

APIGW- 利用 JWT 實作 APIGW Authorizer

July 30, 2019

Version 1.0

Index

Lambda Functions	3
Authorizer	3
Login	4
Test	4
API Gateway	4
Create API Gateway	4
Deploy API	9
測試流程	11
Login	11
Invoke Test Function	12

Lambda Functions

兩個 Lambda functions Runtime 都是 Node.js, 並透過 jsonwebtoken 這個 module 來完成 JWT 的實作

簽發跟認證 jwt 的過程都會有一個設定好的 environemnt variable (JWT_SECRET)

在 sample code 中, 唯一能夠登入的 username/password 為 test/1234

並且 JWT 的有效期限都設為 5m

Authorizer

1. 在 Lambda 上建立一個新 function

- Function Name: 輸入一個此 functoin 的名字, 例如 apigw_token_autherizer
- Runtime 選擇 Node.js 10.x

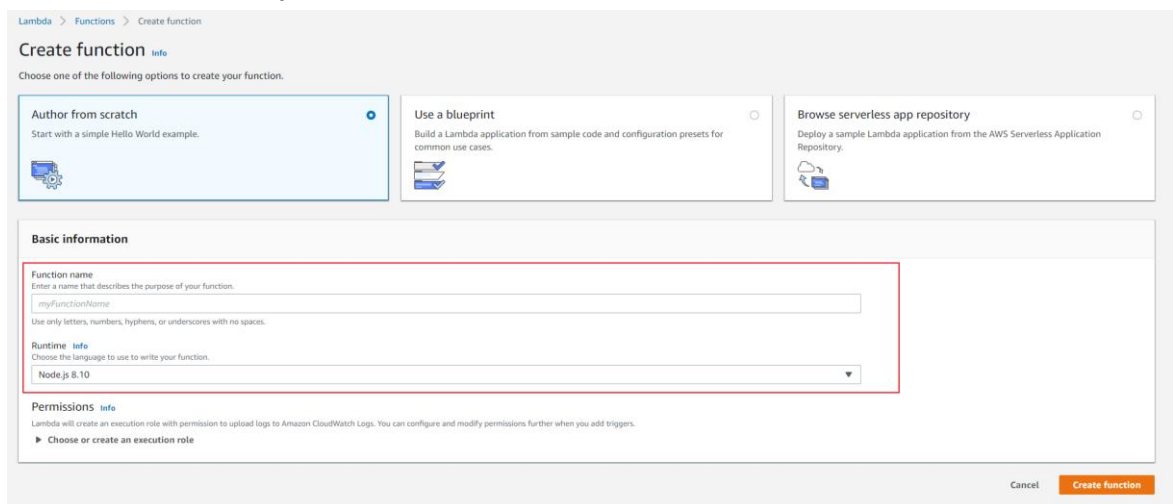


Figure 1

2. 在本地端透過 CLI 把 apigw_token_autherizer.zip 上傳到 lambda

```
aws lambda update-function-code --function-name <lambda-function-name> --zip-file  
fileb://apigw_token_autherizer.zip
```

3. 在 Lambda 裡加入以下 Environment variables

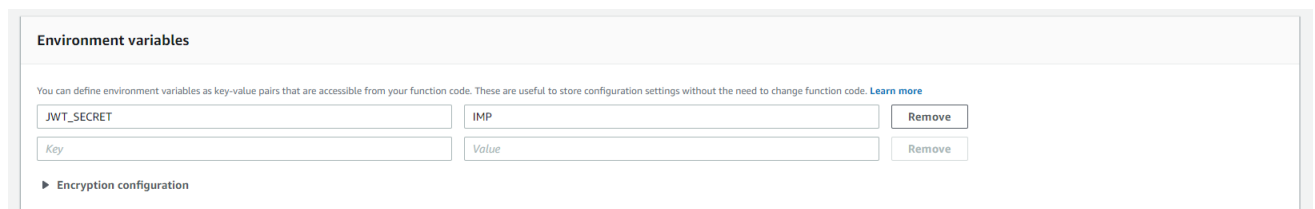


Figure 2

Login

參考前面步驟 upload 另一個 Lambda function 來處理 Login Request

1. 在 Lambda 上建立一個新 function
 - Function Name: 輸入一個此 function 的名字, 例如 jwt_login-test
 - Runtime 選擇 Node.js 10.x
2. 在本地端透過 CLI 把 jwt_login_test.zip 上傳到 lambda

```
aws lambda update-function-code --function-name <lambda-function-name> --zip-file  
fileb://jwt_login_test.zip
```

3. 在 Lambda 裡加入 Figure 2 的 Environment variables

Test

最後再建立一個 test function, 此 function 之後會接在 API Gateway 後面, 並且需要經過 authorizer 驗證後才會執行, 直接用 Default 的 code 就可以完成測試

1. 在 Lambda 上建立一個新 function
 - Function Name: 輸入一個此 function 的名字, 例如 test
 - Runtime 選擇 Node.js 10.x
2. 直接新建完成就可以了

API Gateway

Create API Gateway

1. 在 AWS Console 進入 API Gateway 後點選 Create API (Figure 3)
或是 如果重來沒建立過 API Gateway, 直接點選 Get Started (Figure 4)

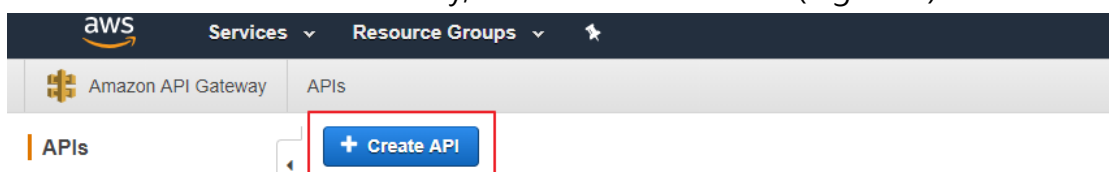


Figure 3



Figure 4

2. Protocol 選擇 Rest (Figure 5.1)

Create New API 下選擇 New API (Figure 5.2)

API name 輸入此 API 的名字, 例如: tokentest (Figure 5.3)

都設定好了後, 點選 Create API

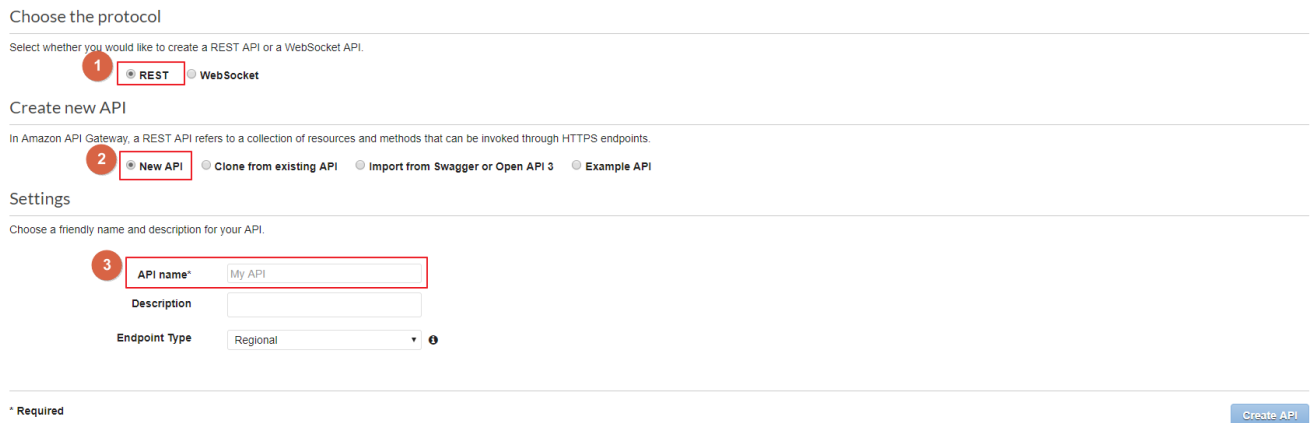


Figure 5

3. API 建立好了後, 要 create Resource, 點選 Actions->Create Resource (Figure 6), 建立如 Figure 7 的兩個 Resources

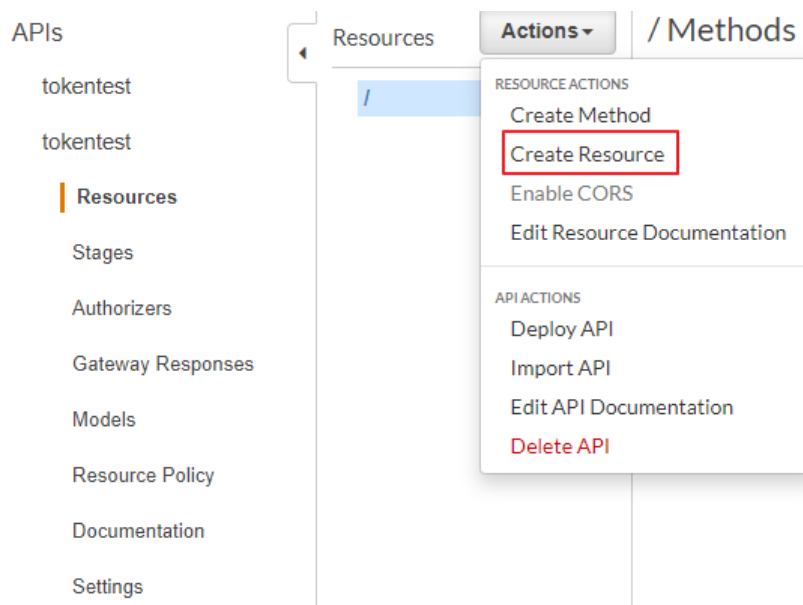


Figure 6

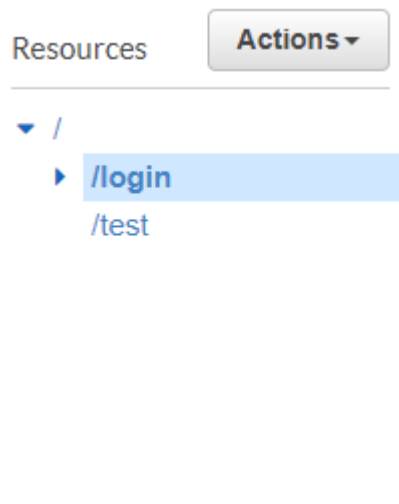


Figure 7

4. 在 login 這個 resource 底下建立一個 post method 並把他接上之前建立的 Lambda Login Function (Figure 9)

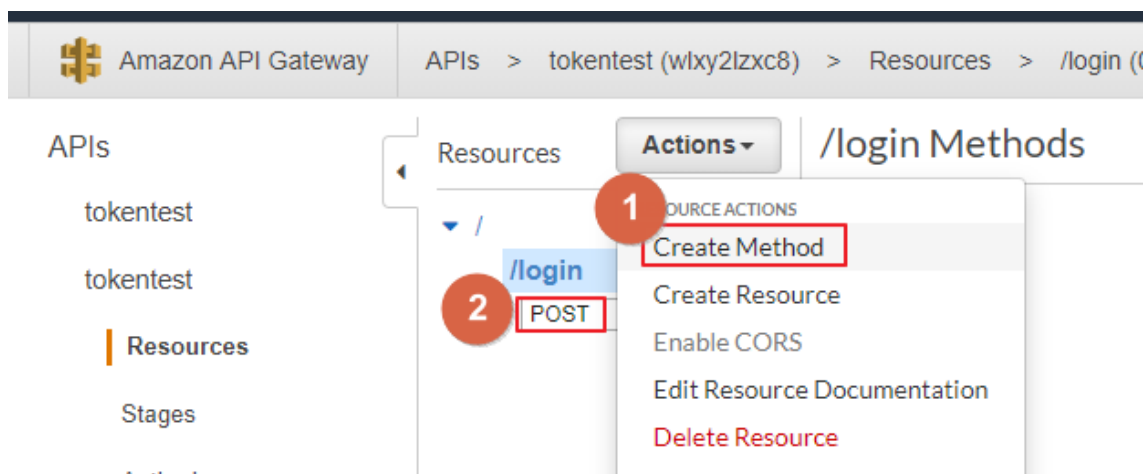


Figure 8



Figure 9

5. 在 test 這個 resource 底下建立一個 get method (Figure 10)並把他接上之前建立的 Lambda test function (Figure 11)



Figure 10



Figure 11

6. 在 Authorizers 裡建立一個 Authorizer (Figure 12.1-12.2), 並接上之前建立的 Lambda Authorizer Function (Figure 12.4)

Note: Token Source (Figure 12.5)這裡的設定會決定之後 Authorizer 會從 Request 的哪個 header 抓取 Token

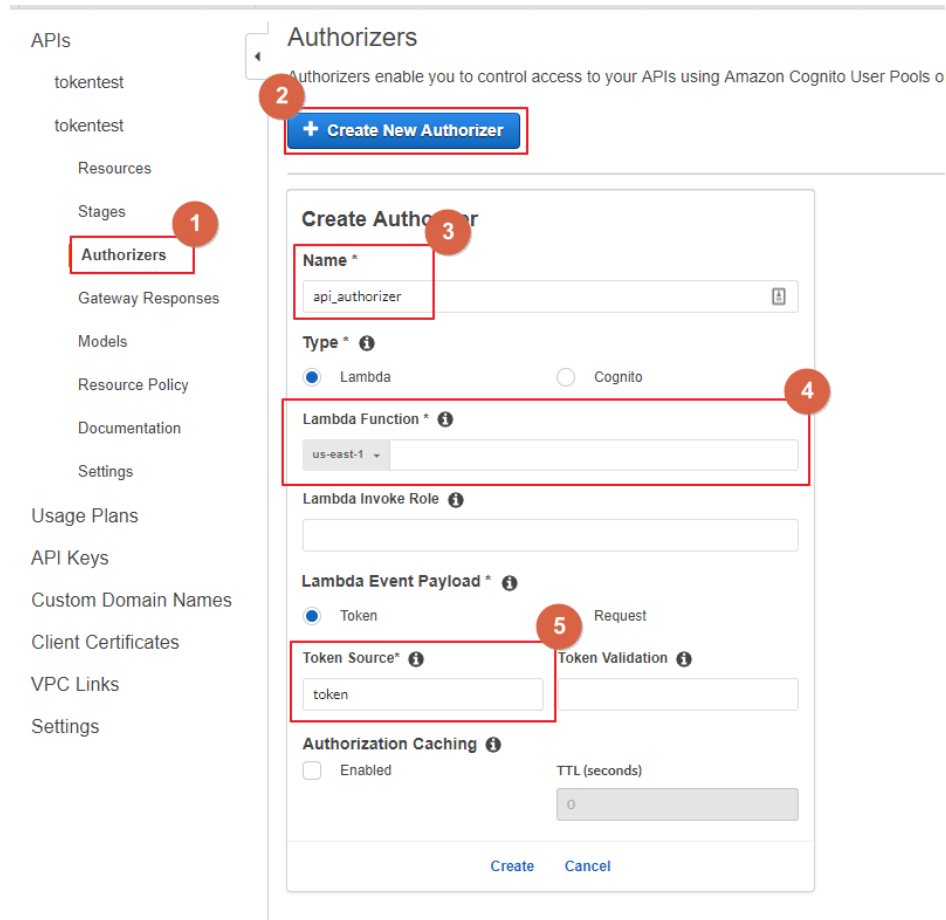


Figure 12

7. 點選/test 的 Get Method, 然後點選 Method Request (Figure 13)

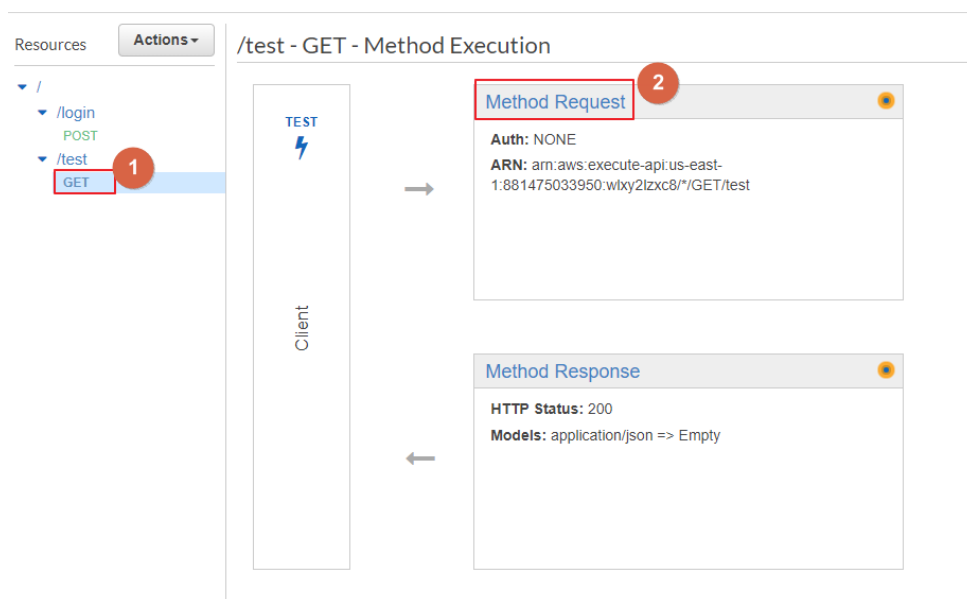


Figure 13

8. 在 Authorization 的設定裡, 選擇剛剛建立的 Authorizer (Figure 14)

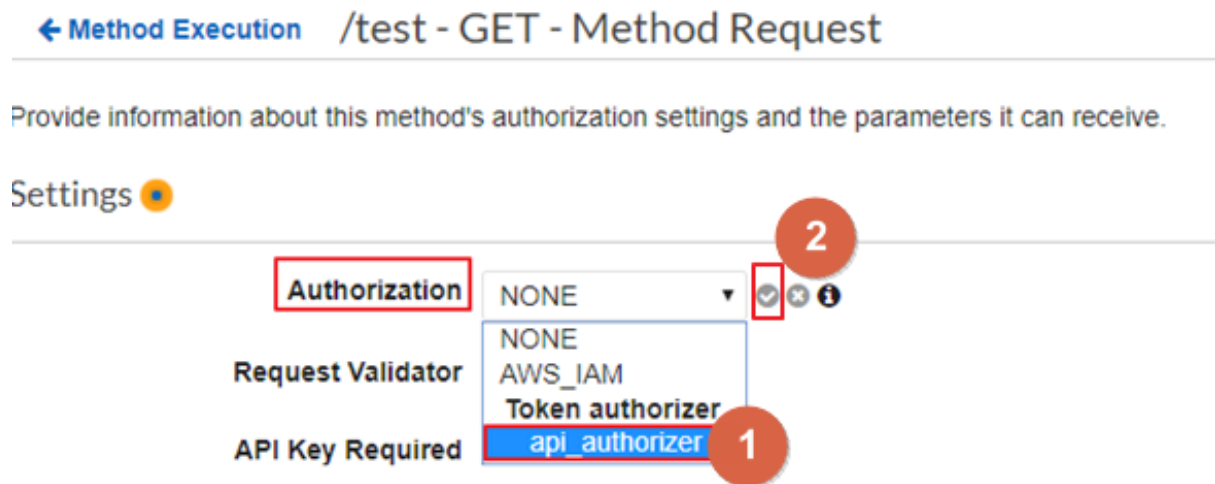


Figure 14

Deploy API

1. 在 Resources 介面下, 選擇 Actions, 點選 Deploy API (Figure 15)

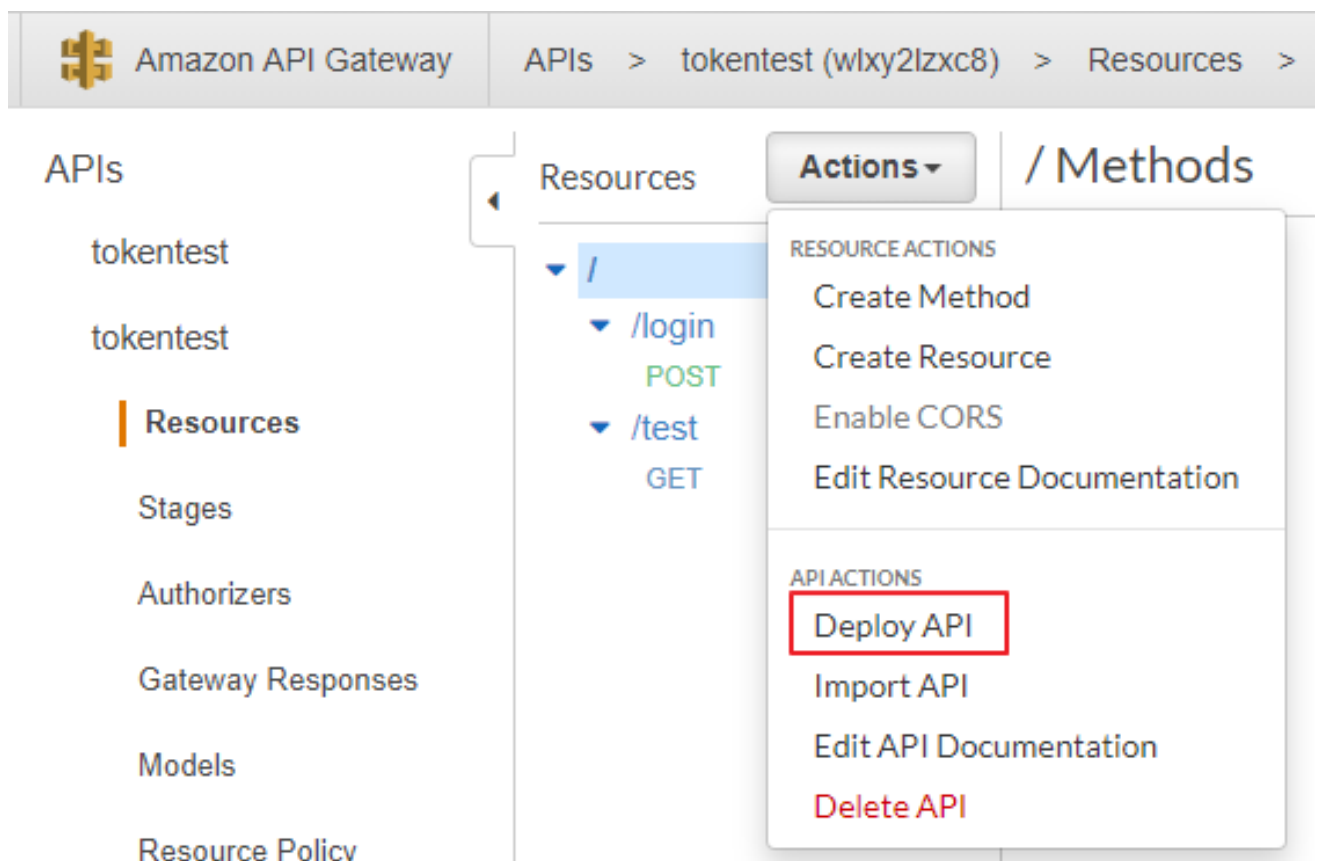
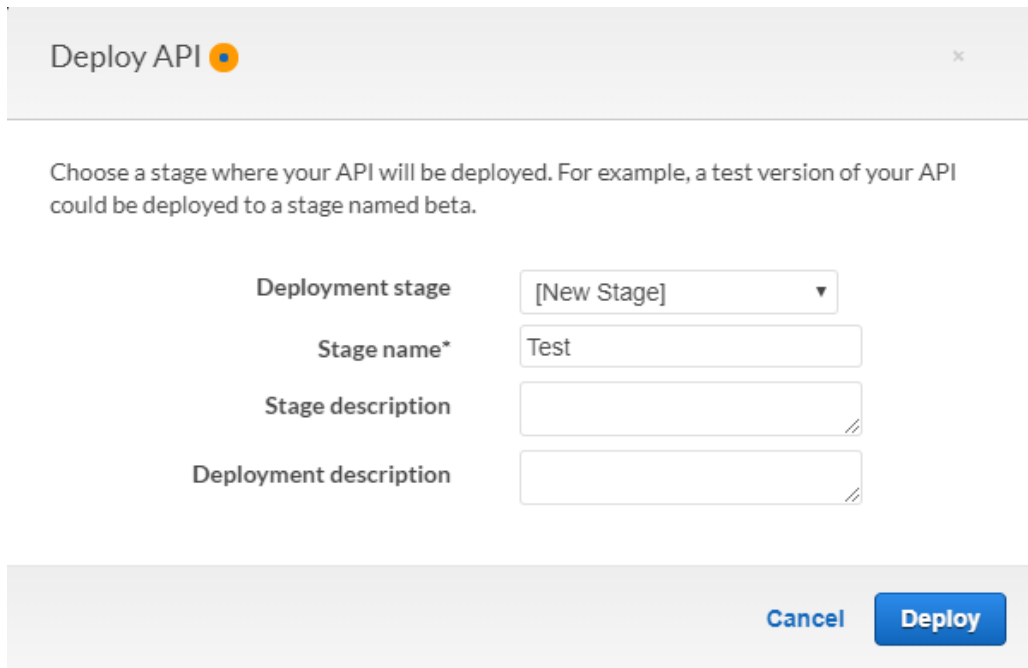


Figure 15

2. Deploy API 到一個 New Stage 並設定 Stage 的名字 (Figure 16)



Deploy API

Choose a stage where your API will be deployed. For example, a test version of your API could be deployed to a stage named beta.

Deployment stage: [New Stage]

Stage name*: Test

Stage description:

Deployment description:

Cancel Deploy

Figure 16

3. Deploy 完成後, 在 Stages 裡可以看到剛剛 Deploy 的 API, 並會有一串 Invoke URL (Figure 17)

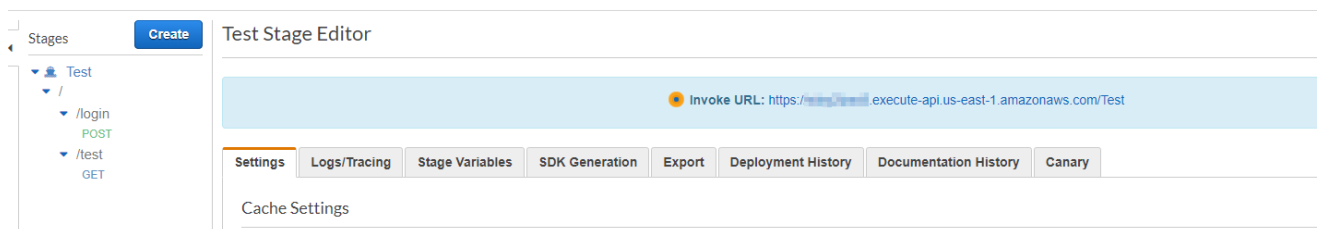


Figure 17

測試流程

以下測試使用 Postman

Login

1. 在 Postman 裡對 <Invoke-URL>/Test/login 做 Post Request, 並在 Request Body 裡輸入以下內容來取得 Token (Figure 18)

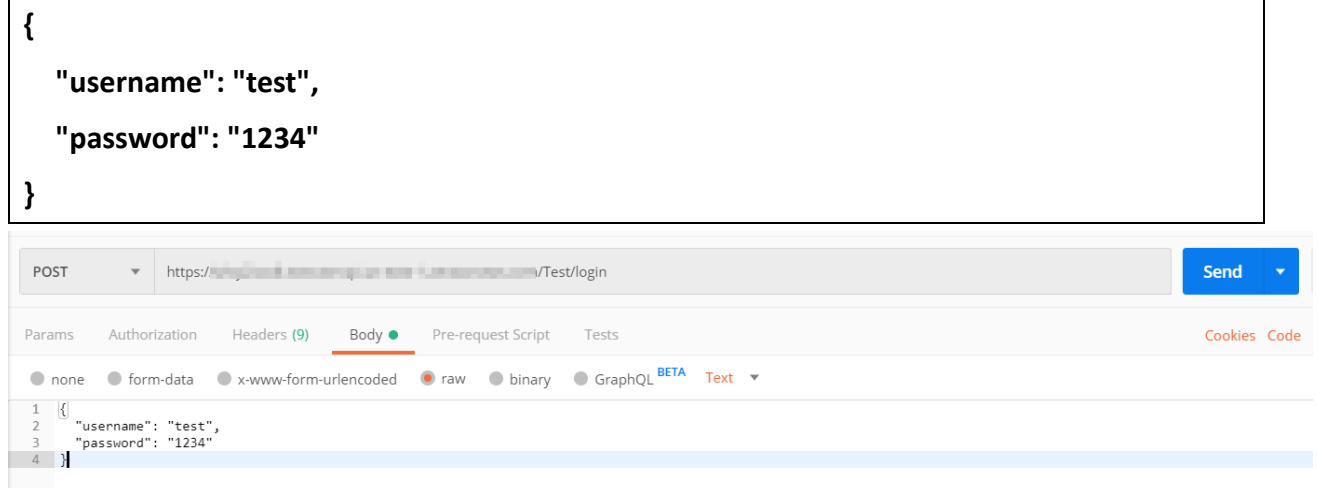


Figure 18

2. 點擊 Send 送出後, 會看到以下 Response (Figure 19), 把 token 複製下來

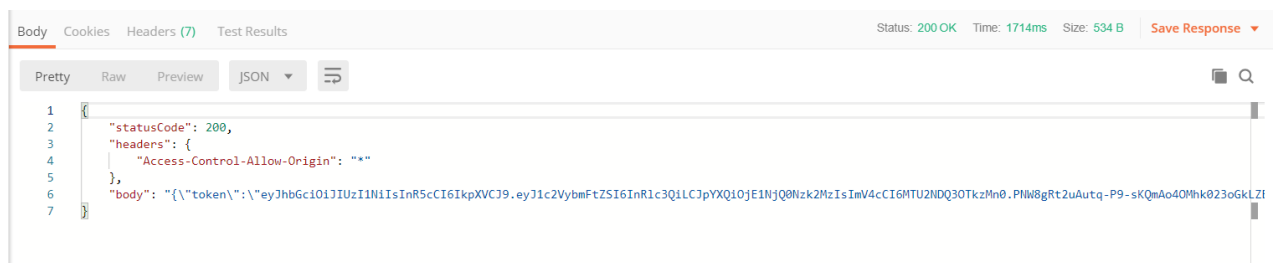


Figure 19

3. 把這個 Token 拿去 <https://jwt.io/> 解密會得到以下資訊, 可以看到其中包含了設定的 username 跟 Expired time (Figure 20)



Figure 20

Invoke Test Function

1. 拿著這個 Token (Figure 21.1)去對 <Invoke-URL>/Test/test 做 Get Request 才能夠成功拿到正確的 Response (Figure 21.2)

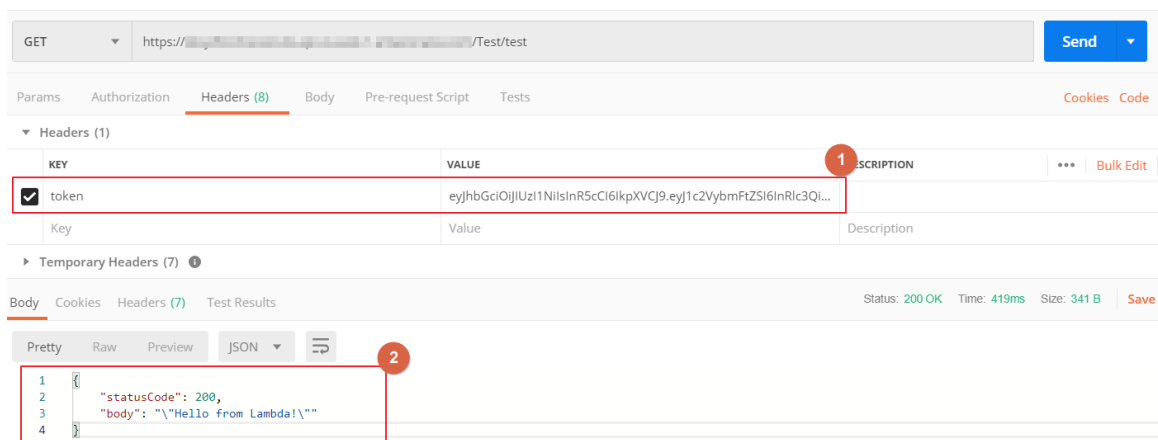


Figure 21

2. 若是輸入錯誤 Token, 會得到以下錯誤

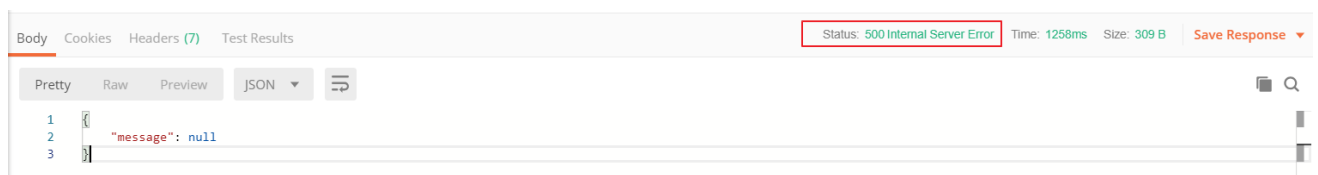


Figure 22