



EBook Gratis

APRENDIZAJE

nginx

Free unaffiliated eBook created from
Stack Overflow contributors.

#nginx

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con nginx.....	2
Observaciones.....	2
Versiones.....	2
Examples.....	2
Instalación y configuración.....	2
Instalación de Nginx en Debian y distribuciones basadas en Debian como Ubuntu.....	3
Reiniciar NGINX.....	3
Ejemplo de Ubuntu.....	3
Recargar el archivo de configuración NGINX.....	3
Ejemplo de Ubuntu 14.04.....	4
Ejemplo de Ubuntu 16.04.....	4
Cierre NGINX.....	4
Nginx dentro.....	4
Probar si sus cambios en nginx.config son válidos.....	5
Capítulo 2: Configuraciones de Nginx.....	6
Examples.....	6
Estructura del archivo de configuración.....	6
Directivas simples.....	6
Directiva de bloque.....	6
Contexto.....	6
Comentario.....	6
Archivo de configuración de prueba NGINX.....	6
Especificar archivo de configuración para cargar.....	7
cambios de configuración sin reinicio.....	7
Iniciar sesión en Syslog, incl. mapeo de códigos HTTP a severidades de syslog.....	7
Limitar los métodos de solicitud.....	8
Capítulo 3: Explotación florestal.....	9
Examples.....	9
Ejemplo basico.....	9

Sintaxis	9
Reabrir los archivos de registro.....	9
Evite el registro de favicon.ico y robots.txt.....	9
Capítulo 4: Integración de blog de Wordpress con la aplicación de rieles usando nginx	10
Examples.....	10
bloque de servidor nginx.....	10
Capítulo 5: proxy inverso nginx	12
Examples.....	12
proxy inverso simple.....	12
Capítulo 6: Redirecciones útiles	13
Examples.....	13
HTTPS Redirect.....	13
Redirigir las solicitudes a otro servidor.....	13
Redirección de sitio móvil.....	13
Ubicación HTTP en el servidor HTTPS.....	14
No permitir solicitudes basadas en código de país o host.....	14
Capítulo 7: Usando nginx para proporcionar URLs de navegador limpias	16
Examples.....	16
Redireccionar vs proxy inverso.....	16
Creditos	18

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [nginx](#)

It is an unofficial and free nginx ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official nginx.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con nginx

Observaciones

NGINX se pronuncia como "motor x" y se usa comúnmente como un servidor de alto rendimiento para los protocolos HTTP, HTTPS, SMTP, POP3 e IMAP. Puede utilizarse como servidor proxy inverso, caché HTTP o balanceo de carga.

Es un proyecto de código abierto con fuente disponible [aquí](#) .

Versiones

Versión	Fecha de lanzamiento original	Ultima versión	Estado	Fecha de lanzamiento
0.5	2006-12-04	0.5.38	Legado	2009-09-14
0.6	2007-06-14	0.6.39	Legado	2009-09-14
0.7	2008-05-19	0.7.69	Legado	2011-07-19
0.8	2009-06-02	0.8.55	Legado	2011-07-19
1.0	2011-04-12	1.0.15	Legado	2012-04-12
1.2	2012-04-23	1.2.9	Legado	2013-05-13
1.4	2013-04-24	1.4.7	Legado	2014-03-18
1.6	2014-04-24	1.6.3	Legado	2015-04-07
1.8	2015-04-21	1.8.1	Legado	2016-01-26
1.9	2015-04-28	1.9.15	Legado	2016-04-19
1.10	2016-04-26	1.10.3	Estable	2016-05-31
1.11	2016-05-24	1.11.9	Chutarse	2016-07-26

Examples

Instalación y configuración

Nginx es un servidor web utilizado para atender solicitudes HTTP a través de Internet.

Nginx está disponible en Linux, Windows y otros sistemas operativos como descarga directa, y

también puede compilarse desde la fuente. Para obtener instrucciones detalladas, consulte la [referencia oficial de Nginx](#).

ubuntu / debian

La versión estable de nginx está disponible en repositorio oficial, se puede instalar usando

```
sudo apt-get install nginx
```

Instalará y configurará los archivos de inicio del sistema, pero si necesita la última versión, es posible que deba agregar un paquete oficial.

```
sudo add-apt-repository ppa:nginx/stable
sudo apt-get update
sudo apt-get install nginx
```

Las instrucciones anteriores instalarán la última edición estable.

Instalación de Nginx en Debian y distribuciones basadas en Debian como Ubuntu

Ejecute el siguiente comando para instalar nginx.

```
sudo apt-get install nginx
```

Por defecto, Nginx se inicia automáticamente cuando se instala. Puede acceder a la página de inicio predeterminada de Nginx para confirmar que el software se ejecuta correctamente visitando el nombre de dominio o la dirección IP pública de su servidor en su navegador web.

pero si necesita la última versión, es posible que necesite agregar ppa oficial.

```
sudo add-apt-repository ppa:nginx/stable
sudo apt-get update
sudo apt-get install nginx
```

Reiniciar NGINX

Como usuario root:

```
nginx -s restart
```

Ejemplo de Ubuntu

```
sudo service nginx restart
```

Recargar el archivo de configuración NGINX

Como usuario root:

```
sudo nginx -s reload
```

Ejemplo de Ubuntu 14.04

```
sudo service nginx reload
```

Ejemplo de Ubuntu 16.04

```
sudo systemctl reload nginx
```

Antes de volver a cargar, es una buena idea verificar la configuración en busca de errores de sintaxis:

```
sudo nginx -t
```

O

```
sudo service nginx configtest
```

Cierre NGINX

Ejecutar como usuario root.

Cierre rápido:

```
nginx -s stop
```

Cierre agradado:

```
nginx -s quit
```

Nginx dentro

Uno de los mayores atractivos de Nginx es la diferencia en cómo funciona internamente en comparación con los otros servidores populares, especialmente Apache.

Los servidores son programas ocupados, ya que tienen que atender solicitudes de varios clientes. Cuantas más solicitudes un servidor pueda servir con éxito por segundo, mejor.

Nginx trabaja en un paradigma de concurrencia conocido como IO asíncrono.

En un servidor convencional, un hilo está dedicado a una solicitud. Esto significa que, una vez que un subproceso ocupa una solicitud, no está disponible para otras solicitudes. Pero en

realidad, un subproceso podría hacerlo mucho mejor al aceptar un montón de solicitudes y atenderlas simultáneamente. IO asíncrono es lo que permite esto.

Nginx, por lo tanto, con su arquitectura de E / S asíncrona, puede atender muchas solicitudes dentro de un hilo.

Otra cosa buena sobre Nginx es su huella de recursos relativamente más reducida. En comparación con Apache, Nginx tiene menos recursos, y esto lo hace adecuado para servidores en la nube, lo que no suele ser muy poderoso.

Ciertamente hay otros servidores Async IO por ahí, pero Nginx es el más compatible entre todos en términos de pluginx (también conocido como Módulos Nginx).

Probar si sus cambios en nginx.config son válidos

Ejemplo de Ubuntu 14.04

```
sudo nginx -t
```

Lea Empezando con nginx en línea: <https://riptutorial.com/es/nginx/topic/1121/empezando-con-nginx>

Capítulo 2: Configuraciones de Nginx

Examples

Estructura del archivo de configuración

nginx consta de módulos que se controlan mediante directivas especificadas en el archivo de configuración.

Directivas simples

Una directiva simple consiste en el nombre y los parámetros separados por espacios y termina con un punto y coma (;).

Directiva de bloque

Una directiva de bloque tiene la misma estructura que una directiva simple, pero en lugar del punto y coma finaliza con un conjunto de instrucciones adicionales entre llaves ({}).

Contexto

Si una directiva de bloque puede tener otras directivas entre llaves, se denomina contexto (ejemplos: eventos, http, servidor y ubicación).

Las directivas colocadas en el archivo de configuración fuera de cualquier contexto se consideran en el contexto principal. Los eventos y las directivas http residen en el contexto principal, el servidor en http y la ubicación en el servidor.

Comentario

El resto de una línea después del signo # se considera un comentario.

Archivo de configuración de prueba NGINX

Puede verificar los errores de sintaxis y los archivos referenciados en un archivo de configuración NGINX antes de ejecutarlo con:

```
nginx -t
```

Alternativamente, puede ejecutar el script de servicio

```
service nginx configtest
```

Mientras que estos dos comandos le dirán si su nueva configuración de nginx está bien [sin matar su instancia actual]. Configtest utiliza el servicio en ejecución y le informa si pasa o no la verificación, mientras que nginx -t no solo verificará la configuración sino que imprimirá cualquier información, advertencia y mensajes de error.

Fuente: <http://devget.net/nginxapache/nginx-configtest-vs-nginx-t/>

Especificar archivo de configuración para cargar

```
nginx -c <file name>
```

Inicie NGINX con un archivo de configuración explícito.

cambios de configuración sin reinicio

```
nginx -s reload
```

Iniciar sesión en Syslog, incl. mapeo de códigos HTTP a severidades de syslog.

Pega este fragmento en algún lugar del bloque `http {}`; o colóquelo en su propio archivo en la carpeta `/etc/nginx/conf.d/`. También vea los [documentos oficiales](#) para iniciar sesión en syslog.

```
#
# Access Log
#
log_format fmt_syslog '[$time_local] $status $remote_addr $http_host "$request"
$body_bytes_sent $request_time "$http_user_agent" $remote_user';
map $status $log_is_error { "~^5\d\d" 1; default 0; }
map $status $log_is_warn { "~^4[0-8]{2}" 1; default 0; }
map $status $log_is_info { "~^[1-3]\d\d" 1; default 0; }
access_log
syslog:server=unix:/run/systemd/journal/syslog,nohostname,facility=local2,severity=error
fmt_syslog if=$log_is_error;
access_log
syslog:server=unix:/run/systemd/journal/syslog,nohostname,facility=local2,severity=warn
fmt_syslog if=$log_is_warn;
access_log
syslog:server=unix:/run/systemd/journal/syslog,nohostname,facility=local2,severity=info
fmt_syslog if=$log_is_info;
#
# Error Log
#
error_log syslog:server=unix:/run/systemd/journal/syslog,nohostname,facility=local2 error;
```

Este ejemplo asume que rsyslog (o similar) está escuchando en Socket

`/run/systemd/journal/syslog`, como está predeterminado en Debian 8 cuando journald ha activado [ForwardToSyslog](#). Usando este zócalo, se pasa por alto el journald. Si ese socket no está disponible, intente `/dev/log` lugar.

Siéntase libre de usar otra instalación en lugar de local2. También puede cambiar el [log_format](#) para satisfacer sus necesidades.

Limitar los métodos de solicitud

Un sitio web habitual solo necesita 3 métodos HTTP: `GET` , `HEAD` y `POST` . Bloquea todos los otros métodos usando [limit_except](#) :

```
location / {
    [...]
    # Note: GET includes HEAD
    limit_except GET POST {
        deny all;
    }
    [...]
}
```

Lea Configuraciones de Nginx en línea: <https://riptutorial.com/es/nginx/topic/2550/configuraciones-de-nginx>

Capítulo 3: Explotación florestal

Examples

Ejemplo basico

Sintaxis

```
Syntax: log_format name string ...;
Syntax: access_log path [format [buffer=size] [gzip[=level]] [flush=time] [if=condition]];
access_log off;
```

Usándolos juntos

```
log_format compression '$remote_addr - $remote_user [$time_local] '
    '$request' $status $bytes_sent '
    '$http_referer' '$http_user_agent' '$gzip_ratio';

access_log /spool/logs/nginx-access.log compression buffer=32k;
error_log /spool/logs/nginx-error.log;
```

Reabrir los archivos de registro

Como usuario root ejecuta:

```
nginx -s reopen
```

Evite el registro de favicon.ico y robots.txt

```
location = /favicon.ico {
    log_not_found off;
    access_log off;
}

location = /robots.txt {
    allow all;
    log_not_found off;
    access_log off;
}
```

Lea Explotación florestal en línea: <https://riptutorial.com/es/nginx/topic/2551/explotacion-florestal>

Capítulo 4: Integración de blog de Wordpress con la aplicación de rieles usando nginx.

Examples

bloque de servidor nginx

Una vez que haya configurado y configurado los ajustes php5-fpm y wordpress, puede configurar el archivo `/etc/nginx/conf/nginx.conf` como se muestra a continuación.

Debe definir los bloques de ubicación dentro del bloque del servidor y volver a escribir la url allí como se define.

```
server {
listen 443 ssl;
    server_name abc.co.uk;
    root /home/ubuntu/www/abc/current/public;
    try_files $uri/index.html $uri @unicorn;
    ssl on;
    ssl_certificate /etc/nginx/ssl/abc.crt;
    ssl_certificate_key /etc/nginx/ssl/abc.key;
    location /blog/wp-admin/ {
        root /var/www/html/;
        index index.php;
        try_files $uri $uri/ /index.php?$args;
        location ~* \.(js|css|xml|txt|jpg)$ {
            expires 14d;
            root /var/www/html/;
            access_log off;
        }

        location ~ /\.php$ {
            try_files $uri $uri/ /index.php;
            fastcgi_pass unix:/var/run/php5-fpm.sock;
            fastcgi_param SCRIPT_FILENAME $request_filename;
            fastcgi_index index.php;
            include /etc/nginx/conf/fastcgi_params;
        }
    }

    location ^~ /blog {
        root /var/www/html/;
        index index.php;
        try_files $uri $uri/ /index.php?$args;
        rewrite ^/blog/(.*)+$ /blog/index.php?$1;
        location ~* \.(js|css|xml|txt|jpg)$ {
            expires 14d;
            root /var/www/html/;
            access_log off;
        }
    }

    location ~ /\.php$ {
        try_files $uri $uri/ /index.php;
        fastcgi_pass unix:/var/run/php5-fpm.sock;
    }
}
```

```
        fastcgi_param SCRIPT_FILENAME $request_filename;
        fastcgi_index index.php;
        include /etc/nginx/conf/fastcgi_params;
    }

}
```

Lea Integración de blog de Wordpress con la aplicación de rieles usando nginx. en línea:
<https://riptutorial.com/es/nginx/topic/3891/integracion-de-blog-de-wordpress-con-la-aplicacion-de-rieles-usando-nginx->

Capítulo 5: proxy inverso nginx

Examples

proxy inverso simple

```
# Define which servers to include in the load balancing scheme.
# It's best to use the servers' private IPs for better performance and security.

upstream backend {

    ip_hash;
    server 10.10.10.10 slow_start=30s max_fails=3 fail_timeout=15s;
    server 10.10.10.12 slow_start=30s max_fails=3 fail_timeout=15s;

    # Activates the cache for connections to upstream servers.
    keepalive 20;
}

# This server accepts all traffic to port 80 and passes it to the upstream.
# Notice that the upstream name and the proxy_pass need to match.

server {
    listen 80;
    server_name example.com;

    location / {
        proxy_pass http://backend/;
    }
}
```

Lea proxy inverso nginx en línea: <https://riptutorial.com/es/nginx/topic/7431/proxy-inverso-nginx>

Capítulo 6: Redirecciones útiles

Examples

HTTPS Redirect

```
server {
    listen 80 default_server;
    listen [::]:80 default_server;
    server_name example.com www.example.com;
    return 307 https://$host$request_uri;
}
```

Un redireccionamiento 301 también es una alternativa, pero si se realiza un POST a un 301, muchos clientes volverán a enviar la solicitud como un GET, que casi siempre fallará en el servidor. Un 307 ordena el mismo tipo de solicitud.

Redirigir las solicitudes a otro servidor

```
server {
    server_name example.com;
    return 301 $scheme://example.net$request_uri;
}
```

Redirección de sitio móvil

Configuración de Nginx para detectar solicitudes del agente de usuario móvil y redirigirlas al sitio móvil.

```
location / {

    #mobile site handling as per user agent
    set $mobile_rewrite do_not_perform; // variable to store action. default set to not
    perform redirection to mobile site.

    if ($http_user_agent ~*
"(android|bb\d+|meego).+mobile|avantgo|bada\/|blackberry|blazer|compal|elaine|fennec|hiptop|iemo-
bile|ip-
|maemo|midp|mmp|mobile.+firefox|netfront|opera m(ob|in)i|palm(
os)?|phone|p(ixi|re)\\/|plucker|pocket|psp|series(4|6)0|symbian|treo|up\.(browser|link)|vodafone|wap|win-
ce|xda|xiino") {
        set $mobile_rewrite perform;
    }

    if ($http_user_agent ~* "^(1207|6310|6590|3gso|4thp|50[1-6]i|770s|802s|a-
wa|abac|ac(er|oo|s\-
)|ai(ko|rn)|al(av|ca|co)|amoi|an(ex|ny|yw)|aptu|ar(ch|go)|as(te|us)|attw|au(di|\-m|r |s
)|avan|be(ck|ll|nq)|bi(lb|rd)|bl(ac|az)|br(e|v)w|bumb|bw\-(n|u)|c55\\/|capi|ccwa|cdm\-
|cell|chtm|cldc|cmd\-|co(mp|nd)|craw|da(it|ll|ng)|dbte|dc\-s|devi|dica|dmob|do(c|p)o|ds(12|\-
d)|el(49|ai)|em(l2|ul)|er(ic|k0)|esl8|ez([4-7]0|os|wa|ze)|fetc|fly(\-|_)|g1 u|g560|gene|gf\-
5|g\-mo|go(\.w|od)|gr(ad|un)|haie|hcit|hd\-(m|p|t)|hei\-(hi|pt|ta)|hp( |i|ip)|hs\-c|ht(c|\-|
|_|a|g|p|s|t)|tp)|hu(aw|tc)|i\-(20|go|ma)|i230|iac( |)\-
```

```

|\\)|libro|idea|ig01|ikom|imlk|inno|ipaq|iris|ja(t|v)a|jbro|jemu|jigs|kddi|keji|kgt(
|\\)|klon|kpt |kwc\|-|kyo(c|k)|le(no|xi)|lg( g|l|u)|50|54|\-[a-w]|libw|lynx|m1\|-
w|m3ga|m50\|/ma(te|ui|x)|mc(01|21|ca)|m\|-
cr|me(rc|ri)|mi(o8|oa|ts)|mmef|mo(01|02|bi|de|do|t(\-| |o|v)|zz)|mt(50|p1|v )|mwbp|mywa|n10[0-
2]|n20[2-3]|n30(0|2)|n50(0|2|5)|n7(0(0|1)|10)|ne((c|m)\-
|on|tf|wf|wg|wt)|nok(6|i)|nzph|o2im|op(ti|wv)|oran|owg1|p800|pan(a|d|t)|pdxg|pg(13|\-([1-
8]|c))|phil|pire|pl(ay|uc)|pn\|-2|po(ck|rt|se)|prox|psio|pt\|-g|qa\|-a|qc(07|12|21|32|60|\-([2-
7]|i\-) |qtek|r380|r600|raks|rim9|ro(ve|zo)|s55\|/sa(ge|ma|mm|ms|ny|va)|sc(01|h\|-|oo|p\|-
)|sdk\|/se(c(\-|0|1)|47|mc|nd|ri)|sgh\|-|shar|sie(\-|m)|sk\|-
0|sl(45|id)|sm(al|ar|b3|it|t5)|so(ft|ny)|sp(01|h\|-|v\|-|v
)|sy(01|mb)|t2(18|50)|t6(00|10|18)|ta(gt|lk)|tcl\|-|tdg\|-|tel(i|m)|tim\|-|t\|-
mo|to(pl|sh)|ts(70|m\|-|m3|m5)|tx\|-9|up(\.b|g1|si)|utst|v400|v750|veri|vi(rg|te)|vk(40|5[0-
3]|\-v)|vm40|voda|vulc|vx(52|53|60|61|70|80|81|83|85|98)|w3c(\-| )|webc|whit|wi(g
|nc|nw)|wmlb|wonu|x700|yas\|-|your|zeto|zte\-)") {
    set $mobile_rewrite perform;
}

#google bot mobile handling
if ($http_user_agent ~* "(googlebot-mobile)") {
    set $mobile_rewrite perform;
}

if ($mobile_rewrite = perform) {
    proxy_pass http://www.mobile-domain.com:$port;
}
}

```

Ubicación HTTP en el servidor HTTPS

Servidor HTTPS con ubicación http:

```

server {
    listen 443;
    root /var/www/
    location / {
        ...
    }
    location /http {
        rewrite ^ http://$host$request_uri? permanent;
    }
}

```

El servidor HTTP redirige a HTTPS excepto una ubicación:

```

server {
    root /var/www/
    location / {
        rewrite ^ https://$host$request_uri? permanent;
    }
    location /http {
        ...
    }
}

```

No permitir solicitudes basadas en código de país o host

Utilice el [mapa Nginx](#) para analizar los campos y rechazar las solicitudes.

```
# Allowed hosts
map $http_host $name {
    hostnames;

    default      no;

    example.com  yes;
    *.example.com yes;
    example.org  yes;
    *.example.org yes;
    .example.net yes;
    wap.*        yes;
}

# Allowed countries
map $geoip_country_code $allowed_country {
    default no;
    country_code_1 yes;
    country_code_2 yes;
}
```

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

    # Disallow access based on hostname
    if ($name = no) {
        return 444;
    }

    # Disallow access based on GeoIP
    if ($allowed_country = no) {
        return 444;
    }

    ...
}
```

Lea Redirecciones útiles en línea: <https://riptutorial.com/es/nginx/topic/3631/redirecciones-utiles>

Capítulo 7: Usando nginx para proporcionar URLs de navegador limpias

Examples

Redireccionar vs proxy inverso

Las aplicaciones web creadas profesionalmente no exponen los detalles internos del entorno del servidor al usuario. Cuando realiza un pedido en su distribuidor, no ve (o tiene que escribir) <https://mydealer.com:8443/Dealerapp/entryPage.html> , sino solo mydealer.com, aunque el servidor de aplicaciones necesita todos los detalles dado en la URL más larga. Esto se puede lograr a través de:

```
if ($scheme = http) {
    return 301 https://$server_name$request_uri;
}
```

Esto hace una **redirección** . Incluso si el cliente no solicitó una conexión segura, el navegador se redirige a la vez. En países que tienen leyes estrictas de privacidad de datos, incluso podría estar obligado a hacer esto en cualquier sitio comercial. Se toma la forma de redireccionar porque aquí el navegador necesita saber acerca de la conexión segura, de lo contrario no negociaría con el servidor para hacerlo seguro.

```
location /app/ {
    proxy_pass https://mydealer.com:8443/Dealerapp/entryPage.html;
}
```

Este es un **proxy inverso** . Es transparente para el navegador. Por lo tanto, puede ocultarse completamente del usuario final si tiene uno o más servidores de aplicaciones, en qué puertos escuchan o cómo se nombran sus aplicaciones. De lo contrario, si su centro de datos necesita mover la aplicación a otro servidor que escucha en 8543, todos los marcadores de sus clientes serían inválidos. Este ejemplo también dirige al usuario a su página de entrada. Esto podría omitirse si nombra la página de entrada index.html. Pero quizás tenga varias páginas de entrada dentro de una aplicación que el usuario quiera marcar, por lo que es más flexible si no está obligado a index.html.

Publiqué la URL de la empresa con / app. Esto podría omitirse si su dominio solo sirve esta aplicación. En caso de que además de la aplicación, también haya algún contenido estático, como la descripción de su empresa, es posible que desee tener una URL simple para eso.

Este proxy solo funciona para la página de entrada. Por lo general, necesito dos esquemas de URL más, uno para el contenido estático de la aplicación web, como JavaScript, CSS e imágenes. Pongo todos esos en una estructura de carpetas debajo de una carpeta llamada serverapp, y escribo el siguiente proxy:

```
location /app/serverapp/ {
    proxy_pass https://mydealer.com:8443/Dealerapp/serverapp/;
}
```

Y otro camino para los servicios REST; aquí la ruta URL del resto no coincide con una carpeta, sino con la ruta de un servicio REST de jax-rs:

```
location /rest/ {
    proxy_pass https://mydealer.com:8443/Dealerapp/rest/;
}
```

Se necesita dar un paso más, que rara vez se menciona en ninguna parte. El servidor de aplicaciones se ejecuta bajo la ruta URL / Dealerapp, por lo que emitirá una cookie de sesión con la ruta de propiedad = Dealerapp. El navegador no conoce esta ruta y, debido a su Política del mismo origen, ignorará la cookie. Podríamos convencerlo mediante el uso compartido de recursos de origen cruzado para permitir esto, pero probablemente sea más fácil modificar la ruta de las cookies configurando la ruta en /, escribiendo algo como

```
<session-config>
  <session-timeout>720</session-timeout>
  <cookie-config>
    <name>SZSESSION</name>
    <path>/</path>
    <http-only>true</http-only>
    <secure>true</secure>
  </cookie-config>
</session-config>
```

a nuestro web.xml.

Lea Usando nginx para proporcionar URLs de navegador limpias en línea:

<https://riptutorial.com/es/nginx/topic/6270/usando-nginx-para-proporcionar-urls-de-navegador-limpias>

Creditos

S. No	Capítulos	Contributors
1	Empezando con nginx	Bbak , Community , I Am Batman , James , Marek Skiba , Mark Stosberg , Neo , Przemysław Jagielski , rajarshig , RamenChef , RationalDev , theDrifter , treecoder , Xevaquor
2	Configuraciones de Nginx	Bbak , James , Pablo Fernandez , RationalDev
3	Explotación florestal	Gustav , RationalDev , timbo
4	Integración de blog de Wordpress con la aplicación de rieles usando nginx.	Abid Iqbal
5	proxy inverso nginx	smart-developer
6	Redirecciones útiles	Aleksey Deryagin , Alexandre Maciel , Alexey Ten , Gaurav Kumar , Joshua DeWald , Justin W. , Keelan , Muaaz Rafi , smart-developer , timbo
7	Usando nginx para proporcionar URLs de navegador limpias	TAM