905-908	錯誤代碼；AGV連線異常。
909	錯誤代碼；無效參數。
910	錯誤代碼；JSON格式錯誤。
911	錯誤代碼；AGV命令代碼錯誤。
912	錯誤代碼；AGV參數錯誤。
921	錯誤代碼；Web Service異常。

TEST

- 取得ITRI_3-1狀態資訊，並將狀態內容實作。
- 補充說明：狀態內容投影片並未說明，請根據取得JSON內容，進行物件建立(object)，並將JSON內容進行反序列化。