

RSAR

Leo Shiang · 2019/02/14

簡介

RSAR 是一個用正規表示式作文字搜尋的工具，特點是輸出的樣式很彈性。

使用方式

```
C:> RSAR Sample.json
```

使用範例

定義正規表示式

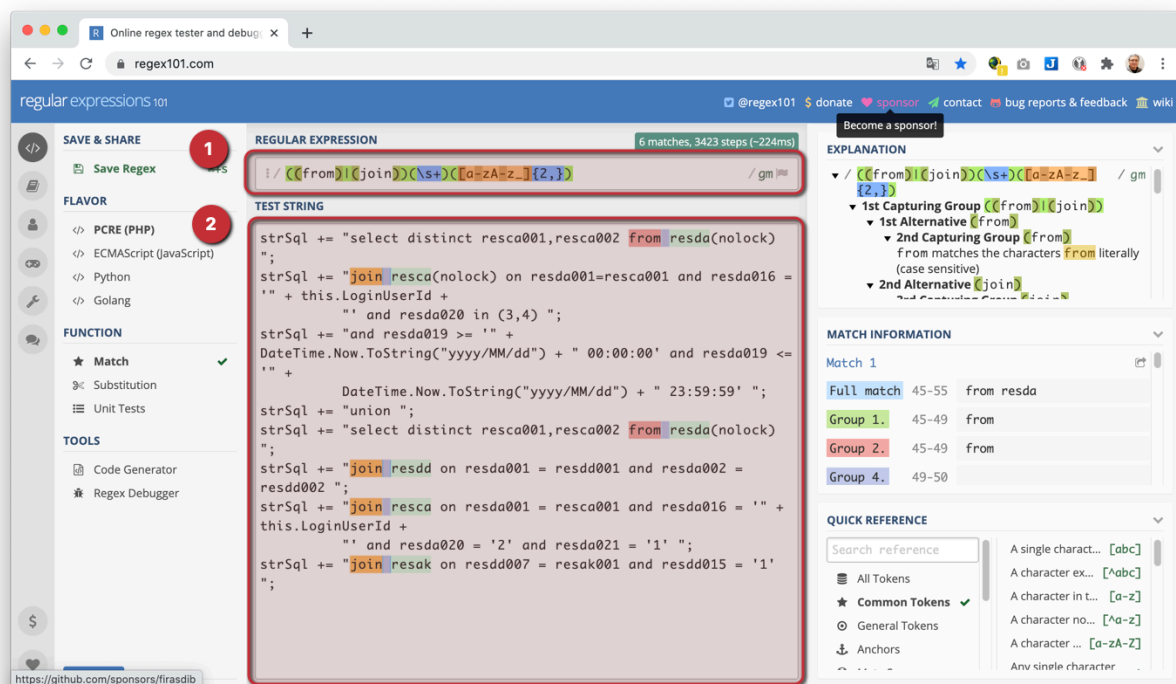
如果要在數百個 C# 程式裡面的 SQL 找出 SELECT 和 JOIN 的 TABLE 名稱，將會是一件很累人的事情，此時我們可以透過 RSAR 的搜尋功能很快的找出來。首先，看一下我們要找的東西長什麼樣子，以下是一段 C# 程式：

```
1  strSql += "select distinct resca001,resca002 from resda(nolock) ";
2  strSql += "join resca(nolock) on resda001=resca001 and resda016 = '" +
    this.LoginUserId +
3      "' and resda020 in (3,4) ";
4  strSql += "and resda019 >= '" + DateTime.Now.ToString("yyyy/MM/dd") + " 00:00:00'
    and resda019 <= '" +
5      DateTime.Now.ToString("yyyy/MM/dd") + " 23:59:59' ";
6  strSql += "union ";
7  strSql += "select distinct resca001,resca002 from resda(nolock) ";
8  strSql += "join resdd on resda001 = resdd001 and resda002 = resdd002 ";
9  strSql += "join resca on resda001 = resca001 and resda016 = '" + this.LoginUserId
    +
10     "' and resda020 = '2' and resda021 = '1' ";
11 strSql += "join resak on resdd007 = resak001 and resdd015 = '1' ";
```

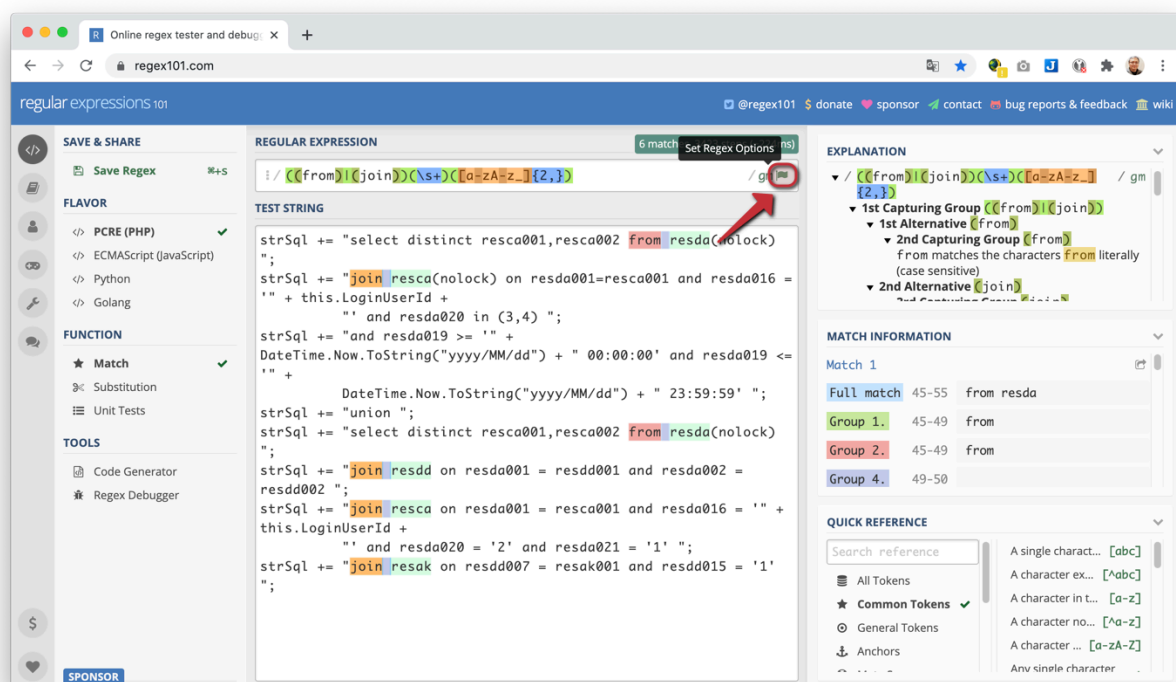
我們要找 `from resda` , `join resak` 之類的文字，因此可以定義出以下的正規表示式

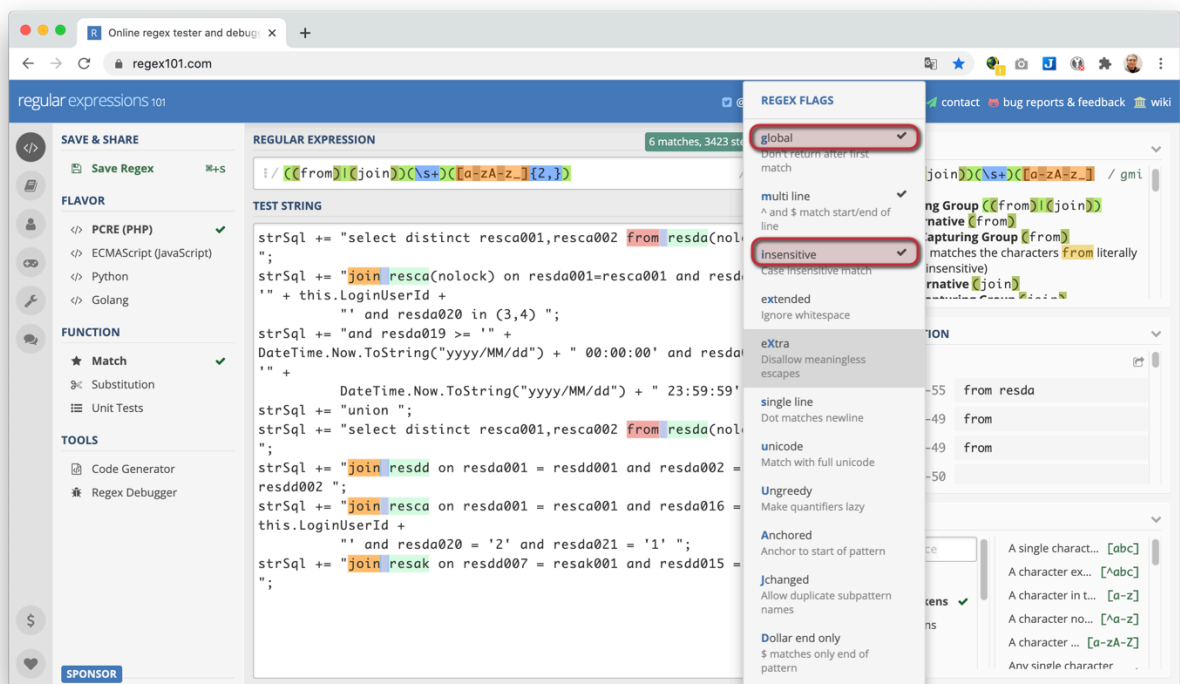
```
1  ((from)|(join))(\s+)([a-zA-z_]{2,})
```

為方便測試，可以開啟 <https://regex101.com>，將上面的正規表示式貼到 REGULAR EXPRESS 區域，將要找的資料貼到下方的 TEST STRING

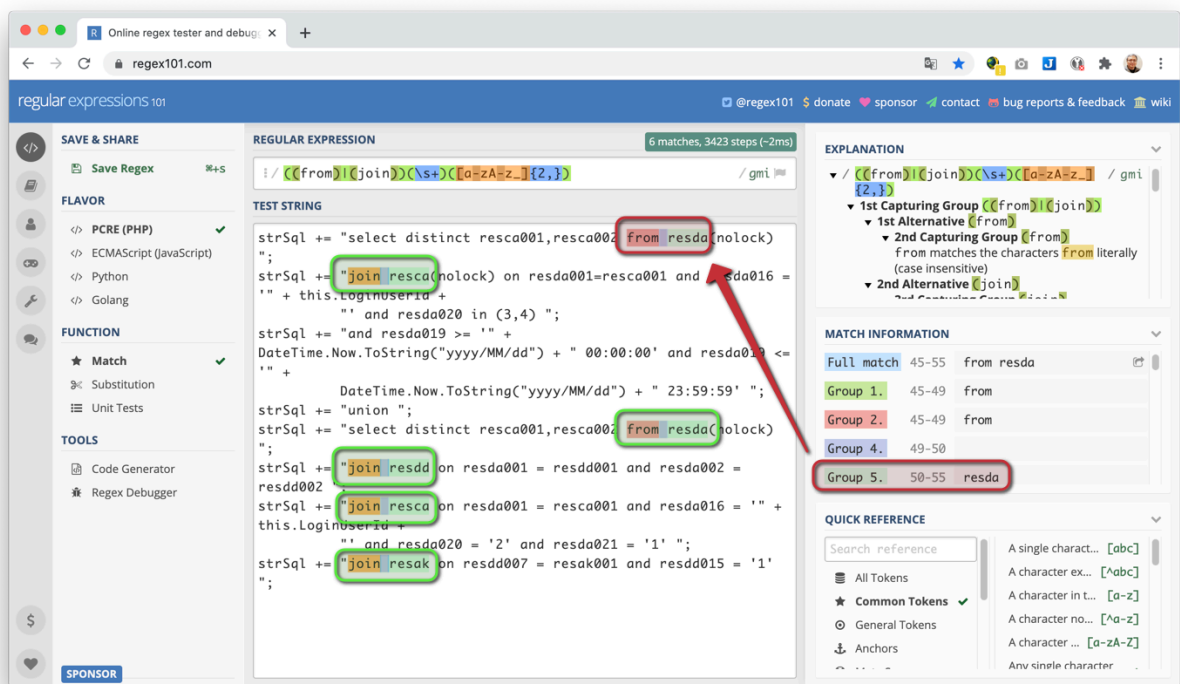


在 REGULAR EXPRESS 輸入區的尾端有一個旗標圖示，點下去，勾選 global 和 insensitive





符合條件的文字會用顏色標示出來。



撰寫設定檔

RSAR 的設定檔為 JSON 格式，我們將上面測試過的正規表示式寫到 Search 的 Regex 區段。

注意：原本一個斜線要變成兩個斜線

RootDirectory 填上程式檔案的路徑，**IgnoreCase** 設為 true

我們在 regex101.com 可以看到 table name 的群組是在第五個，將**Output** 的 **MatchedGroups** 設定為 5，就會只顯示 table name 的文字：

```
1  {
2    "Search": {
3      "RootDirectory": "D:\\Project",
4      "ExcludedExtensions": [],
5      "ExcludedDirectories": [],
6      "IncludedExtensions": [ ".cs" ],
7      "IncludedDirectories": [],
8      "Regex": "((from)|(join))(\\s+)([a-zA-z_]{2,})",
9      "IgnoreCase": true
10   },
11   "Output": {
12     "FileName": false,
13     "MatchedCount": false,
14     "MatchedText": true,
15     "MatchedGroups": [ 5 ],
16     "TextTransform": "U"
17   }
18 }
```

將結果寫入檔案

我們只要在指令的尾端加上 > 檔案名稱即可

```
C:> RSAR Sample.json > TableName.txt
```

設定檔說明

完整的設定檔內容如下

```
1  {
2    "Search": {
3      "RootDirectory": "D:\\Project",
4      "ExcludedExtensions": [],
5      "ExcludedDirectories": [],
6      "IncludedExtensions": [ ".cs" ],
7      "IncludedDirectories": [],
8      "Regex": "((from)|(join))(\\s+)([a-zA-z_]{2,})",
9      "IgnoreCase": true
10   },
11   "Output": {
12     "FileName": false,
13     "MatchedCount": false,
14     "MatchedText": true,
```

```
15     "MatchedGroups": [ 5 ],
16     "TextTransform": "U"
17   }
18 }
```

搜尋選項 Search

RootDirectory

要搜尋的檔案目錄

ExcludedExtensions

要排除的檔案副檔名，例如：要排除 .bat, .dat 就設定 [".bat", ".dat"]

ExcludedDirectories

要排除的目錄，例如：["D:\Projects\test", "D:\Projects\data"]

IncludedExtensions

如果只想搜尋 .bat, .dat 就設定 [".bat", ".dat"]

Regex

正規表示式。注意：原本一個斜線要變成兩個斜線

輸出選項 Output

FileName

如果設定為 true，處理每一個檔案時，會先列出檔案完整路徑名稱，接下來才是符合的文字

```
D:\Projects\USER_PWD_CHG.aspx.cs
USERS
USERS
GROUPS
D:\Projects\BaseData.cs
SERVICE_LOCATION
D:\Projects\CodingFunction.cs
WEB_SERVER_LIST
```

MatchCount

顯示檔案符合搜尋條件的文字數目

```
D:\Projects\USER_PWD_CHG.aspx.cs
3
USERS
USERS
GROUPS
D:\Projects\BaseData.cs
1
SERVICE_LOCATION
D:\Projects\CodingFunction.cs
1
WEB_SERVER_LIST
```

MatchedText

顯示檔案裏面符合的文字，如果設成 false 範例的輸出將會變成

```
D:\Projects\USER_PWD_CHG.aspx.cs
3
D:\Projects\BaseData.cs
1
D:\Projects\CodingFunction.cs
1
```

MatchedGroups

如果正規表示式有群組，可以在此屬性設定只要顯示哪幾個群組，以前面的例子，table name 會在第五個群組，因此 MatchedGroups 要設成 [5]

TextTransform

輸出要轉成大寫，請設定為 "U"

輸出要轉成小寫，請設定為 "L"

如果不改變大小寫，請設定為空字串 ""