



EBook Gratuito

APPENDIMENTO

nginx

Free unaffiliated eBook created from
Stack Overflow contributors.

#nginx

Sommario

Di.....	1
Capitolo 1: Iniziare con nginx.....	2
Osservazioni.....	2
Versioni.....	2
Examples.....	2
Installazione e configurazione.....	2
Installazione di Nginx su distribuzioni basate su Debian e Debian come Ubuntu.....	3
Riavvia NGINX.....	3
Esempio di Ubuntu.....	3
Ricarica il file di configurazione NGINX.....	4
Esempio di Ubuntu 14.04.....	4
Esempio di Ubuntu 16.04.....	4
Spegni NGINX.....	4
Nginx dentro.....	4
Verifica se le tue modifiche in nginx.config sono valide.....	5
Capitolo 2: Configurazioni Nginx.....	6
Examples.....	6
Struttura del file di configurazione.....	6
Direttive semplici.....	6
Direttiva Block.....	6
Contesto.....	6
Commento.....	6
Test del file di configurazione NGINX.....	6
Specifica il file di configurazione da caricare.....	7
modifiche di configurazione senza riavvio.....	7
Registrazione a Syslog, incl. mappatura dei codici HTTP alle gravità di Syslog.....	7
Limita i metodi di richiesta.....	8
Capitolo 3: Integrazione del blog Wordpress con l'applicazione rails utilizzando nginx.....	9
Examples.....	9
blocco del server nginx.....	9

Capitolo 4: Registrazione	11
Examples.....	11
Esempio di base.....	11
Sintassi	11
Riapri i file di registro.....	11
Evita la registrazione di favicon.ico e robots.txt.....	11
Capitolo 5: Reindirizzamenti utili	12
Examples.....	12
Reindirizzamento HTTPS.....	12
Reindirizza le richieste a un altro server.....	12
Reindirizzamento dei siti mobili.....	12
Posizione HTTP sul server HTTPS.....	13
Non consentire le richieste in base al codice host o paese.....	13
Capitolo 6: reverse proxy nginx	15
Examples.....	15
semplice proxy inverso.....	15
Capitolo 7: Utilizzare nginx per fornire URL del browser puliti	16
Examples.....	16
Reindirizzamento vs reverse proxy.....	16
Titoli di coda	18

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

Capitolo 1: Iniziare con nginx

Osservazioni

NGINX è pronunciato come "engine x" ed è comunemente usato come server ad alte prestazioni per i protocolli HTTP, HTTPS, SMTP, POP3 e IMAP. Può essere utilizzato come server proxy inverso, cache HTTP o bilanciamento del carico.

È un progetto open source con la fonte disponibile [qui](#).

Versioni

Versione	Data di rilascio originale	Ultima versione	Stato	Data di rilascio
0.5	2006-12-04	0.5.38	eredità	2009-09-14
0.6	2007-06-14	0.6.39	eredità	2009-09-14
0.7	2008-05-19	0.7.69	eredità	2011-07-19
0.8	2009-06-02	0.8.55	eredità	2011-07-19
1.0	2011-04-12	1.0.15	eredità	2012-04-12
1.2	2012-04-23	1.2.9	eredità	2013/05/13
1.4	2013/04/24	1.4.7	eredità	2014/03/18
1.6	2014/04/24	1.6.3	eredità	2015/04/07
1.8	2015/04/21	1.8.1	eredità	2016/01/26
1.9	2015/04/28	1.9.15	eredità	2016/04/19
1.10	2016/04/26	1.10.3	Stabile	2016/05/31
1.11	2016/05/24	1.11.9	Linea principale	2016/07/26

Examples

Installazione e configurazione

Nginx è un server Web utilizzato per servire le richieste HTTP su Internet.

Nginx è disponibile su Linux, Windows e altri sistemi operativi come download diretto e può anche essere creato dalla sorgente. Per istruzioni dettagliate, consultare il [riferimento ufficiale Nginx](#).

ubuntu / debian

La versione stabile di nginx è disponibile nel repository ufficiale, può essere installata usando

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

Installa e configura i file di avvio del sistema, ma se hai bisogno dell'ultima versione, potresti dover aggiungere un ppa ufficiale.

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

le istruzioni sopra installeranno l'ultima edizione stabile.

Installazione di Nginx su distribuzioni basate su Debian e Debian come Ubuntu

Esegui il comando qui sotto per installare nginx.

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

Per impostazione predefinita, Nginx si avvia automaticamente quando viene installato. È possibile accedere alla pagina di destinazione Nginx predefinita per confermare che il software sta funzionando correttamente visitando il nome di dominio del server o l'indirizzo IP pubblico nel browser web.

ma se hai bisogno dell'ultima versione, potresti dover aggiungere un ppa ufficiale.

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

Riavvia NGINX

Come utente root:

```
nginx -s restart
```

Esempio di Ubuntu

```
sudo service nginx restart
```

Ricarica il file di configurazione NGINX

Come utente root:

```
sudo nginx -s reload
```

Esempio di Ubuntu 14.04

```
sudo service nginx reload
```

Esempio di Ubuntu 16.04

```
sudo systemctl reload nginx
```

Prima di ricaricare, è una buona idea controllare la configurazione per gli errori di sintassi:

```
sudo nginx -t
```

O

```
sudo service nginx configtest
```

Spegni NGINX

Esegui come utente root.

Arresto rapido:

```
nginx -s stop
```

Arresto grazioso:

```
nginx -s quit
```

Nginx dentro

Uno dei più grandi richiami di Nginx è la differenza nel modo in cui funziona internamente rispetto agli altri server popolari, specialmente Apache.

I server sono programmi occupati in quanto devono servire le richieste di più client. Più richieste un server può servire con successo al secondo, meglio è.

Nginx funziona su un paradigma di concorrenza noto come IO asincrono.

In un server convenzionale, un thread è dedicato a una richiesta. Ciò significa che, una volta che

un thread prende una richiesta, non è effettivamente disponibile per altre richieste. Ma in realtà, un thread potrebbe fare molto meglio accettando un sacco di richieste e servendole contemporaneamente. L'IO asincrono è ciò che abilita questo.

Nginx, quindi con la sua architettura IO asincrona, può servire molte richieste all'interno di un thread.

Un altro aspetto positivo di Nginx è il suo ingombro di risorse relativamente più snello. Rispetto ad Apache, Nginx è meno pesante delle risorse, e questo lo rende adatto ai server cloud che tendono a non essere molto potenti.

Esistono sicuramente altri server Async IO, ma Nginx è il più supportato tra tutti in termini di plugin (ovvero i moduli Nginx).

Verifica se le tue modifiche in nginx.config sono valide

Esempio di Ubuntu 14.04

```
sudo nginx -t
```

Leggi Iniziare con nginx online: <https://riptutorial.com/it/nginx/topic/1121/iniziare-con-nginx>

Capitolo 2: Configurazioni Nginx

Examples

Struttura del file di configurazione

nginx è composto da moduli controllati da direttive specificate nel file di configurazione.

Direttive semplici

Una direttiva semplice è composta da nome e parametri separati da spazi e termina con un punto e virgola (;).

Direttiva Block

Una direttiva di blocco ha la stessa struttura di una direttiva semplice, ma al posto del punto e virgola termina con una serie di istruzioni aggiuntive racchiuse tra parentesi graffe ({e}).

Contesto

Se una direttiva di blocco può avere altre direttive all'interno di parentesi, viene chiamata contesto (esempi: eventi, http, server e posizione).

Le direttive inserite nel file di configurazione al di fuori di qualsiasi contesto sono considerate nel contesto principale. Gli eventi e le direttive http si trovano nel contesto principale, il server in http e la posizione nel server.

Commento

Il resto di una riga dopo il segno # è considerato un commento.

Test del file di configurazione NGINX

È possibile verificare la presenza di errori di sintassi e file di riferimento in un file di configurazione NGINX prima di eseguirlo con:

```
nginx -t
```

In alternativa è possibile eseguire lo script di servizio

```
service nginx configtest
```

Mentre entrambi questi comandi ti diranno se la tua nuova configurazione di nginx è ok [senza uccidere l'istanza corrente]. Configtest usa il servizio in esecuzione e ti dice se passa o fallisce il controllo mentre nginx -t non solo controlla la configurazione ma stampa qualsiasi informazione, avviso e messaggi di errore.

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

Specifica il file di configurazione da caricare

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

Avvia NGINX con un file di configurazione esplicito.

modifiche di configurazione senza riavvio

```
nginx -s reload
```

Registrazione a Syslog, incl. mappatura dei codici HTTP alle gravità di Syslog.

Incolla questo snippet da qualche parte nel blocco `http {}`; o posizionarlo nel proprio file nella cartella `/etc/nginx/conf.d/`. Vedi anche i [documenti ufficiali](#) per accedere a 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;
```

Questo esempio presuppone che rsyslog (o simile) stia ascoltando Socket

`/run/systemd/journal/syslog` - come impostazione predefinita su Debian 8 quando journalald ha attivato [ForwardToSyslog](#). Usando questo socket, si ignora il journal. Se quel socket non è disponibile, prova invece a `/dev/log`.

Sentiti libero di usare un'altra struttura invece di local2. Puoi anche modificare il [log_format](#) in

base alle tue esigenze.

Limita i metodi di richiesta

Un sito web abituale richiede solo 3 metodi HTTP: GET , HEAD e POST . Blocca tutti gli altri metodi usando `limit_except` :

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

Leggi Configurazioni Nginx online: <https://riptutorial.com/it/nginx/topic/2550/configurazioni-nginx>

Capitolo 3: Integrazione del blog Wordpress con l'applicazione rails utilizzando nginx

Examples

blocco del server nginx

Dopo aver configurato e configurato le impostazioni di php5-fpm e wordpress, è possibile configurare il file `/etc/nginx/conf/nginx.conf` come di seguito.

È necessario definire i blocchi di posizione all'interno del blocco del server e riscrivere l'url lì come definito.

```
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;
    }

}
```

Leggi [Integrazione del blog Wordpress con l'applicazione rails utilizzando nginx online](https://riptutorial.com/it/nginx/topic/3891/integrazione-del-blog-wordpress-con-l-applicazione-rails-utilizzando-nginx):

<https://riptutorial.com/it/nginx/topic/3891/integrazione-del-blog-wordpress-con-l-applicazione-rails-utilizzando-nginx>

Capitolo 4: Registrazione

Examples

Esempio di base

Sintassi

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

Usarli insieme

```
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;
```

Riapri i file di registro

Come utente root esegui:

```
nginx -s reopen
```

Evita la registrazione di favicon.ico e robots.txt

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

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

Leggi Registrazione online: <https://riptutorial.com/it/nginx/topic/2551/registrazione>

Capitolo 5: Reindirizzamenti utili

Examples

Reindirizzamento HTTPS

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

Un reindirizzamento 301 è anche un'alternativa, ma se un POST viene eseguito su un 301, molti client invieranno nuovamente la richiesta come GET - che quasi sempre fallirà sul lato server. Un 307 richiede lo stesso tipo di richiesta.

Reindirizza le richieste a un altro server

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

Reindirizzamento dei siti mobili

Configurazione di Nginx per rilevare la richiesta da un utente-agente mobile e reindirizzarla sul sito mobile.

```
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|iemoile|i
|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|hcrit|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|\\|(k|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;
}
}
```

Posizione HTTP sul server HTTPS

Server HTTPS con posizione http:

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

Il server HTTP reindirizza a HTTPS tranne una posizione:

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

Non consentire le richieste in base al codice host o paese

Usa la [mappa Nginx](#) per analizzare i campi e rifiutare le richieste.

```
# 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;
    }

    ...
}
```

Leggi Reindirizzamenti utili online: <https://riptutorial.com/it/nginx/topic/3631/reindirizzamenti-utili>

Capitolo 6: reverse proxy nginx

Examples

semplice proxy inverso

```
# 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/;
    }
}
```

Leggi reverse proxy nginx online: <https://riptutorial.com/it/nginx/topic/7431/reverse-proxy-nginx>

Capitolo 7: Utilizzare nginx per fornire URL del browser puliti

Examples

Reindirizzamento vs reverse proxy

Le applicazioni Web create professionalmente non espongono all'utente i dettagli interni dell'ambiente server. Quando effettui un ordine presso il tuo rivenditore, non vedi (o devi digitare) <https://mydealer.com:8443/Dealerapp/entryPage.html> , ma solo mydealer.com, anche se il server dell'app ha bisogno di tutti i dettagli dato nell'URL più lungo. Questo può essere ottenuto tramite:

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

Questo fa un **reindirizzamento** . Anche se il client non ha richiesto una connessione sicura, il browser viene immediatamente reindirizzato ad esso. Nei paesi che applicano severe leggi sulla privacy dei dati, potresti persino essere obbligato a farlo per qualsiasi sito commerciale. La modalità di reindirizzamento è presa perché qui il browser deve conoscere la connessione sicura, altrimenti non negozierebbe con il server per renderlo sicuro.

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

Questo è un **proxy inverso** . È trasparente per il browser. In questo modo puoi nasconderti completamente dall'utente finale, a prescindere dal fatto che tu abbia uno o più server di app, su quali porte ascoltano o in che modo vengono denominate le loro applicazioni. Altrimenti, se il tuo datacenter deve spostare l'app su un altro server che è in ascolto su 8543, tutti i segnalibri dei tuoi client diventerebbero non validi. Questo esempio indirizza anche l'utente alla tua pagina di entrata. Questo potrebbe essere omesso se si nomina la pagina di entrata index.html. Ma forse hai diverse pagine di accesso all'interno di un'app che l'utente potrebbe voler aggiungere ai segnalibri, quindi sei più flessibile se non sei legato a index.html.

Ho postfixato l'URL della società con / app. Questo potrebbe essere omesso se il tuo dominio serve solo questa app. Nel caso in cui oltre all'app, ci sia anche del contenuto statico, come la descrizione della tua azienda, potresti voler avere l'URL semplice per quello.

Questo proxy funziona solo per la pagina di entrata. Di solito ho bisogno di altri due schemi di URL, uno per il contenuto statico dell'app Web, come JavaScript, CSS e immagini. Metto tutti quelli in una struttura di cartelle sotto una cartella chiamata serverapp, e scrivo il seguente proxy:

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

```
}
```

E un altro percorso per i servizi REST; qui il resto del percorso URL non corrisponde a una cartella, ma a un percorso di un servizio REST jax-rs:

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

Un altro passo, raramente menzionato ovunque, deve essere preso. Il server dell'app viene eseguito sotto il percorso URL / Dealerapp, quindi emetterà un cookie di sessione con percorso proprietà = Dealerapp. Il browser non conosce questo percorso e, a causa della sua stessa politica di origine, ignorerà il cookie. Potremmo convincerlo tramite Cross-Origin Resource Sharing per consentirlo, ma probabilmente è più semplice modificare il percorso del cookie impostando il percorso su /, scrivendo qualcosa come

```
<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>
```

al nostro web.xml.

Leggi Utilizzare nginx per fornire URL del browser puliti online:

<https://riptutorial.com/it/nginx/topic/6270/utilizzare-nginx-per-fornire-url-del-browser-puliti>

Titoli di coda

S. No	Capitoli	Contributors
1	Iniziare con nginx	Bbak , Community , I Am Batman , James , Marek Skiba , Mark Stosberg , Neo , Przemysław Jagielski , rajarshig , RamenChef , RationalDev , theDrifter , treecoder , Xevaquor
2	Configurazioni Nginx	Bbak , James , Pablo Fernandez , RationalDev
3	Integrazione del blog Wordpress con l'applicazione rails utilizzando nginx	Abid Iqbal
4	Registrazione	Gustav , RationalDev , timbo
5	Reindirizzamenti utili	Aleksy Deryagin , Alexandre Maciel , Alexey Ten , Gaurav Kumar , Joshua DeWald , Justin W. , Keelan , Muaaz Rafi , smart-developer , timbo
6	reverse proxy nginx	smart-developer
7	Utilizzare nginx per fornire URL del browser puliti	TAM