

Ticket System API

1、 登入驗證

如果想要取得ticket列表，亦或是增修查改，都必須先登入系統取得token進行驗證。

- ```
1. curl -H "Content-Type: application/json" -X POST -d '{"user":"bob","passwd":"123"}'
https://TicketSystem.com.tw/Login
```

登入成功後會回傳JWT Token。

```
1. {
2. "token": "RAHORNARNANRNAJFNAINFJAENIOUNAIUNGIANIUENAUTBAUIENTANIUTNUAIENAITNEIABITEBNA
ITBEIABTIAB131564465161654ETANTOUIAENUITONAE"
3. }
```

## 2、透過JWT取得資料

透過將標頭加入Token來驗證取得資料

```
1. curl -H "Authorization: Bearer {Token}" https://TicketSystem.com.tw/Search
```

資料結果範例:{

```
4. {
5. "OrderBy":Asc,
6. "SortBy":ticketId
7. "Pagenation":10,
8. "Page":1
9. "tickets":[{
10. "ticketId":1,
11. "Summary":"Bug1",
12. "Description ":"Bug1",
13. "type":"bug",
14. "Severity":1,
15. "Priority ":1,
16. "Resolve":false
17. },
18. {
19. "ticketId":2,
20. "Summary":"TC 1",
21. "Description ":"TC 1",
22. "type":"Test Case",
23. "Severity":2,
24. "Priority ".2,
25. "Resolve":true
26. }]
27. }
```

而如果當token過期或是不符合：

```
1. {
2. "timestamp": "2020-02-29T20:53:35.761+0000",
3. "status": 403,
4. "error": "Forbidden",
5. "message": "Access Denied",
6. "path": "/login"
7. }
```

而當token快過期時，可以透過目前的token去取回新的token：

```
2. curl -H "Authorization: Bearer {Token}" https://TicketSystem.com.tw/Refresh
```

取回的範例：

```
28. {
29. "token": "FIOJGOIJOANGUINANIENTIUEAIUTNAENNAFNEAUINGIUANTINAEIUNTIUAENIUENAITA54984A1E51T
8A984ET984ATAE489489"
30. }
```

也可以透過在Pagination、Page、OrderBy、SortBy來使用排序與分頁

```
3. curl -H "Authorization: Bearer {Token}"
https://TicketSystem.com.tw/Search?Pagenation=10&OrderBy=Asc&SortBy=ticketId
```

而也可以透過Token, 取得使用者的資訊

```
4. curl -H "Authorization: Bearer {Token}" https://TicketSystem.com.tw/User
```

取回的範例結果:

```
8. {
9. "username": "QA1",
10. "status": 1
11. }
```

### 3、 對資料進行增修查改

透過Token的認證，來對資料進行增修查改，而沒有權限的就無法執行。

沒有權限範例：

```
12. {
13. "timestamp": "2020-02-29T20:53:35.761+0000",
14. "status": 403,
15. "error": "Forbidden",
16. "message": "Access Denied",
17. "path": "/login"
18. }
```

使用QA使用者的Token登入，來創建ticket。

### Post 指令

```
5. curl -H "Authorization: Bearer {Token}" -X POST -d
'{"Summary":"Test1","Description":"Test1","Serveriy":5,"Priority":4 , TicketTypeId:1}'
https://TicketSystem.com.tw/Create
```

而當成功時會回傳成功與創建時間：

```
19. {
20. "createtime": "2021-11-29 11:00:00",
21. "status": success,
22. }
```

而要更新或是刪除則是如下範例：

#### 更新

```
6. curl -H "Authorization: Bearer {Token}" -X POST -d
'{"Summary":"Test11","Description":"Test11","Serveriy":5,"Priority":4 , TicketTypeId:1}'
https://TicketSystem.com.tw/Update/1
```

#### 刪除

```
7. curl -H "Authorization: Bearer {Token}" https://TicketSystem.com.tw/Delete/1
```

而同樣的，我們也可以透過PM使用者的Token創建Feature Request Ticket:

```
8. curl -H "Authorization: Bearer {Token}" -X POST -d
'{"Summary":"Test2","Description":"Test2","Serveriy":7,"Priority":7 , TicketTypeId:3}'
https://TicketSystem.com.tw/Create
```

而當成功時一樣會回傳成功與創建時間:

```
23. {
24. "createtime": "2021-11-29 11:00:00",
25. "status": success,
26. }
```

而要更新或是刪除則是如下範例:

更新

```
9. curl -H "Authorization: Bearer {Token}" -X POST -d
'{"Summary":"Test11","Description":"Test11","Serveriy":5,"Priority":4 , TicketTypeId:1}'
https://TicketSystem.com.tw/Update/1
```

刪除

```
10. curl -H "Authorization: Bearer {Token}" https://TicketSystem.com.tw/Delete/1
```

相同的RD和QA也可以透過該方式，將某個Ticket標註成已解決。

```
11. curl -H "Authorization: Bearer {Token}" https://TicketSystem.com.tw/Resolve/1
```

這樣我們就可以將TicketId為1的資料標註成已解決：

```
27. {
28. "resolvetime": "2021-11-29 11:00:00",
29. "status": success,
30. }
```

#### 4、透過Admin權限對使用者進行管理

```
5. curl -H "Authorization: Bearer {Token}" -X POST -d
'{"UserName":"QA2","Status":1,"Passwd":"AF12354DA4F7A489D4A98F4A8489
DA4F89A4D98AS41AD5616A1AS1DF19","Salt":"AFA7984A4FA6G1A487G4A6E
G41A4G8AE4G41AEE84GA694G89EA"}'
https://TicketSystem.com.tw/User/Create
```

當成功時會回傳成功與創建時間:

```
31. {
32. "createtime": "2021-11-29 11:00:00",
33. "status": success,
34. }
```

而要更新或是刪除則是如下範例:

更新

```
12. curl -H "Authorization: Bearer {Token}" -X POST -d
'{"UserName":"QA2","Status":1,"Passwd":"AF12354DA4F7A489D4A98F4A8489DA4F89
A4D98AS41AD5616A1AS1DF19","Salt":"AFA7984A4FA6G1A487G4A6EG41A4G8AE4G
41AEE84GA694G89EA"}'
https://TicketSystem.com.tw/User/Update/1
```

刪除

```
13. curl -H "Authorization: Bearer {Token}" https://TicketSystem.com.tw/User/Delete/1
```

## 五、URL與欄位說明

登入：

2. <https://TicketSystem.com.tw/Login>

| 欄位       | 說明    |
|----------|-------|
| username | 使用者帳號 |
| passwd   | 密碼    |

## 取得Ticket資訊:

```
14. curl -H "Authorization: Bearer {Token}" https://TicketSystem.com.tw/Search
```

| 欄位          | 說明          |
|-------------|-------------|
| OrderBy     | 升序或降序       |
| SortBy      | 要升序或降序判斷的欄位 |
| Pagenation  | 一頁回傳多少資料    |
| Page        | 第幾頁         |
| ticketId    | Ticket的編號   |
| Summary     | 主旨          |
| Description | 描述          |
| type        | 是哪種ticket   |
| Severity    | 嚴重程度        |
| Priority    | 優先程度        |
| Resolve     | 是否已被解決      |

## 取得使用者資訊

```
15. curl -H "Authorization: Bearer {Token}" https://TicketSystem.com.tw/User
```

| 欄位       | 說明    |
|----------|-------|
| username | 使用者帳號 |
| status   | 權限代碼  |

| 權限代碼 | 對應    |
|------|-------|
| 1    | QA    |
| 2    | RD    |
| 3    | PM    |
| 4    | ADMIN |

## 新增、修改ticket

```
16. curl -H "Authorization: Bearer {Token}" -X POST -d
'{"Summary":"Test1","Description":"Test1","Serveriy":5,"Priority":4 , TicketTypeId:1}'
https://TicketSystem.com.tw/Create
```

| 欄位           | 說明        |
|--------------|-----------|
| Summary      | 主旨        |
| Description  | 描述        |
| TicketTypeId | 是哪種ticket |
| Severity     | 嚴重程度      |
| Priority     | 優先程度      |

## 新增、更新使用者資訊

```
17. curl -H "Authorization: Bearer {Token}" -X POST -d
'{"UserName":"QA2","Status":1,"passwd":"AF12354DA4F7A489D4A98F4A8489DA4F89
A4D98AS41AD5616A1AS1DF19","Salt":"AFA7984A4FA6G1A487G4A6EG41A4G8AE4G
41AEE84GA694G89EA"}' https://TicketSystem.com.tw/User/Update/1
```

| 欄位       | 說明             |
|----------|----------------|
| UserName | 使用者名稱          |
| Status   | 權限代碼           |
| Passwd   | 密碼             |
| Salt     | 隨機生成的字串，用來加密密碼 |