Nginx as Load Balancer - Steps

(Tested on Windows 11)

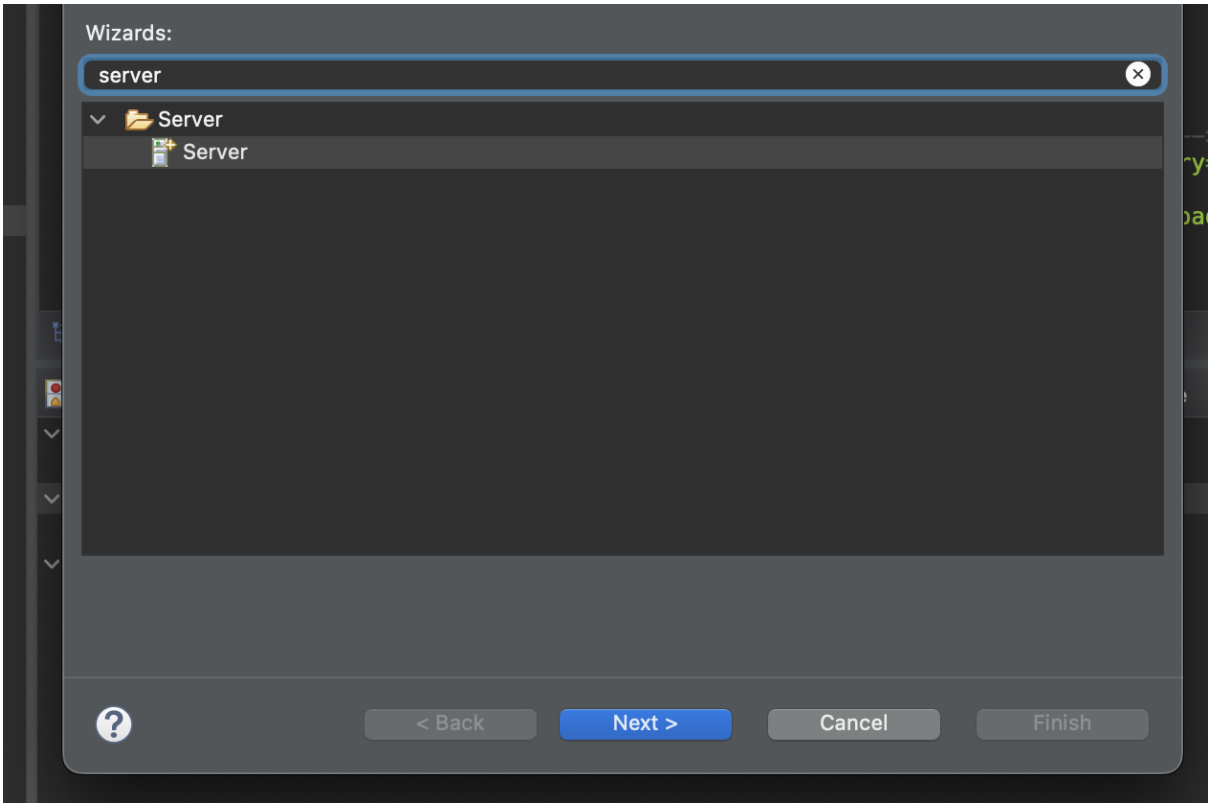
- Install **nginx** using **brew** (*brew install nginx* on terminal)
- start service: *brew services start nginx*
- now you can use command nginx …
- test the config file syntax: *./nginx -t*
- *code /opt/homebrew/etc/nginx/nginx.conf* to edit the file with vscode
- go to http://localhost:8081/ and test that nginx is working
- run the following steps

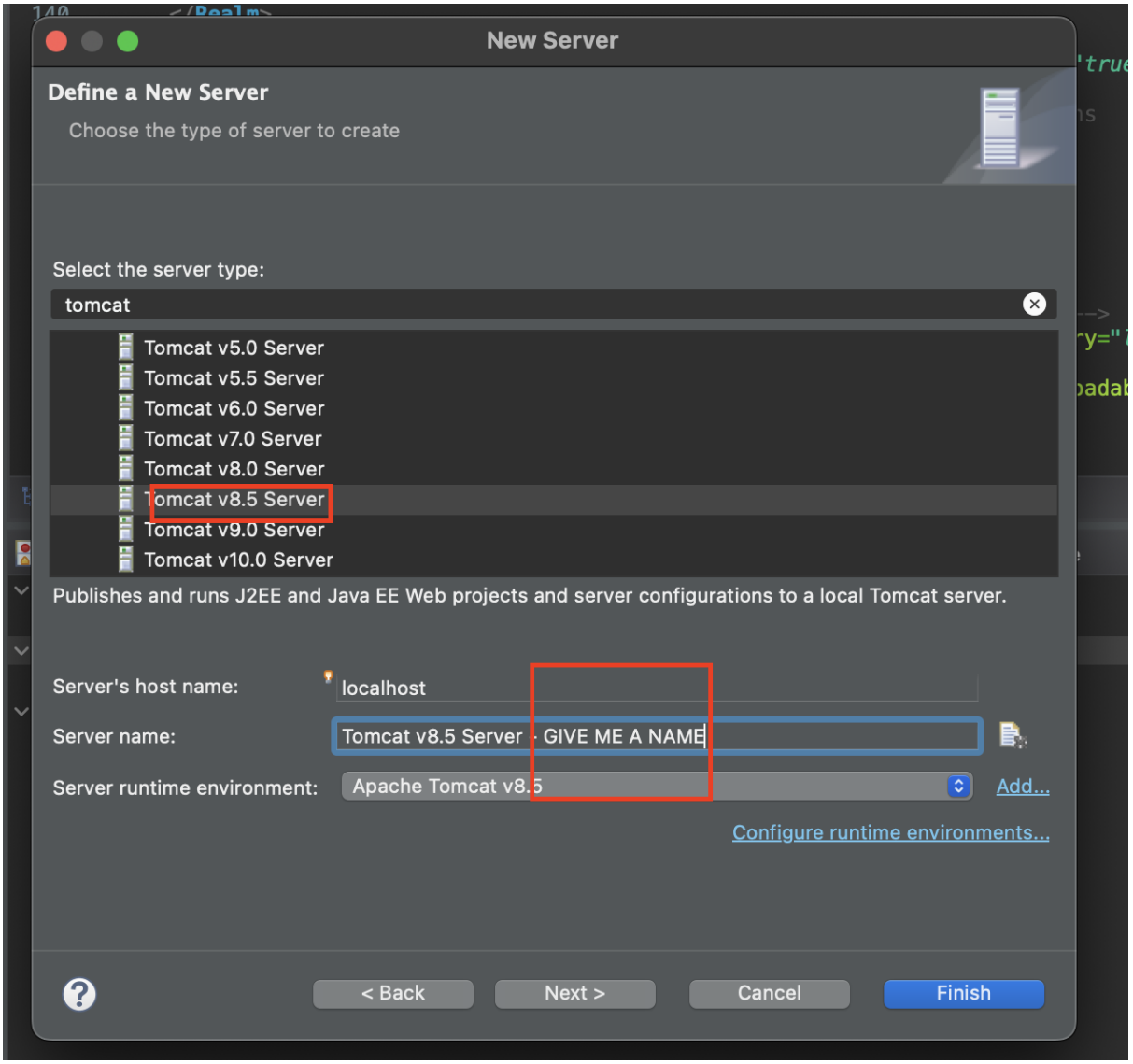
Create the servers on eclipse

Since the three servers configured below are:

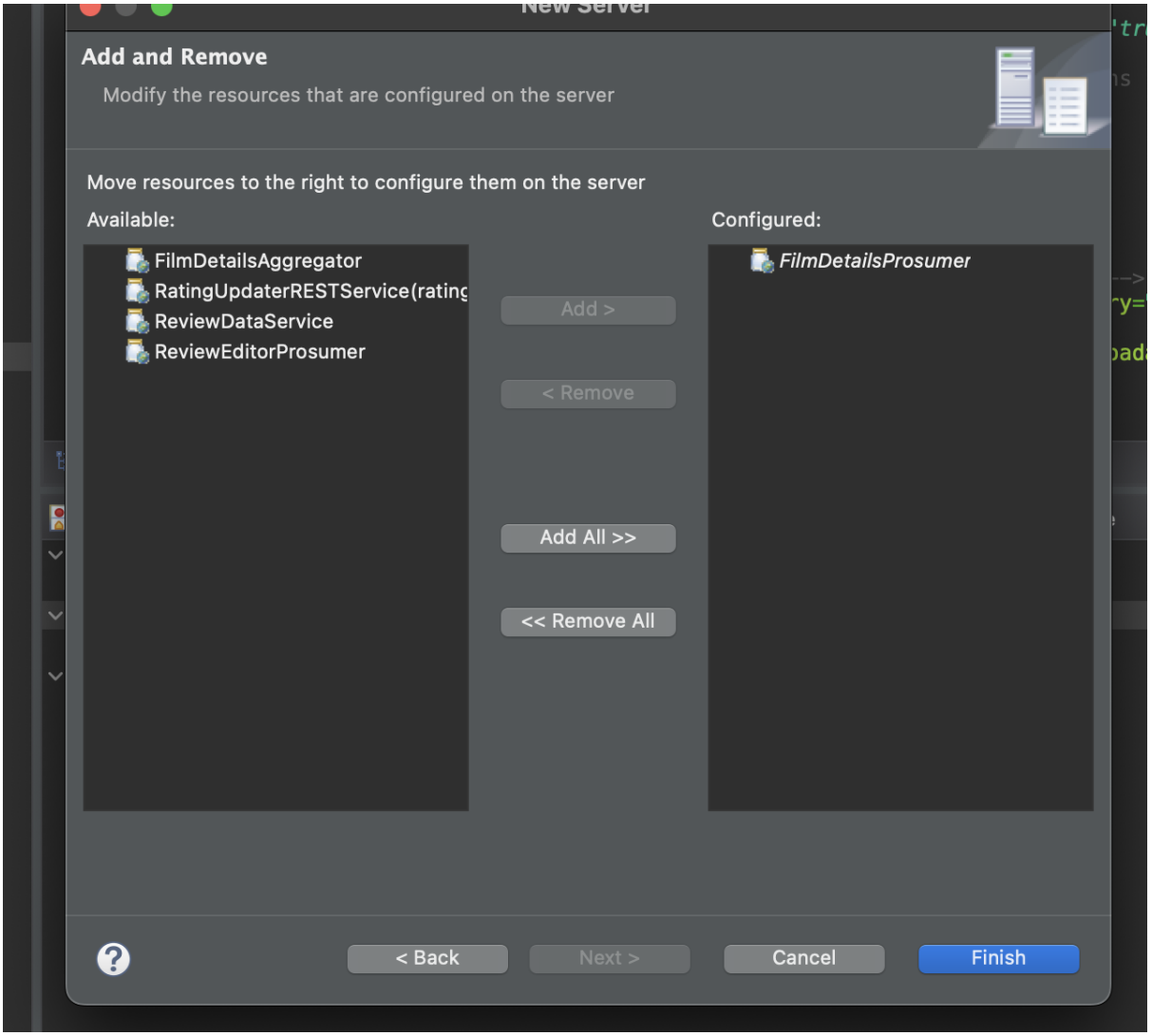
- http://localhost:8080/FoodDetailsProsumer
- http://localhost:8088/FoodDetailsProsumer
- http://localhost:8089/FoodDetailsProsumer

You have to create the servers on eclipse:

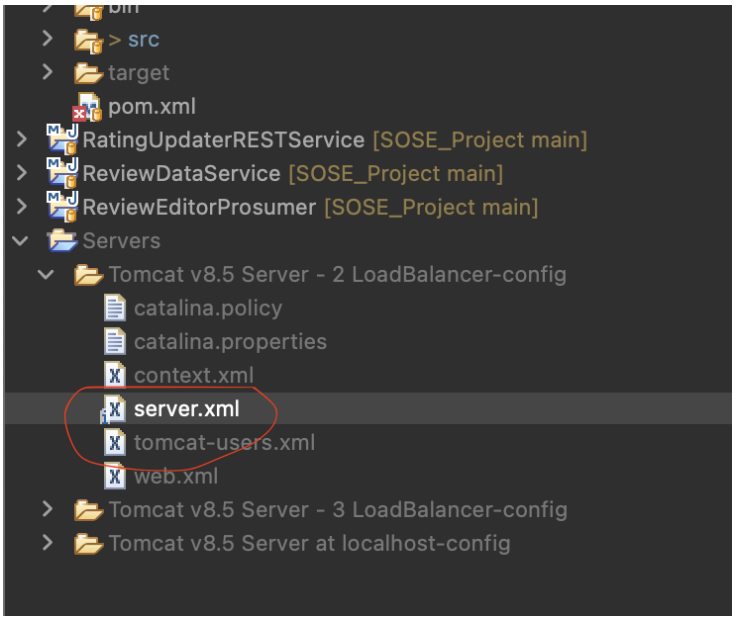




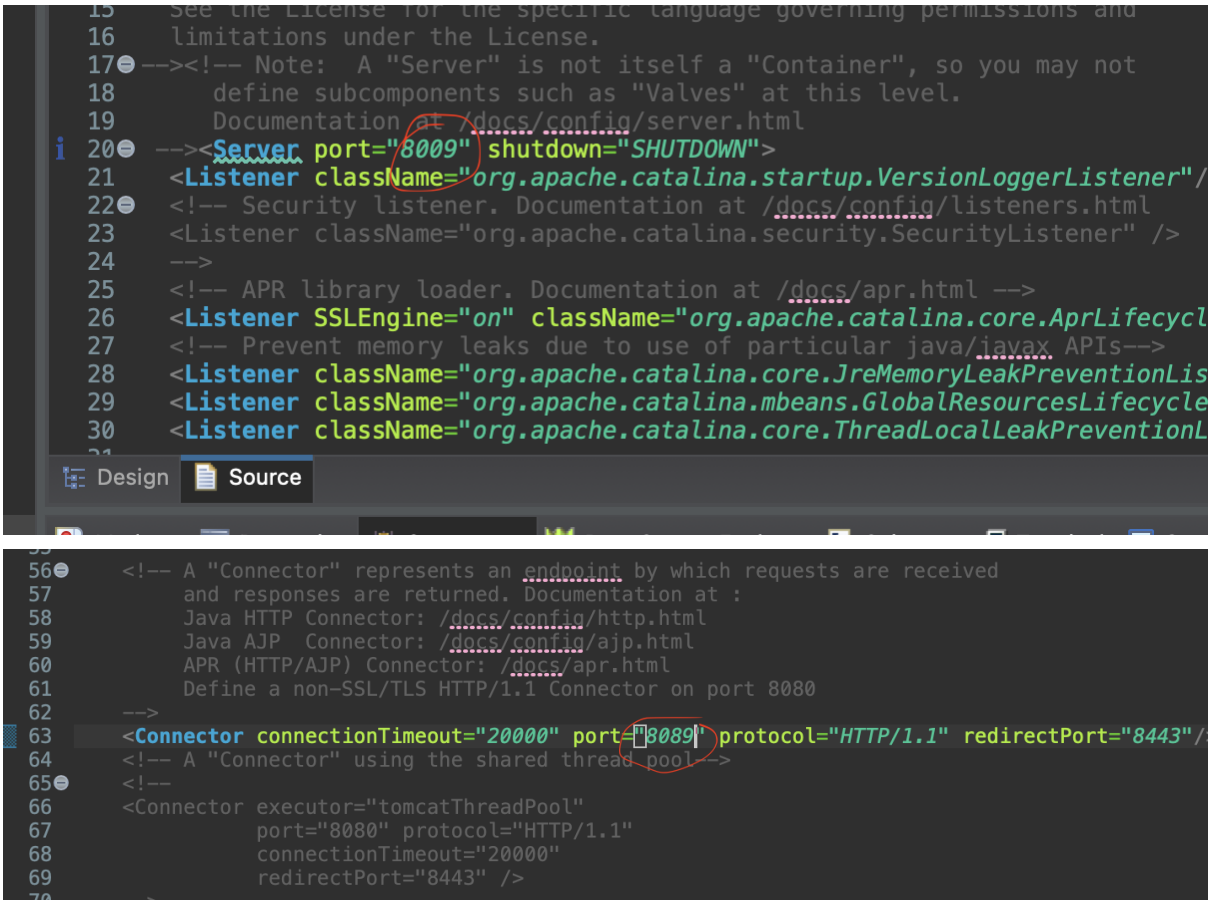
Add the FoodDetailsProsumer to the server.



Edit the server configuration to avoid conflict with ports (edit ***server.xml*** file):



Examples:



```
16    limitations under the License.
17  --><!-- Note:  A "Server" is not itself a "Container", so you may not
18       define subcomponents such as "Valves" at this level.
19       Documentation at /docs/config/server.html
20  --><Server port="8009" shutdown="SHUTDOWN">
21    <Listener className="org.apache.catalina.startup.VersionLoggerListener"/
22    <!-- Security listener. Documentation at /docs/config/listeners.html
23    <Listener className="org.apache.catalina.security.SecurityListener" />
24    -->
25    <!-- APR library loader. Documentation at /docs/apr.html -->
26    <Listener SSLEngine="on" className="org.apache.catalina.core.AprLifecycl
27    <!-- Prevent memory leaks due to use of particular java/javax APIs-->
28    <Listener className="org.apache.catalina.core.JreMemoryLeakPreventionLis
29    <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycle
30    <Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionL
```

Design | Source


```
56    <!-- A "Connector" represents an endpoint by which requests are received
57         and responses are returned. Documentation at :
58         Java HTTP Connector: /docs/config/http.html
59         Java AJP  Connector: /docs/config/ajp.html
60         APR (HTTP/AJP) Connector: /docs/apr.html
61         Define a non-SSL/TLS HTTP/1.1 Connector on port 8080
62    -->
63    <Connector connectionTimeout="20000" port="8089" protocol="HTTP/1.1" redirectPort="8443"/
64    <!-- A "Connector" using the shared thread pool-->
65    <!--
66    <Connector executor="tomcatThreadPool"
67               port="8080" protocol="HTTP/1.1"
68               connectionTimeout="20000"
69               redirectPort="8443" />
```

Configure the nginx.conf file

```
http {
    include       mime.types;
    default_type  application/octet-stream;

    log_format  main  '$server_name to: $upstream_addr [$request] $remote_addr -
$remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  /var/logs/access_nginx.log  main;

    sendfile        on;
    #tcp_nopush     on;

    #keepalive_timeout  0;
    keepalive_timeout  65;
```

```
    #gzip  on;

    upstream samplecluster {
        # The upstream elements lists all
        # the backend servers that take part in
        # the Nginx load balancer example
        server localhost:8080;
        server localhost:8088;
        server localhost:8089;

        }

    server {
        listen       8081;
        server_name  localhost;

        #charset koi8-r;

        access_log  /var/logs/host.access_nginx.log  main;

        location /FoodDetailsProsumer {
            proxy_pass http://samplecluster/FoodDetailsProsumer;
            #root   html;
            #index  index.html index.htm;
        }

        location / {
            proxy_pass http://localhost:8089/FoodDetailsProsumer;
            #root   html;
            #index  index.html index.htm;
        }

        #error_page  404              /404.html;

        # redirect server error pages to the static page /50x.html
        #
        error_page   500 502 503 504  /50x.html;
        location = /50x.html {
            root   html;
        }

        # proxy the PHP scripts to Apache listening on 127.0.0.1:80
        #
        #location ~ \.php$ {
        #    proxy_pass   http://127.0.0.1;
```

```
    #}

    # pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
    #
    #location ~ \.php$ {
    #    root           html;
    #    fastcgi_pass   127.0.0.1:9000;
    #    fastcgi_index  index.php;
    #    fastcgi_param  SCRIPT_FILENAME  /scripts$fastcgi_script_name;
    #    include        fastcgi_params;
    #}

    # deny access to .htaccess files, if Apache's document root
    # concurs with nginx's one
    #
    #location ~ /\.ht {
    #    deny  all;
    #}
}


# another virtual host using mix of IP-, name-, and port-based configuration
#
#server {
#    listen       8000;
#    listen       somename:8080;
#    server_name  somename  alias  another.alias;

#    location / {
#        root   html;
#        index  index.html index.htm;
#    }
#}


# HTTPS server
#
#server {
#    listen       443 ssl;
#    server_name  localhost;

#    ssl_certificate      cert.pem;
#    ssl_certificate_key  cert.key;

#    ssl_session_cache    shared:SSL:1m;
```

```
    #    ssl_session_timeout  5m;

    #    ssl_ciphers  HIGH:!aNULL:!MD5;
    #    ssl_prefer_server_ciphers  on;

    #    location / {
    #        root   html;
    #        index  index.html index.htm;
    #    }
    #}
    include servers/*;
}
```

Test

test to http://localhost:8081/FoodDetailsProsumer