

XAMPP: SSL/HTTPS für lokale Projekte

Tutorial von [Stefan Trost](#) | 09.08.2017

In diesem Tutorial möchte ich euch zeigen, wie ihr eure XAMPP-Konfiguration so einrichten könnt, dass ihr eure lokalen PHP-Projekte auch über SSL bzw. im Browser über eine HTTPS-URL erreichen könnt. Praktisch ist dies zum Beispiel in dem Fall, dass auch eure Online-Website über HTTPS läuft und ihr so in beiden Fällen dieselbe Linkstruktur nutzen könnt.

In meinem [Tutorial zum Thema virtuelle Domains](#) hatte ich schon die Grundlagen erklärt, wie man mit XAMPP und Apache Domains definieren kann, die dann auf ein lokales Verzeichnis in der XAMPP-Installation zeigen können. Nach dieser Einrichtung sind alle Domains nur über HTTP erreichbar. Um auch HTTPS zu erlauben, sind noch die folgenden Schritte notwendig.

Zertifikat erstellen

Zunächst müssen wir uns im ersten Schritt ein eigenes SSL-Zertifikat erstellen. Zum Glück hält XAMPP bereits ein kleines Skript dafür bereit, das wir folgendermaßen öffnen können:

1. Das Skript finden wir im Apache-Ordner der XAMPP-Installation, bei einer Standard-Installation unter Windows ist das der Pfad: **C:\xampp\apache**
2. In diesem Ordner finden wir die Datei **makecert.bat**, mit einem Doppelklick auf die Datei können wir das Skript starten.

Nach dem Starten des Scripts öffnet sich die Kommandozeile, wo wir nur noch den Anweisungen folgen brauchen. Als erstes müssen wir ein Passwort für unseren Key eingeben, das sollte so aussehen.

```
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to 'privkey.pem'
Enter PEM pass phrase:
```

Hier können wir ein beliebiges Passwort verwenden und anschließend wiederholen. Es folgen weitere Fragen wie beispielsweise Land (Country Name) oder Unternehmen (Organization Name). Hier können wir im Prinzip eingeben, was wir möchten oder einen Punkt eintippen, wenn wir das Feld freilassen möchten. Diese Angaben werden später in unserem Zertifikat verwendet. Als "Common Name" geben wir den Namen unserer URL ein. Zuletzt müssen wir das Passwort vom Beginn noch einmal wiederholen, dann wird das Zertifikat erstellt.

Das ganze sieht dann beispielsweise so aus:

```
Generating a 2048 bit RSA private key
.....+++
```

.....+++

writing new private key to 'privkey.pem'

Enter PEM pass phrase:

Verifying - Enter PEM pass phrase:

You are about to be asked to enter information that will be incorporated into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

Country Name (2 letter code) [AU]:DE

State or Province Name (full name) [Some-State]:NRW

Locality Name (eg, city) []:BOT

Organization Name (eg, company) [Internet Widgits Pty Ltd]:STT

Organizational Unit Name (eg, section) []:WEB

Common Name (e.g. server FQDN or YOUR name) []:localhost

Email Address []:local@example.com

Please enter the following 'extra' attributes

to be sent with your certificate request

A challenge password []:.

An optional company name []:.

Enter pass phrase for privkey.pem:

writing RSA key

Signature ok

subject=/C=DE/ST=NRW/L=BOT/O=STT/OU=WEB/CN=localhost/emailAddress=local@example.com

Getting Private key

1 Datei(en) verschoben.

1 Datei(en) verschoben.

Das Zertifikat wurde erstellt.

The certificate was provided.

Drücken Sie eine beliebige Taste . . .

Nun ist unser Zertifikat erstellt und dieser Schritt ist fertig. Das fertige Zertifikat wurde in den Ordnern C:\xampp\apache\conf\ssl.crt und C:\xampp\apache\conf\ssl.key abgelegt, sofern wir XAMPP in C:\xampp installiert haben.

VirtualHost definieren

Im nächsten Schritt müssen wir für jede Domain einen neuen VirtualHost einrichten, ähnlich wie wir es auch schon in dem ersten Tutorial gemacht haben.

An dieser Stelle ist wichtig zu wissen, dass HTTP über Port 80 läuft, SSL dagegen aber über Port 443. Im Folgenden definieren wir die URL example.com, die sowohl über http://www.example.com als auch über https://www.example.com erreichbar sein soll. Der erste Block <VirtualHost *:80> gilt für alle Anfragen über Port 80 und sieht in etwa so aus, wie die Blöcke aus dem ersten [Tutorial](#). Unter DocumentRoot können wir das Stammverzeichnis angeben.

```
# HTTP
```

```
<VirtualHost *:80>
  DocumentRoot "C:/xampp/htdocs/myproject"
  ServerName example.com
  ServerAlias www.example.com
</VirtualHost>
```

```
# HTTPS
```

```
<VirtualHost *:443>
  DocumentRoot "C:/xampp/htdocs/myproject"
  ServerName example.com
  ServerAlias www.example.com
  SSLEngine on
  SSLCertificateFile "conf/ssl.crt/server.crt"
  SSLCertificateKeyFile "conf/ssl.key/server.key"
</VirtualHost>
```

Die HTTPS-Anfragen werden über den zweiten Block <VirtualHost *:443> definiert. Die Angaben zum DocumentRoot-Verzeichnis, ServerName und ServerAlias sind identisch. Zusätzlich müssen wir aber noch Angaben zu unserem SSL-Zertifikat und Key machen. Damit sagen wir dem Server, wo er die entsprechenden Dateien findet. Wie man sieht, sind es die Ordner, in denen gerade unser Zertifikat gespeichert wurde.

In dem Beispiel hier haben wir nur eine URL definiert, die über HTTP und HTTPS erreichbar sein soll. Jede weitere Domain bzw. URL muss genauso definiert werden, wir können beliebig viele Blöcke dieser Art untereinander schreiben.

Apache neu starten

Damit unsere Änderungen aktiv werden, müssen wir jetzt Apache neu starten. Dafür öffnen wir das XAMPP Control Panel und klicken hinter dem Modul "Apache" auf "Stop" und anschließend wieder auf "Start", damit der Server neu mit der neuen Konfiguration startet.

Ausnahmen hinzufügen

Nun sind wir mit unserer Konfiguration fertig. Wenn wir jedoch den Browser aufrufen, erhalten wir unter Umständen (je nach Browser und Browsereinstellungen) die Fehlermeldung "Diese Verbindung ist nicht sicher" mit dem Fehlercode SEC_ERROR_UNKNOWN_ISSUER. Dazu liefert der Firefox-Browser den abschreckenden Text "Der Inhaber von ... hat die Website nicht richtig konfiguriert. Firefox hat keine Verbindung mit der Website aufgebaut, um ihre Informationen vor Diebstahl zu stützen". Kurz gesagt: Wir können unsere Website gar nicht aufrufen.

Der Grund dafür ist, dass wir das Zertifikat selbst erstellt haben und Firefox von sich aus nur Zertifikaten von offiziellen Zertifizierungsstellen traut. Für unsere Zwecke wäre das natürlich übertrieben, weswegen wir getrost auf den Button "Ausnahme hinzufügen..." klicken können. In dem sich nun öffnenden Dialog sollten wir die Option "Diese Ausnahme dauerhaft speichern" aktivieren, sonst werden wir bei jedem Aufruf erneut gefragt. Anschließend können wir auf "Sicherheits-Ausnahmeregel bestätigen" klicken und sollten nun unsere Website über die HTTPS-URL sehen können.

<https://gist.github.com/nguyenanhtu/33aa7ffb6c36fdc110ea8624eeb51e69>

Virtuelle Domains für Apache/XAMPP erstellen

Tutorial von [Stefan Trost](#) |

Die Entwicklung einer Website geht meistens folgenden Weg: Zunächst testen wir unsere Internetseite auf unserem lokalen Rechner, wo wir zum Beispiel XAMPP benutzen. Danach soll die Seite auf unseren Server hochgeladen werden und unter einer Adresse wie `example.com` verfügbar sein. Vorher jedoch, beim testen, greifen wir über eine Adresse wie `http://localhost/seiten/xyz` auf unsere Seite über den Browser zu.

Linkstruktur auf Server und lokalem Rechner

Aus dieser Tatsache erwachsen jedoch einige Probleme: Die Linkstruktur auf dem Server und dem lokalen Rechner ist eine andere, die Links funktionieren nur, wenn sie relativ gesetzt sind. Wenn wir jedoch einen Link verwenden, in dem direkt unsere spätere Domain vorkommt, wird diese zwar später auf dem Server funktionieren, nicht aber beim testen.

Dieses Problem kann an vielen Stellen auftreten, denn nicht immer kann man relative Links verwenden. Nehmen wir zum Beispiel den Fall, dass wir Bilder und Script-Dateien auf einen anderen Server ausgelagert haben. Hier müssten wir die direkte Domain angeben, zum testen ist dies mehr als unvorteilhaft.

Virtuelle Domains

Zum Glück gibt es eine Lösung für dieses Problem: Virtuelle Domains, manchmal auch virtuelle Hosts oder Virtual Hosts genannt. Virtuelle Hosts ermöglichen es uns, auf unserem lokalen Rechner Domains verfügbar zu machen, so dass wir, wenn wir eine dieser definierten Domains aufrufen, in einem von uns definierten Verzeichnis landen.

Nicht anders funktioniert das übrigens auch mit `http://localhost`. Auch `localhost` ist auf diese Weise definiert worden, so dass wir in dem entsprechenden Verzeichnis landen.

Virtuelle Domains definieren

Neben `localhost` möchten wir für unser Beispiel nun auch noch die Domain `example.com` und die Subdomain `s.example.com` definieren. Dazu müssen wir in drei Dateien Änderungen vornehmen.

C:\WINDOWS\system32\drivers\etc\hosts

Zunächst müssen wir die Host-Datei von Windows anpassen. Diese heißt `hosts` (ohne Dateiendung) und wir finden sie im Verzeichnis `C:\WINDOWS\system32\drivers\etc` der Windows Installation. In dieser Datei werden alle Zuordnungen von IP-Adressen zu Hostnamen angegeben. Die IP-Adresse `127.0.0.1` entspricht unserem eigenen Rechner.

```
127.0.0.1    localhost
127.0.0.1    example.com www.example.com
127.0.0.1    s.example.com
```

In der ersten Zeile finden wir hier schon unseren localhost. Den lassen wir stehen und schreiben darunter die Domains, die wir benötigen. Im Beispiel sind das example.com, www.example.com und unsere Subdomain s.example.com.

C:\xampp\apache\conf\http.conf

Als nächstes müssen wir noch zwei Dateien in unserer XAMPP Konfiguration anpassen. Wir finden die Datei http.conf in dem Ordner, in dem XAMPP installiert haben unter apache\conf\.

Diese Datei ist die Server-Konfiguration von Apache. Zunächst sollten wir nachschauen, ob die Datei schon folgende Zeile enthält:

```
NameVirtualHost 127.0.0.1
```

Wenn diese Zeile nicht in der http.conf steht oder auskommentiert ist, schreiben wir sie einfach ganz nach unten in die Datei oder entfernen die Auskommentierung.

Danach schauen wir nach folgender Zeile:

```
# Virtual hosts
Include "conf/extra/httpd-vhosts.conf"
```

Auch die Zeile mit "Include" ist unter Umständen auskommentiert oder nicht enthalten. Wenn dies der Fall ist, ändern wir die Zeile so, wie sie oben steht oder schreiben die Zeile noch in die http.conf hinzu.

C:\xampp\apache\conf\extra\httpd-vhosts.conf

Zuletzt müssen wir noch die Datei httpd-vhosts.conf anpassen, die wir in unserer XAMPP Installation unter \apache\conf\extra ein Verzeichnis unter unserer vorherigen Datei finden.

In dieser Datei finden wir unsere virtuellen Hosts, unter anderem sollte hier auch schon localhost eingetragen sein. Genauso, wie auch localhost eingetragen ist, können wir unsere zusätzlichen Domains eintragen und das entsprechende Verzeichnis für die Domain angeben:

```
<VirtualHost www.example.com>
    ServerAdmin webmaster@localhost
    DocumentRoot "c:/xampp/htdocs/seiten/xyz"
    ServerName example.com
    ServerAlias example.com
</VirtualHost>

<VirtualHost s.example.com>
    ServerAdmin webmaster@localhost
    DocumentRoot "c:/xampp/htdocs/seiten/xyzs"
    ServerName s.example.com
    ServerAlias s.example.com
</VirtualHost>
```

Im Beispiel sehen wir das für die Domain www.example.com und die Subdomain s.example.com, die auf die unter DocumentRoot angegebenen Verzeichnisse zeigen sollen. Hier kann man einfach seine Verzeichnisse und Domains eintragen, die man anlegen möchte.

XAMPP neu Starten

Damit die Änderungen wirksam werden, müssen wir XAMPP neu starten. Dazu rufen wir das XAMPP Control Panel auf, das wir zum Beispiel über das Startmenü oder die Desktop Verknüpfung aufrufen können. Hier finden wir den Eintrag "Apache" "Running", wo wir einmal auf "Stop" klicken und danach wieder auf "Start".

Nachdem der Server neu gestartet wurde, sollten unsere neu angelegten Domains verfügbar sein.

HTTPS URLs

Die hier beschriebenen Schritte ermöglichen einen Aufruf der definierten URLs über das HTTP-Protokoll, also zum Beispiel über <http://www.example.com>. Wenn wir unsere Domain auch über HTTPS/SSL, also über <https://www.example.com> aufrufen möchten, sind weitere Schritte erforderlich. Welche das sind, erfahrt ihr in meinem [Tutorial SSL/HTTPS für lokale Projekte einrichten](#). Dort erkläre ich auch, wie ihr ein eigenes SSL-Zertifikat für eure XAMPP-Server-Konfiguration erstellen könnt.

Probleme

Ich möchte an dieser Stelle noch auf ein Problem hinweisen: Diese Lösung ist am besten für Rechner geeignet, die nur zu Testzwecken benutzt werden und von denen die angelegten Domains nicht im Internet aufgerufen werden müssen.

Denn jedesmal, wenn wir eine unserer angelegten Domains aufrufen, werden wir automatisch auf unsere lokale Version der Seite umgeleitet, wir kommen nicht mehr auf die Version im Internet. Um diese zu erreichen, müssen wir unseren Hosteintrag in der host-Datei auskommentieren und den Browser neu starten:

```
#127.0.0.1 example.com
```

Das ist natürlich etwas umständlich und bisher ist mir auch keine bessere Lösung für diese Problematik bekannt. Sollte jemand etwas wissen, bitte posten.

