

ST2-Week1-HW

記事本： 第一個記事本

已建立： 2018/3/25 下午 12:25

已更新： 2018/3/26 下午 11:06

作者： a6513512@gmail.com

URL： <http://www.runoob.com/nodejs/nodejs-express-framework.html>

server.js

```
const express = require('express');
const app = express(); // build an Express server

app.use(express.static('public'));

// set the view engine to ejs
app.set('view engine', 'ejs'); // must install ejs first: npm install ejs --save

app.get('/about', function(req, res){
  var user = {name: req.query.name};
  res.render('index', {user: user});
});

app.listen(3000, function(){
  console.log('Example app is running on port 3000 now');
})
```

如何加上CSS,JS等靜態文件？

首先先修改server.js

```
// 使用 express.static 來設置靜態文件路徑(CSS,JS) , 將css,js等資料夾放到 /public下 , index.ejs裡的路徑改成
app.use(express.static('public'));
```

接著在index.ejs的相關路徑作調整，不需要加上/public，也不用加上/

```
<link rel="stylesheet" type="text/css" href="css/grayscale.min.css" />
```

Reference : <https://code.i-harness.com/en/q/11c42cf>

Git操作筆記：

基本指令

透過 gitshell 建立 Repository

```
$ git init
$ git add README
$ git status
$ git commit -m "First Commit"
```

常用指令

```
$ git status # 查看發生改動的文件列表
$ git checkout . # 取消所有修改
$ git checkout xxxFile # 取消某個文件的修改
$ git add . # 新增所有修改
$ git commit -m 'commit message' # 提交修改
$ git push # 提交程式碼到遠程分支, 預設使用 origin 作為名稱
$ git pull # 從遠程倉庫獲取最新程式碼
$ git checkout branch-name # 切換到 xxx branch, -b 新建並切換到分支
$ git merge branch-name # 將 XXX 分支 merge 到當前分支
$ git help xxxCmd # 查看幫助文件, 會用瀏覽器打開 html 文件
$ git config --global user.name "User-Name" # Git 設定, --global 全局生效
```

多人共同編輯檔案時的常用指令

```
$ git checkout -b branch-name 新建並切換到分支
$ git branch # 查看本地分支, -r 查看遠程分支, -a 查看所有
$ git fetch # 從遠程獲取 branch/tag 訊息, 同步 branch 紀錄
$ git merge XXX # 將 xxx branch merge 到當前分支
$ git rebase
```

merge v.s rebase

合併前

merge 後 產生新的G點

rebase(整理)

後合併,D'or E'若有衝突必須先各自解決)

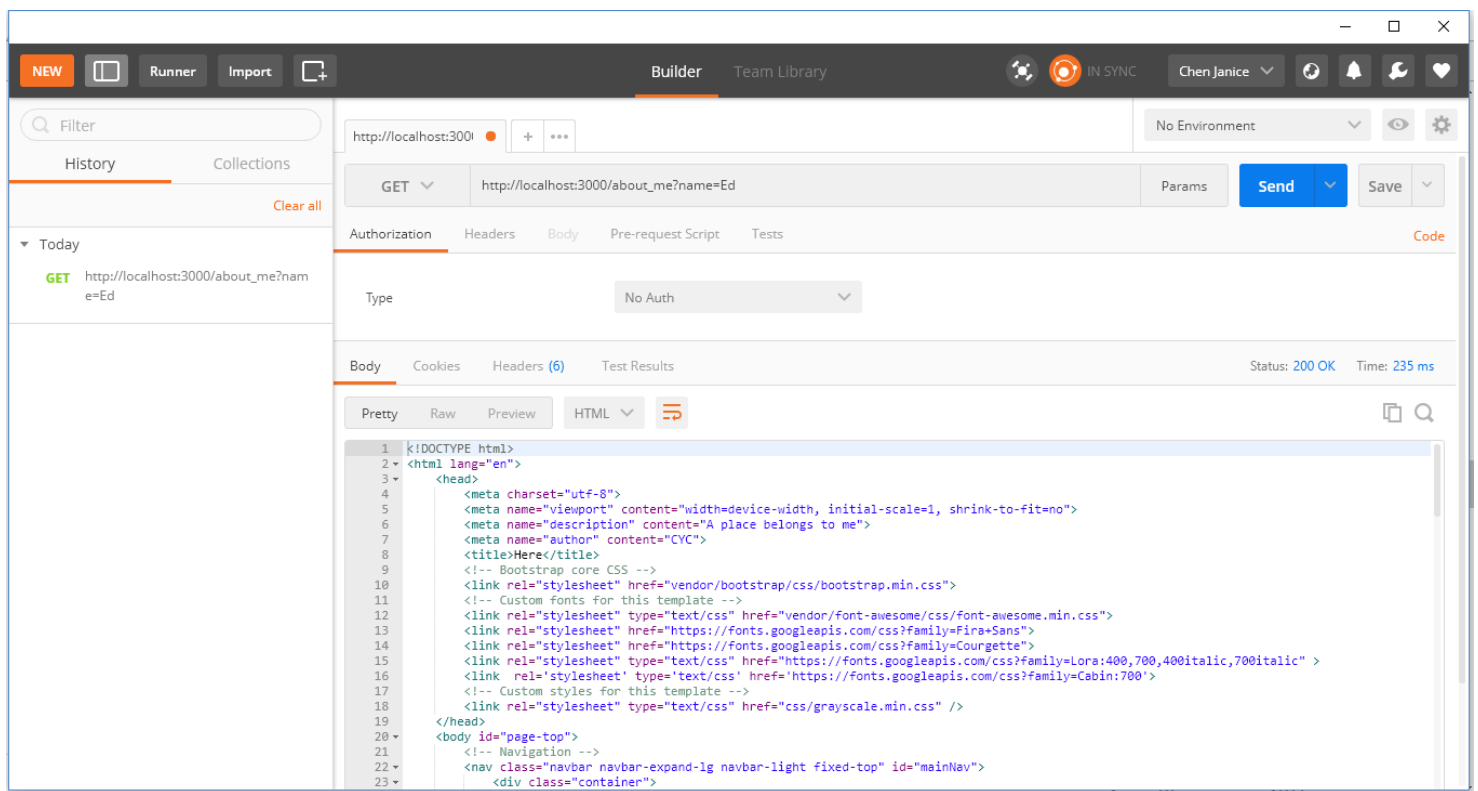
D---E feature
/
-F--D'--E' feature, master
A--B--C--F master

D-----E
/ \
A---B---C---F---G feature, master

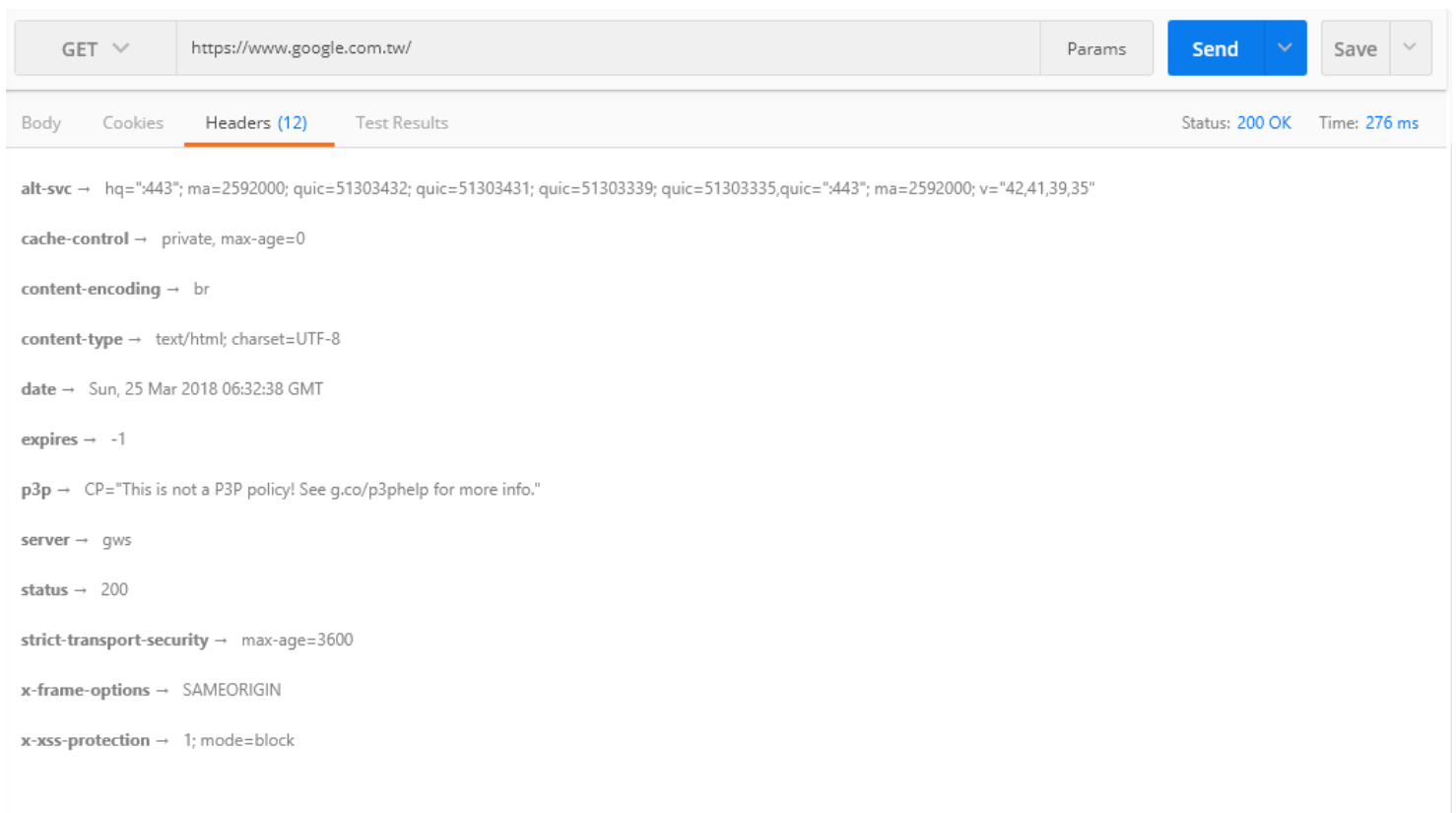
A--B--C-

POSTMAN測試

測試GET method回傳的body



對google網站測試GET method回傳的Header，從status 200可以看出來成功連線，content-type是html



對google網站測試POST method回傳的Header，從status 405可以看出google不允許

POST method

(也可以從回傳的header發現 allow→GET,HEAD)

POST

https://www.google.com.tw/

Params

Send

Save

Body

Cookies

Headers (9)

Test Results

Status: 405 Method Not AllowedTime: 258 ms

allow → GET, HEAD

alt-svc → hq=":443"; ma=2592000; quic=51303432; quic=51303431; quic=51303339; quic=51303335, quic=":443"; ma=2592000; v="42,41,39,35"

content-length → 1589

content-type → text/html; charset=UTF-8

date → Sun, 25 Mar 2018 06:33:12 GMT

server → gws

status → 405

x-frame-options → SAMEORIGIN

x-xss-protection → 1; mode=block