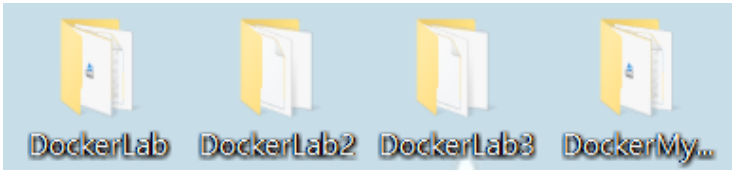
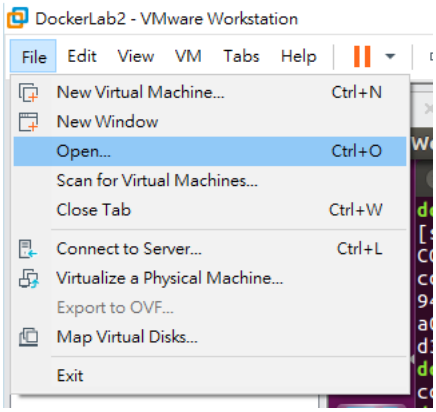
✓ Docker lab 網頁範例 source code:
https://drive.google.com/open?id=1TlJn28YBnGdn4GJJz_aTOHykvpnC6LhD

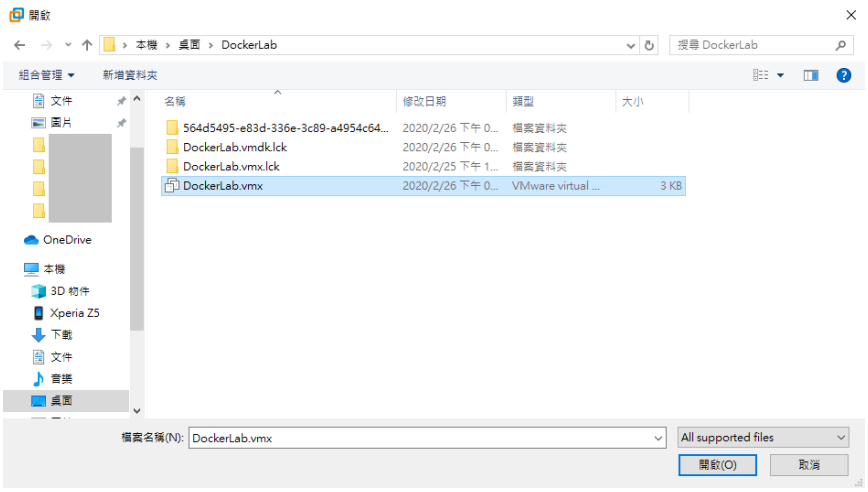
1.複製四個已完成的 DockerLab 資料夾，並為他們取名(ex. DockerLab、DockerLab2、 DockerLab3、DockerMySQL)。



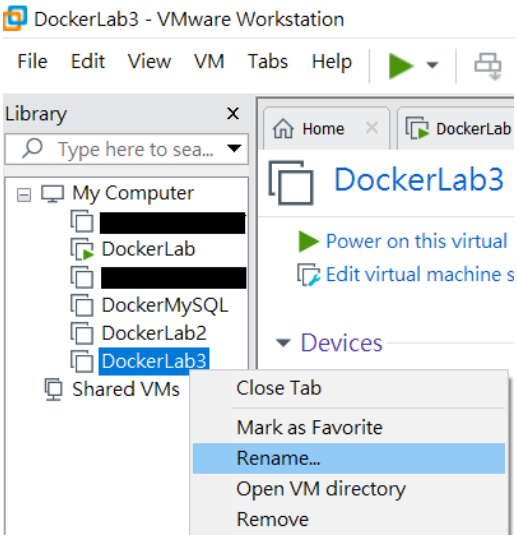
(重複 2-5)將四份都用 vmware 開啟
2.開啟 vmware，點選左上角 File->Open。



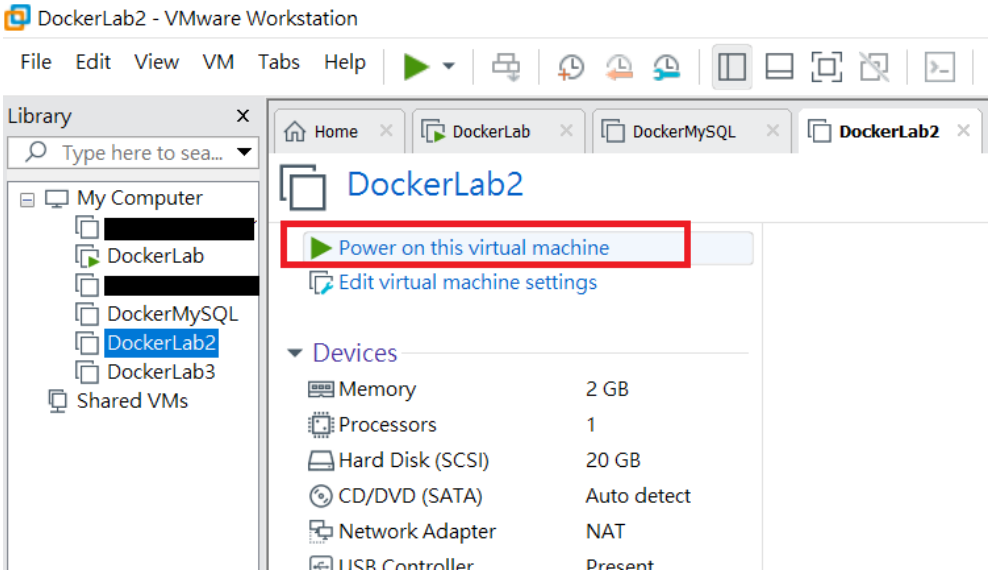
3.選取資料夾內的 vmx 檔後，開啟。



4.可右鍵後點選”Rename”，重新命名成自己易懂的名稱。



5.點選 Power on this virtual machine 開啟虛擬機。



6. DockerLab、DockerLab2、DockerLab3 開啟 tomcat。

```
dockerlab1221@ubuntu:~$ sudo docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS                          PORTS               NAMES
cde5c66e1076        aeea                "catalina.sh run"        7 days ago          Exited (143) 6 seconds ago                          tomcat
946e717ea76a        busybox             "nslookup google.com"    9 days ago          Exited (0) 9 days ago                               thirsty_wescoff
a09310afe6b4        busybox             "nslookup google.com"    9 days ago          Exited (0) 9 days ago                               sleepy_neumann
d3b25935f0a8        791                 "docker-entrypoint.s…"   9 days ago          Exited (0) 47 hours ago                             mysql
dockerlab1221@ubuntu:~$ sudo docker start cde5c66e1076
cde5c66e1076
```

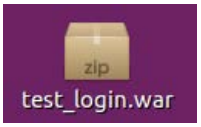
DockerMySQL 開啟 mysql。

```
dockerlab1221@ubuntu:~$ sudo docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS                          PORTS               NAMES
cde5c66e1076        aeea                "catalina.sh run"        7 days ago          Exited (143) 47 hours ago                           tomcat
946e717ea76a        busybox             "nslookup google.com"    9 days ago          Exited (0) 9 days ago                               thirsty_wescoff
a09310afe6b4        busybox             "nslookup google.com"    9 days ago          Exited (0) 9 days ago                               sleepy_neumann
d3b25935f0a8        791                 "docker-entrypoint.s…"   9 days ago          Exited (0) 17 seconds ago                           mysql
dockerlab1221@ubuntu:~$ sudo docker start d3b25935f0a8
d3b25935f0a8
```

7.將範例程式編譯過後產生的 war 檔複製到 DockerLab 的桌面。

(補充教學 : IntelliJ IDEA 將專案打包成 war 檔)

(補充教學 : eclipse 將專案打包成 war 檔)

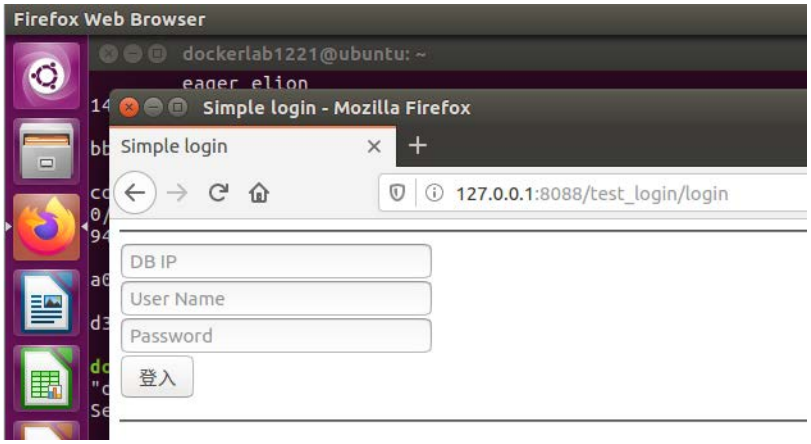


DockerLab 下指令將 war 檔傳入 docker tomcat 執行。

sudo docker cp Desktop/<war 檔> <tomcat 的 container id>:/usr/local/tomcat/webapps

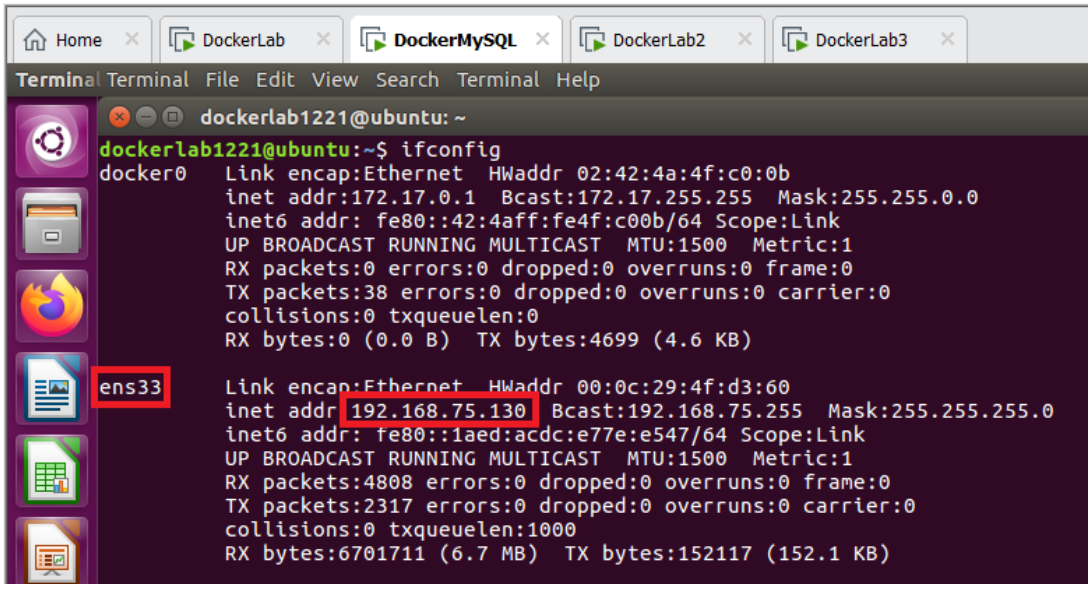
```
dockerlab1221@ubuntu:~$ sudo docker cp Desktop/test_login.war cde5c66e1076:/usr/local/tomcat/webapps
```

8. DockerLab 開啟瀏覽器(http://127.0.0.1:8088/<war 檔名>/<jsp 檔名>)，測試登入系統是否正常。

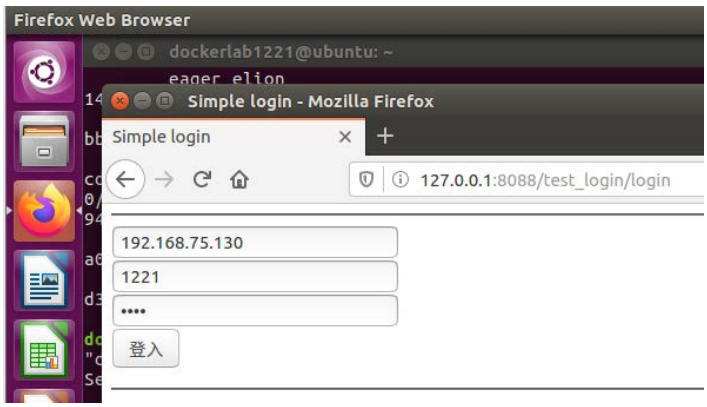


9.在 DockerMySQL 查詢 mysql 的 ip。

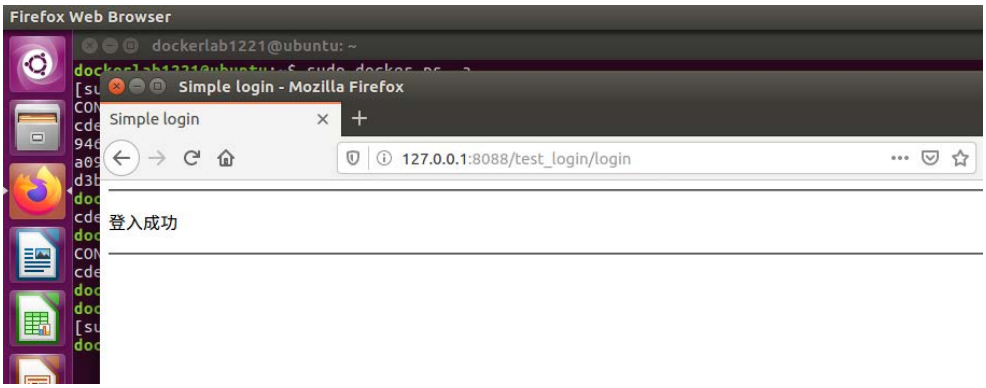
ifconfig



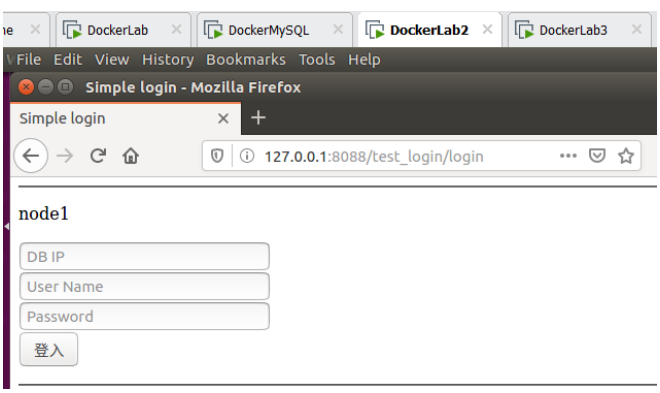
10. 將網頁中 DB IP 的部分設為剛查到的 mysql 的 ip，並輸入 user name 和 password(docker 的 mysql 範例帳號密碼皆為 1221)。

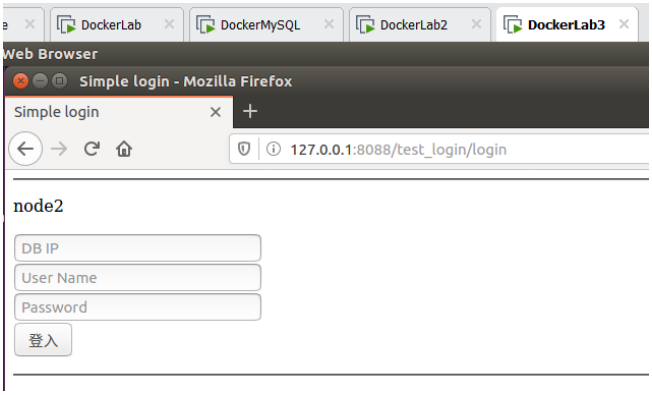


11.登入成功



12. 在範例程式中多加一些字(如:node1、node2 等字樣)，並分別在 DockerLab2、DockerLab3 重複步驟 7 和 8 重新編譯範例程式產生 war 檔，方便自己後續識別是連接到不同 node。





13.停掉 DockerLab 的 tomcat。

sudo docker stop <container id>

```
dockerlab1221@ubuntu:/$ sudo docker stop cde5c66e1076
cde5c66e1076
```

14. 用 ifconfig 查詢 DockerLab、DockerLab2、 DockerLab3 的 ip，並記起來。

例如:

	IP Address	Role
DockerLab	192.168.75.128	master
DockerLab2	192.168.75.131	node1
DockerLab3	192.168.75.132	node2

(註: master 為主要控制節點，node 為應用程式工作節點)

15.master:初始化 swarm 。

sudo docker swarm init

```
dockerlab1221@ubuntu:~$ sudo docker swarm init
[sudo] password for dockerlab1221:
Swarm initialized: current node (usvvd4n3jx1lzu27p3annyf0b) is now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-1x6axb8jvk9ppi4o8jys52eoznzw0wh03buatlw73vvj3la1y8-1nwadro5g6wmyi9chqjwpc8i9 192.168.75.128:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.
```

16.node: 使用上一個步驟隨機產生的 token(即上圖中的 docker swarm join --token ….那行)來為 swarm 連接新 node。(註:記得加 sudo)

```
dockerlab1221@ubuntu:~$ sudo docker swarm join --token SWMTKN-1-1x6axb8jvk9ppi4o8jys52eoznzw0wh03buatlw73vvj3la1y8-1nwadro5g6wmyi9chqjwpc8i9 192.168.75.128:2377
This node joined a swarm as a worker.
```

node:開啟 /etc/default/docker 並將下列程式碼於檔案最下方加入：

sudo gedit /etc/default/docker

```
DOCKER_OPTS="-H <master ip>:2377 -H unix:///var/run/docker.sock"
```

```
dockerlab1221@ubuntu:~$ sudo gedit /etc/default/docker
```

Open ▾ docker /etc/default

```
# Docker Upstart and SysVinit configuration file

#
# THIS FILE DOES NOT APPLY TO SYSTEMD
#
#   Please see the documentation for "systemd drop-ins":
#   https://docs.docker.com/engine/admin/systemd/
#

# Customize location of Docker binary (especially for development testing).
#DOCKERD="/usr/local/bin/dockerd"

# Use DOCKER_OPTS to modify the daemon startup options.
#DOCKER_OPTS="--dns 8.8.8.8 --dns 8.8.4.4"

# If you need Docker to use an HTTP proxy, it can also be specified here.
#export http_proxy="http://127.0.0.1:3128/"

# This is also a handy place to tweak where Docker's temporary files go.
#export DOCKER_TMPDIR="/mnt/bigdrive/docker-tmp"
DOCKER_OPTS="-H 192.168.75.128:2377 -H unix:///var/run/docker.sock"
```

node:存檔後，重新啟動 docker(也就是重新啟動該台虛擬機)。
再次啟動後，將原本執行的 container (如:tomcat)啟動。
(若不知道 <container_id> ，可以透過 sudo docker ps -a 指令尋找)

sudo docker start <container_id>

17. master:查看是否有連接成功 node。

sudo docker node ls

```
dockerlab1221@ubuntu:/$ sudo docker node ls
ID                            HOSTNAME            STATUS              AVAILABILITY        MANAGER STATUS      ENGINE VERSION
r32glp76kty63gjvv1frqgtok     ubuntu              Ready               Active                                  18.09.7
usvvd4n3jx1lzu27p3annyf0b *   ubuntu              Ready               Active              Leader              18.09.7
xcw1lil1e1bx38jrwstciel5v     ubuntu              Ready               Active                                  18.09.7
```

18. master:安裝 Nginx 套件。

sudo apt-get install nginx

```
dockerlab1221@ubuntu:/$ sudo apt-get install nginx
```

19. master:將下列程式碼加入 /etc/nginx/sites-enabled/default 中：

```
upstream nodes {
        server 192.168.75.131:8088 weight=1; #權重 = 1
        server 192.168.75.132:8088 weight=2; #權重 = 2
}
…
```

```
server {
        location / {
                proxy_pass http://nodes;
        }
}
```

*註:server <node ip>:<tomcat port> weight=1;。

*註:要先寫 upstream nodes {}的部分，再寫 server{}的部分。

(說明:nginx 預設 load balancing 演算法為 round robin，也可以透過上述方式調整每個 nodes 所分配到的權重。)

sudo gedit /etc/nginx/sites-enabled/default

```
dockerlab1221@ubuntu:/$ sudo gedit /etc/nginx/sites-enabled/default
```

Open default /etc/nginx/sites-enabled Save

```
# Default server configuration
#

upstream nodes {
    server 192.168.75.131:8088 weight=1; #權重 = 1
    server 192.168.75.132:8088 weight=2; #權重 = 2
}

server {
        listen 80 default_server;
        listen [::]:80 default_server;

        # SSL configuration
        #
        # listen 443 ssl default_server;
        # listen [::]:443 ssl default_server;
        #
        # Note: You should disable gzip for SSL traffic.
        # See: https://bugs.debian.org/773332
        #
        # Read up on ssl_ciphers to ensure a secure configuration.
        # See: https://bugs.debian.org/765782
        #
        # Self signed certs generated by the ssl-cert package
        # Don't use them in a production server!
        #
        # include snippets/snakeoil.conf;

        root /var/www/html;

        # Add index.php to the list if you are using PHP
        index index.html index.htm index.nginx-debian.html;

        server_name _;

        location / {
                # First attempt to serve request as file, then
                # as directory, then fall back to displaying a 404.
                #try_files $uri $uri/ =404;
                proxy_pass http://nodes;
        }
```

20. master: 打開瀏覽器，輸入 127.0.0.1，出現下圖畫面即是有連到。

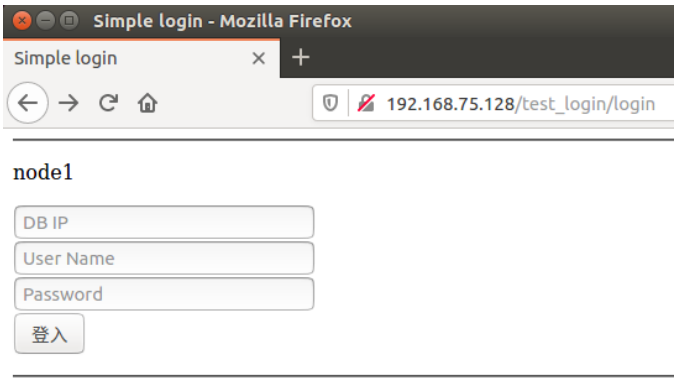
21. master:

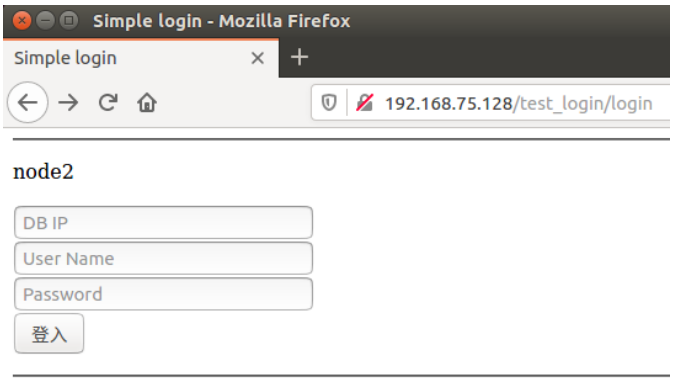
sudo apt-get upgrade nginx

sudo service nginx stop

sudo service nginx start

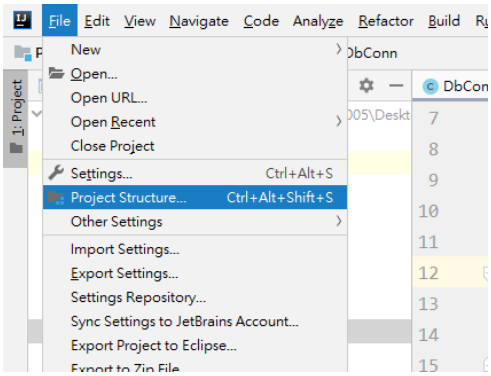
22. master:打開瀏覽器，輸入<master ip>/<war 檔名>/<jsp 檔名>作測試，不斷重整頁面就會發現是連到不同的 node。

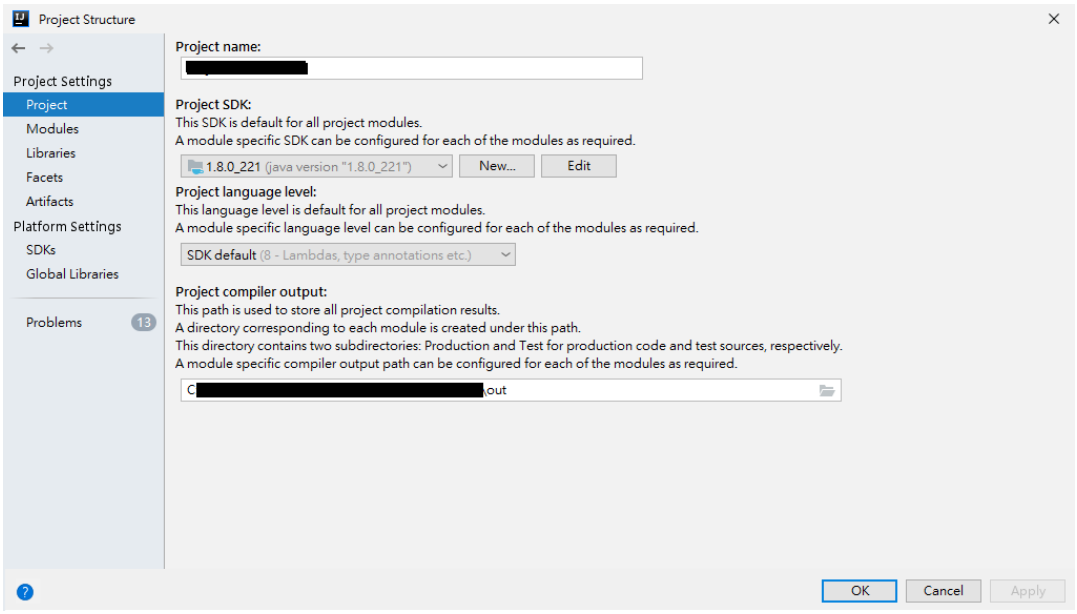




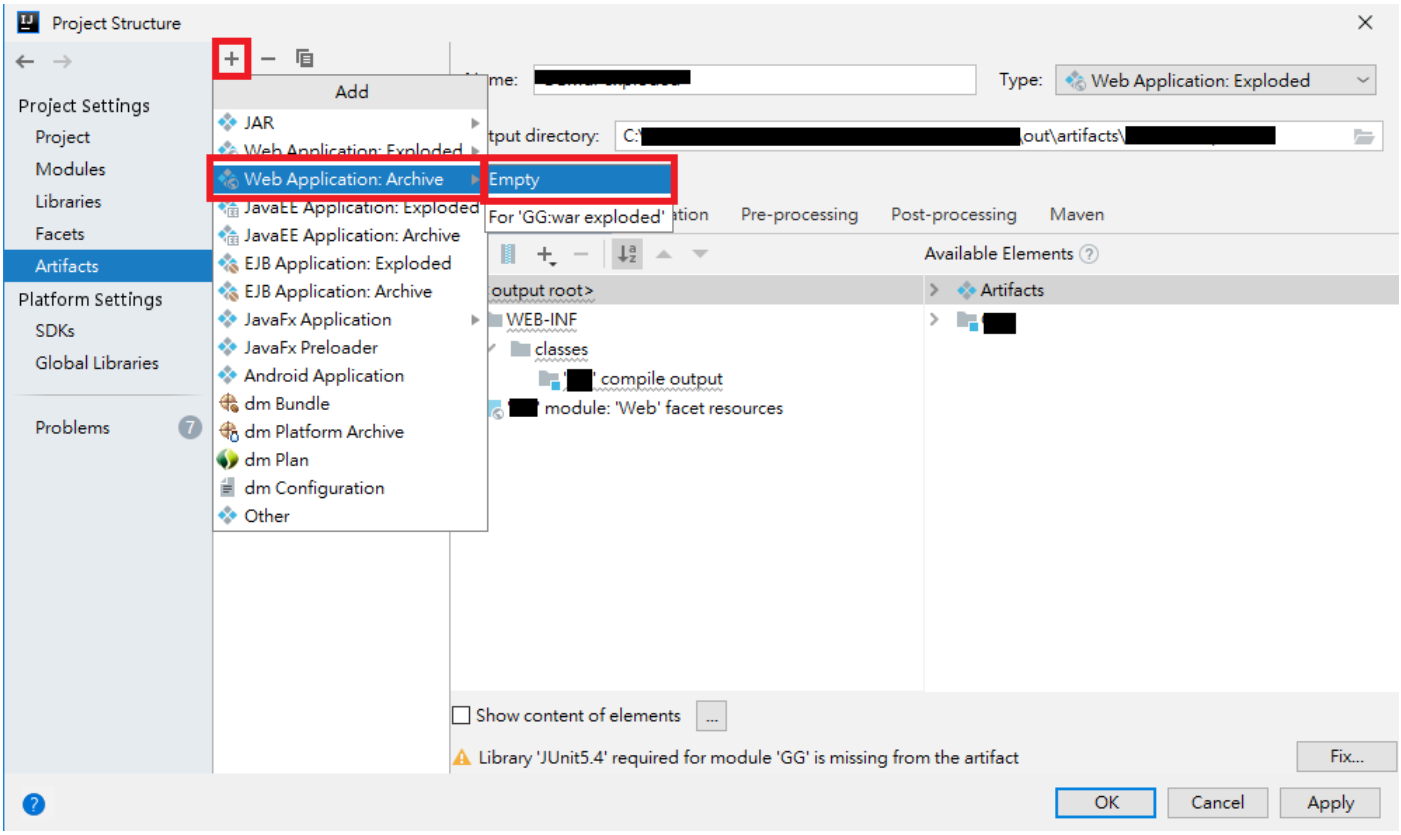
[補充 1] IntelliJ IDEA 將專案打包成 war 檔

1. 點選【File】->【Project Structure】選單，開啟【Project Structure】視窗。

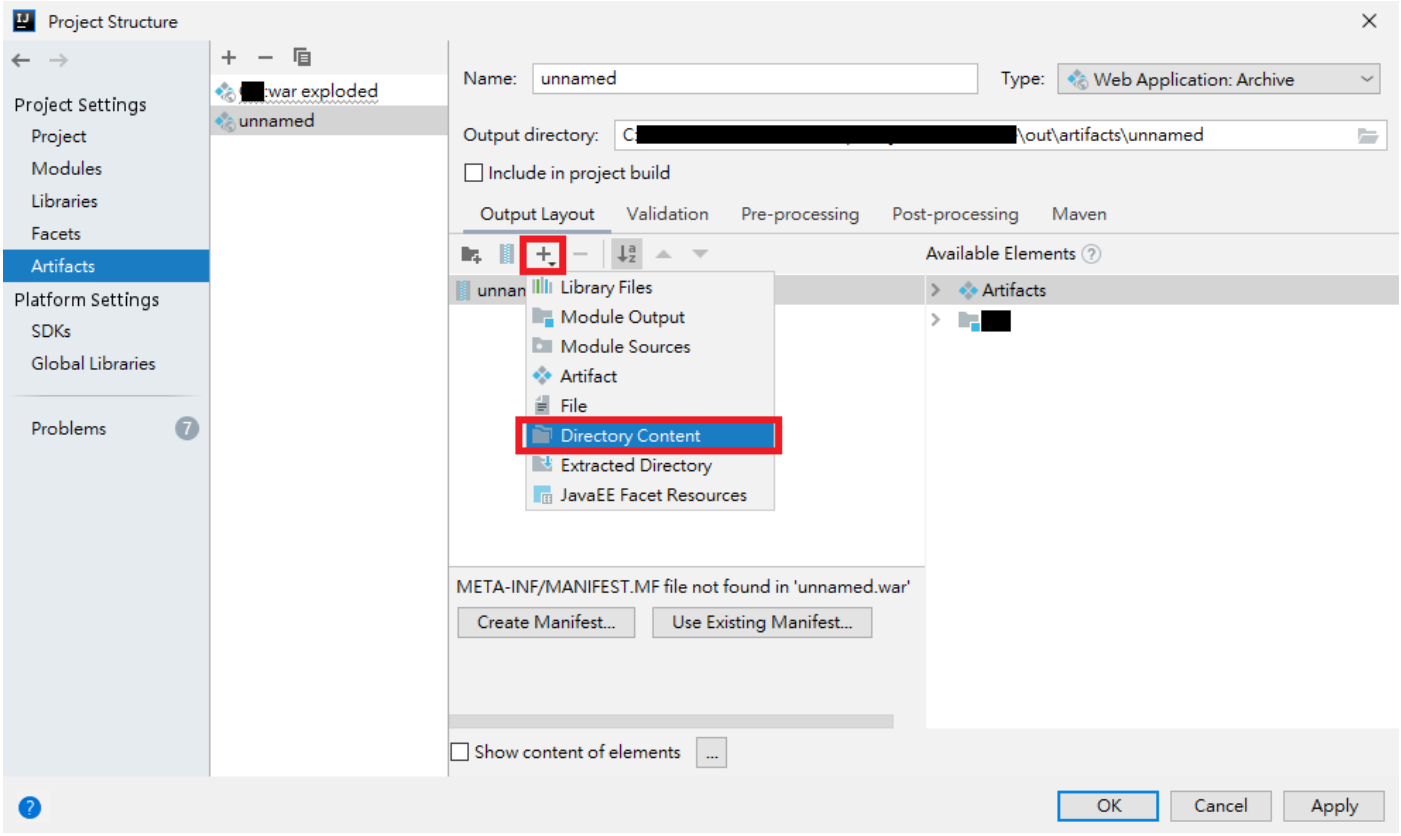




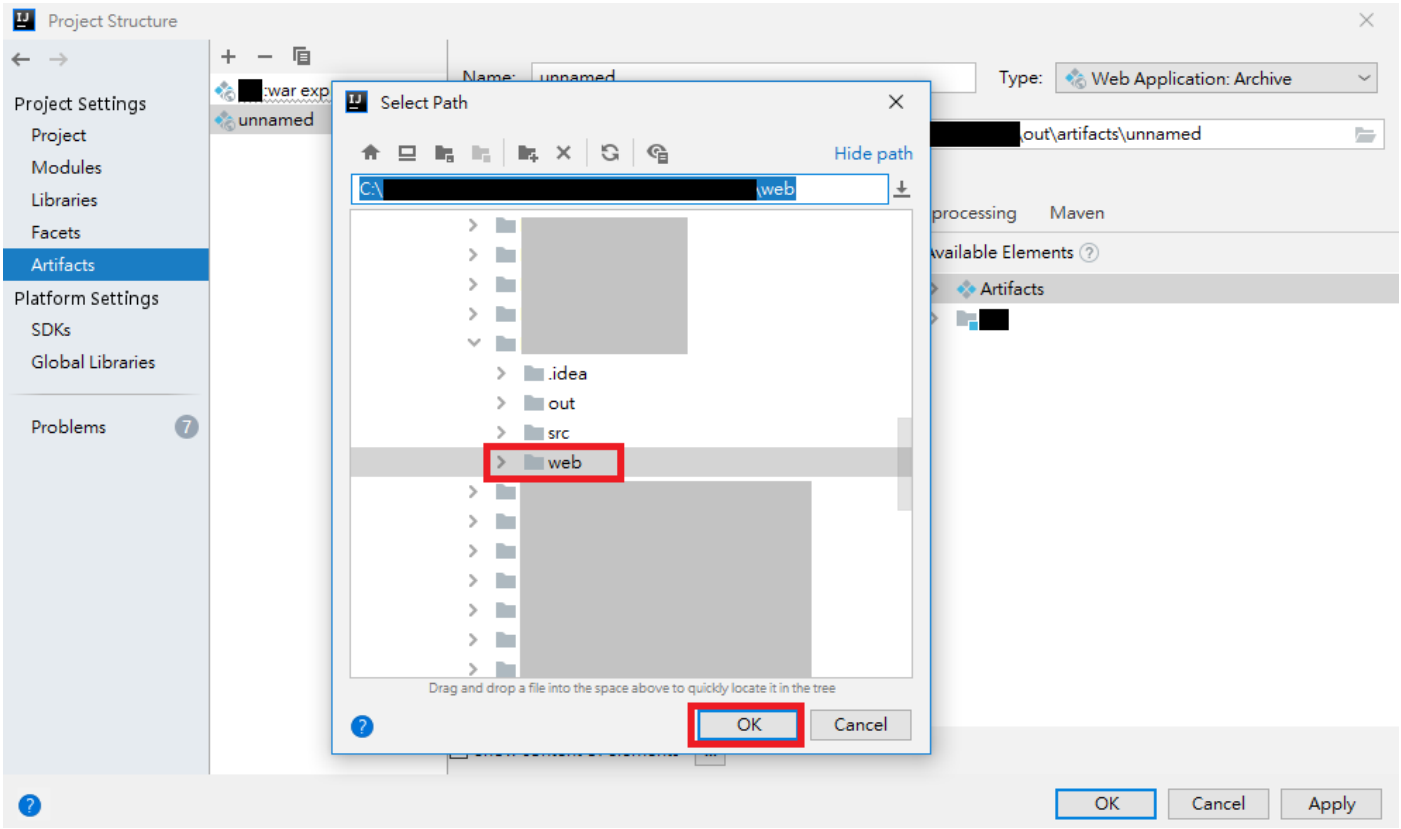
2. 在【ProjectStructure】中選擇左側的【Artifacts】頁籤。
點選中間上面的"+"，選擇【WebApplication:Archive】-> 【Empty】。



3. 點選圖中的"+"，選擇【Directory Content】選單。

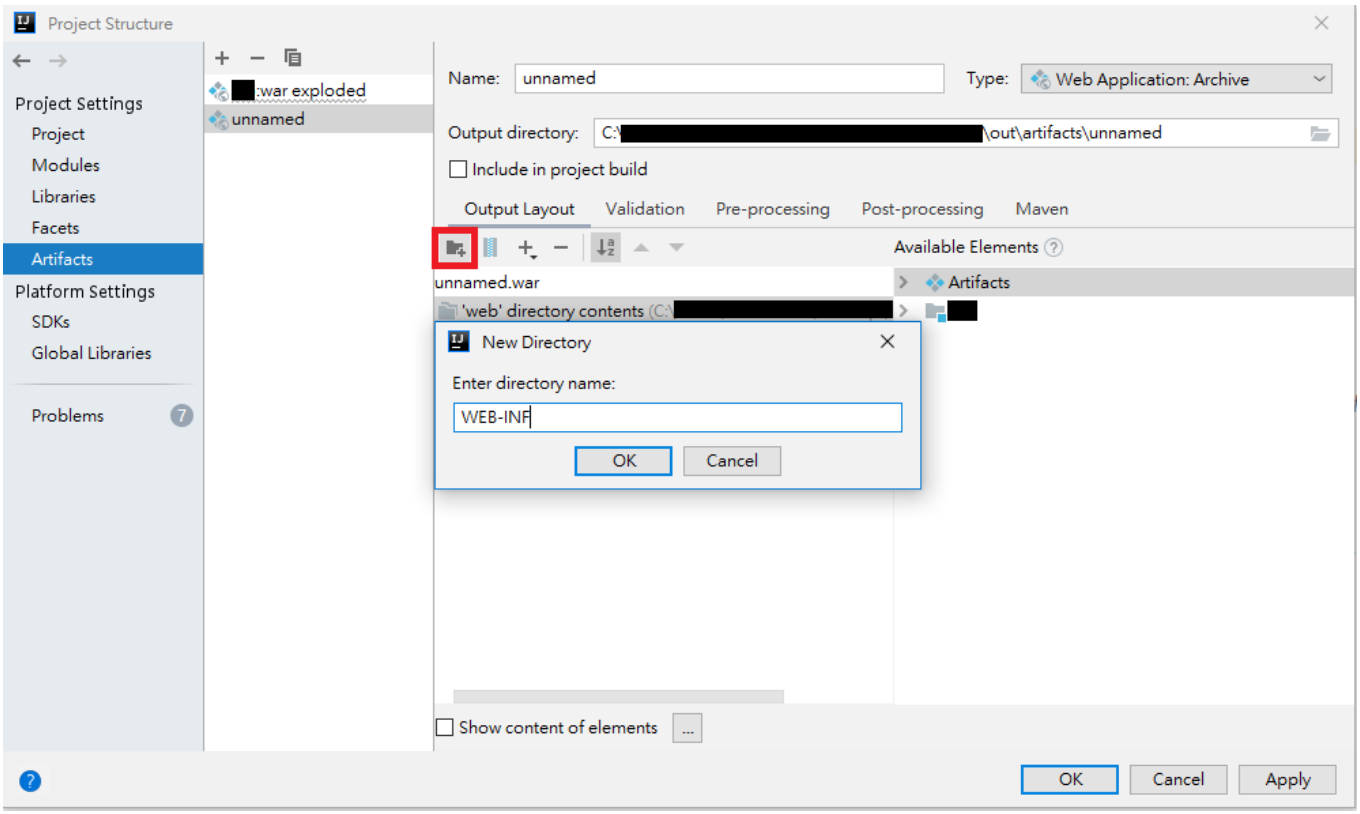


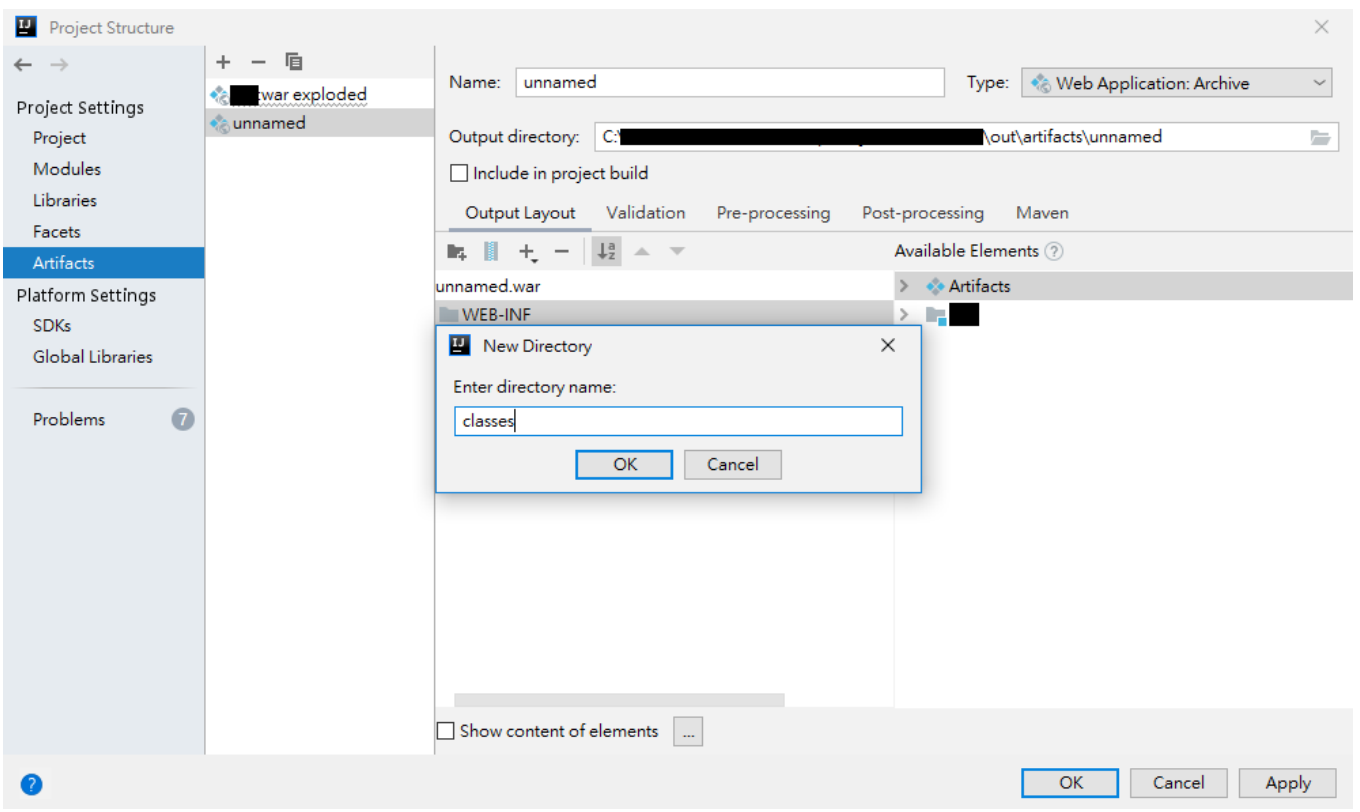
4. 選擇 web root 根目錄。



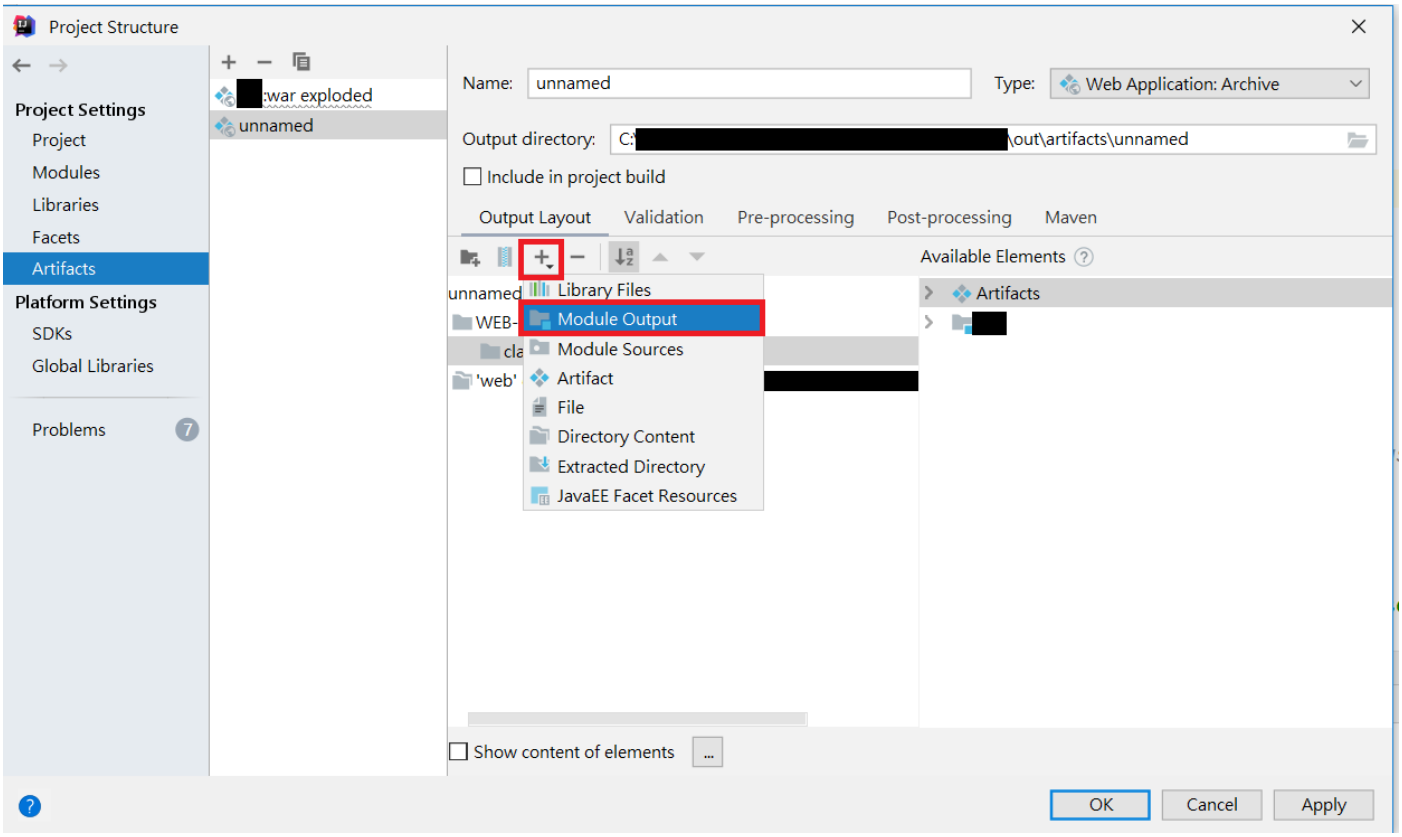
5. 經過上圖的選擇，已經將除 classess 目前之外的結構都準備就緒了。

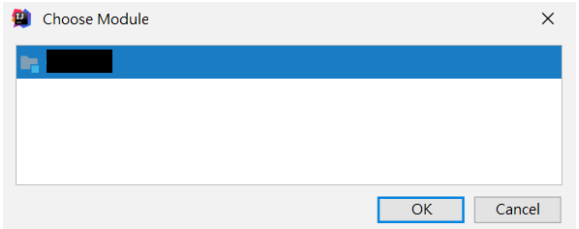
6. 選中 war 包總目錄後，點擊紅框內的圖示，建立【WEB-INF】和子目錄【classes】目錄。



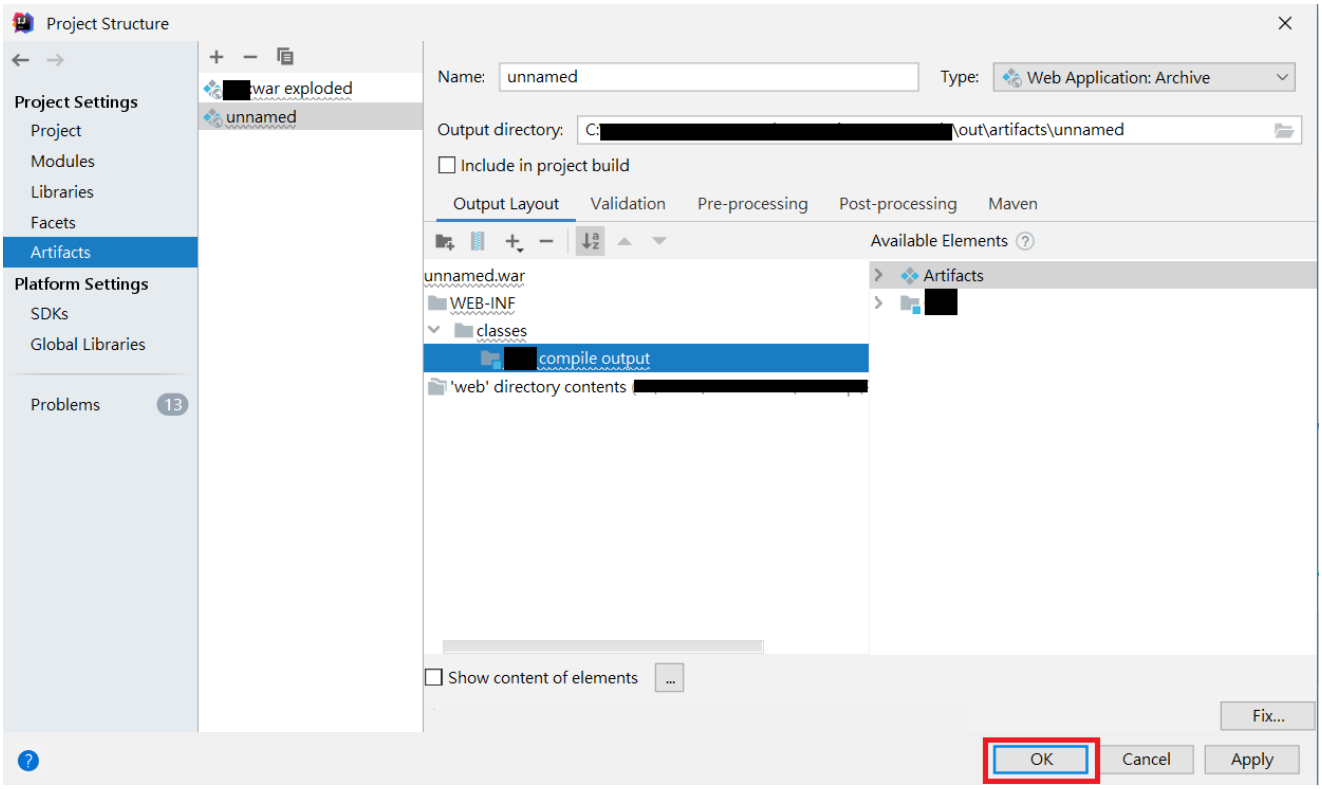


7. 點選紅框內的圖示，選擇【Module Output】選單給【classes】目錄新增內容。
選擇上面內容後，點擊【OK】。

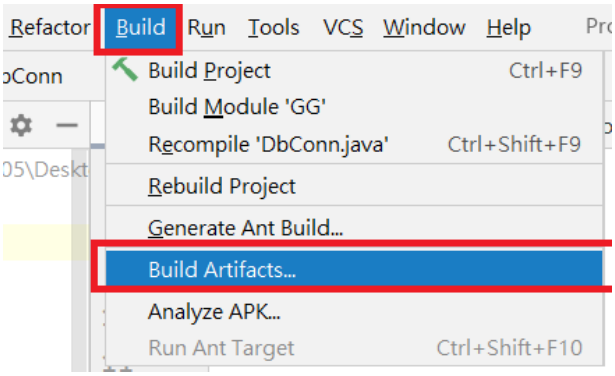


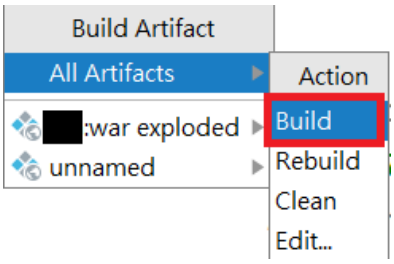


8. 點選圖中的【OK】按鈕，結束 war 的配置。

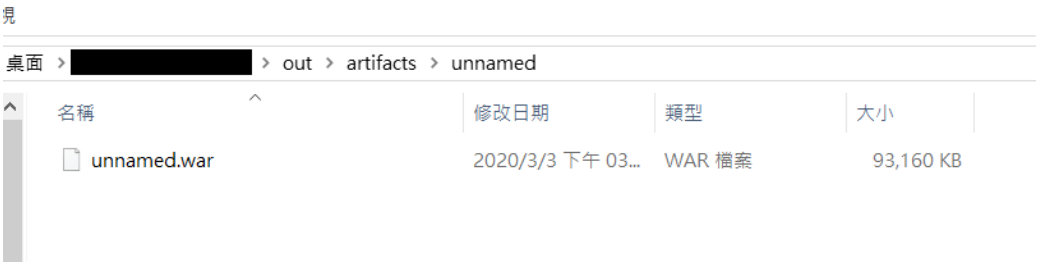


9. 編譯及執行打包 war，點選【Build】->【BuildArtifacts】->【Build】選單。





10. 在資料夾中找到打包好的 war 檔案。

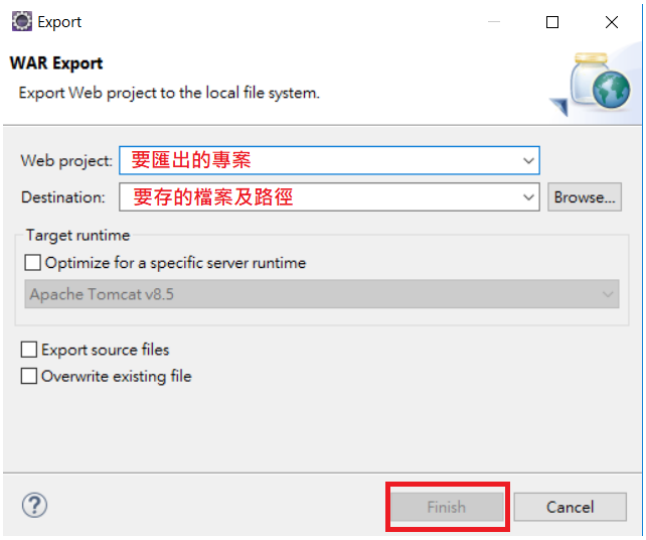


[補充 2] eclipse 將專案打包成 war 檔

1. 在 project 點選右鍵點【export】，再點【WAR file】。
2. WAR 名稱預設就是專案的名稱,可自行修改。

在要匯出的路徑那欄，檔名要加上"war"。

不要勾 Export source files 選項。



範例: