

Übungsblatt 02

Thomas Samy Dafir, Lex Winandy

Aufgabe 1

Gruppe: Installieren Sie ein tool um das Apache log auszuwerten (Vorschlag: webalizer, analog oder awstats). Die Ergebnisse sollen über einen Link von Ihrer Homepage sichtbar sein.

Wir verwenden 'Webalizer'

Schritte, um webalizer für unsere Seite zu verwenden:

1. `sudo apt-get install webalizer:`
Richtet webalizer für `/var/www/webalizer` ein.
2. `sudo mv /var/www/webalizer /var/www/html/`
3. Webalizer konfigurieren:
`sudo nano /etc/webalizer/webalizer.conf`
4. Settings:Logfile angeben: `LogFile /var/log/apache2/access.log`
Output directory auf ordner in unserem web root directory → `html`
Output und Anzeige:
`OutputDir /var/www/html/webalizer`
5. Konfig testen (webalizer starten):
`sudo webalizer`

Quelle: <http://linuxpitstop.com/install-and-configure-webalizer-on-ubuntu/>

Aufgabe 2

Gruppe: Konfigurieren Sie im Webserver einen internen redirect und einen redirect auf eine externe Seite. Wie gehen Sie vor? Was passiert auf HTTP-Ebene? Hinweis: mod rewrite

mod_rewrite:

- `sudo a2enmod rewrite`
- `sudo apache2 restart`
- Nach Änderungen in der Config muss apache ebenfalls neu gestartet werden.

Intern:

Möglichkeit 1: Interner redirect. Liefert im Beispiel 'index.html' statt 'intredir.html'. Das ist für den User hier aber nicht erkennbar. Die Adresszeile zeigt 'intredir.html', sogar in der Netzwerkansicht der Developer Tools ist 'intredir.html' angegeben. Dieses Verhalten wird durch das Flag PT (passthrough) erreicht

```
RewriteEngine on
RewriteRule    "^/intredir\.html$"    "/index.html" [PT]
```

Möglichkeit 2:

Hier wird die URL tatsächlich umgeschrieben. Der Client erhält 302 Found für die ursprüngliche URL zurück, im Location Header aber die Zieladresse des Redirect. Danach fragt der Client diese Adresse an und erhält das Dokument (200 OK). Flag R (redirect)

```
RewriteEngine on
RewriteRule    "^/extredir\.html$"    "/index.html" [R]
```

Extern:

Hier wird die URL auch umgeschrieben. Der Client erhält auf seine Anfrage 302 Found zurück. Zusätzlich erhält er im Location Header das Ziel des Redirect. Diese wird dann mit GET angefragt und der Server (hier unter uni-salzburg.at) gibt (hier nach einem zusätzlichen redirect auf seine index.php) die Seite mit 200 OK zurück. Flags: R,L (redirect, last:stoppt, wenn diese Regel passt).

```
RewriteEngine on
RewriteRule    "^/othersite\.html$"    "https://uni-salzburg.at" [R,L]
```

Quelle: Apache Dokumentation

Aufgabe 3

Erstellen Sie weiters eine HTML-Seite, die den Browser anweist, auf eine andere Seite zu gehen (Hinweis: meta http-equiv="Refresh"). Was passiert auf HTTP-Ebene?

Folgendes meta-tag definiert eine Weiterleitung.

```
<meta http-equiv="refresh" content="timeToRedirect; URL=redirectDestination">
```

Diese Art der Weiterleitung ist im Header nicht sichtbar. Erst wird das erstellte Dokument geladen, das den redirect erhält: 200 OK. Nach der angegebenen Zeit erfolgt die Weiterleitung auf die angegebene Seite. Das entspricht einer clientseitigen Anforderung und ist im Header nicht als Redirect zu erkennen. Man erhält den Status 200 OK (falls das Dokument existiert und der Server erreichbar ist natürlich).

Aufgabe 4

Zeichnen Sie das HTTP-Protokoll auf wenn Sie sich zu <http://pmeerw.net/www-mm18/geheim> verbinden. Username und Passwort sind geheim, geheim. Welche HTTP Requests bzw. Responses und welche Status Codes treten auf und was bedeuten sie? Wann muss das Passwort erneut eingegeben werden?

Es wird das HTTP/1.1 benutzt. Zuerst erhält man einen Status 401 Not Authorized. Die Response enthält außerdem einen WWW-Authenticate Header, der den Client (Browser) anweist, eine Anmeldung zu starten. Gibt man falsche Daten ein, erhält man immer wieder die gleiche Response. Nach erfolgreicher Authentifizierung wird das Dokument mit Status 200 zurückgegeben. Nach dem GET request bekommen wir den Status Code 302: Found (Umleitung). Das Passwort muss nicht nochmal angegeben werden solange man den Cache seines Browsers nicht löscht.

Aufgabe 5

Erstellen Sie eine 'persönliche' Seite in HTML5 für sich (Name, Adresse, Kontaktmöglichkeit, Beschreibung, Bild, Lebenslauf, etc.). Achten Sie dabei ausschließlich auf die Struktur der Seite, nicht die optische Gestaltung (kein CSS).

- a Was ist die Bedeutung des Attributs alt bei ``? 'alt' für alternate gibt einen alternative Text an, der angezeigt wird, falls das Bild nicht angezeigt werden kann. 'alt' trägt auch zur Accessibility der Seite bei, das es z.B. von Screenreadern gelesen wird.
- b Welche Meta-Information sollte in den Header damit die Seite (leichter) gefunden wird? Von der Überflutung mit meta-tags wird heute abgeraten. Viel wichtiger für eine gute google-Indizierung ist heute das Gesamtbild der Website:
Quelle: <https://moz.com/blog/seo-meta-tags>

- Aussagekräftiger Titel
- Struktur der Seite selbst: Sections, Links, Bilder,... aussagekräftig benennen.
- 'description' meta Tag mit guter Beschreibung verwenden.
- Accessibility: z.B. 'lang' Attribut im html-tag → für Screenreader.
- Mobile-friendly
- Korrekter Code
- Keine externen Weiterleitungen/Links auf die eigene Seite platzieren (don't try to fool the algorithm).

c Welche semantischen HTML-Tags können verwendet werden um die Struktur zu kodieren?

Semantische Tags sind dazu da, Content inhaltlich zu trennen. Sie trennen sich nur manchmal auf die Formatierung aus. Vollständige Liste unter:

https://www.w3schools.com/html/html5_semantic_elements.asp

Verwendete Elemente:

- section
- header
- footer
- details
- summary
- time