

Aufgaben zum Apache-Minikurs 2

Seite 1/2

Name:	<input type="text"/>	
Klasse:	<input type="text"/>	Datum: <input type="text"/>

Download-Ordner bereitstellen

1. Was ist die **DocumentRoot**?
2. Was wird durch die Verzeichnisdirektiven **Options None**, **Options All** bzw. **Options Indexes** bewirkt? Aus Sicherheitsgründen sollte hier die Standardeinstellung **Options None** beibehalten werden.
3. Was bezweckt die Verzeichnisdirektive **AllowOverride None**? Warum sollten Sie hier **AllowOverride All** wählen?
4. Wozu kann man die Parameter **Require all granted** und **Require ip 192.168.0.0/16** verwenden?
5. Wozu dient die Datei **.htaccess** und wo wird sie platziert?
6. Beschreiben Sie den Inhalt der in Box 5 abgebildeten **.htaccess**-Datei!
7. Schalten Sie in der Konfiguration des Apache-Servers den Unterordner **public** für den Download beliebiger Dateien frei! Achten Sie auf passende Ordner- (755) und Dateirechte (644)!
8. Dokumentieren Sie Ihre Vorgehensweise!

Basic-Authentifizierung

9. Machen Sie sich mit der Syntax des Befehls **htpasswd** vertraut.
10. Erstellen Sie eine versteckte Passwortdatei und tragen Sie in diese Datei wenigstens zwei verschiedene Benutzer mit zugehörigem Passwort ein!
11. Kann man Apache-Konten anlegen, wenn es kein gleichnamiges Linux-Konto gibt?
12. Wenn Sie kostenlosen Internetplatz nutzen können, bleibt meistens nur die Möglichkeit die Passwortdatei in zu schützenden Ordner selbst abzulegen. Fachleute halten diesen Speicherort für ziemlich unsicher, obwohl der Apache die Auslieferung versteckter Dateien wie **.htaccess** und **.htpasswd** verweigert. Wenn Sie den Zugriff auf die Verzeichnisstruktur des Servers haben, sollten Sie in produktiv genutzten Systemen diese Datei nicht in einem Unterordner der DocumentRoot ablegen, sondern in einen Parallelordner. Standardmäßig ist Apache so konfiguriert, dass keine Inhalte außerhalb der DocumentRoot-Verzeichnisstruktur ausgeliefert werden.
13. Informieren Sie sich über den Aufbau der Authentifizierungsabschnitts der Datei **.htaccess**: <http://d.apache.org/docs/2.4/howto/auth.html#gettingitworking>. Schauen Sie sich insbesondere die Bedeutung der Parameter **AuthType**, **AuthName**, **AuthUserFile** und **require an**!

Aufgaben zum Apache-Minikurs 2

Seite 2/2

Name:

Klasse:

Datum:

14. Erstellen Sie eine eigenen Datei **.htaccess**! Verwenden Sie als Authentifizierungstyp **Basic**! Wie der Name erahnen lässt, beherrscht Basic keine Verschlüsselung des Anmeldevorgangs. Hier wird lediglich eine Base64-Codierung genutzt. Andere, sicherere Authentifizierungstypen können erst nach weiteren Konfigurationsarbeiten genutzt werden.
 15. Informieren Sie sich über die Base64-Codierung und stellen Sie die wichtigsten Merkmale stichwortartig zusammen! Mit dem Programm HTTPClient können Sie einige Merkmale der Base64-Codierung interaktiv ermitteln.
 16. Erstellen Sie öffentliche und mehrere geschützte Ordner für unterschiedliche Zugangsdaten, kopieren Sie einige Dateien in die Ordner und testen die Funktionsweise Ihrer **.htaccess**-Konfigurationen!
 17. Dokumentieren Sie Ihre Vorgehensweise in Form einer Checkliste!
-

Digest-Authentifizierung

18. Base64-codierte Zeichenketten sind unverschlüsselt. Dekodieren Sie die folgende Anmeldung:
QWRtaW5pc3RyYXRvcjpwIXAhc19HIWghIW0= !
19. Machen Sie sich mit der Syntax des Befehls `htdigest` vertraut. Untersuchen Sie vor allem die Bedeutung des Parameters `-c`!
20. Erstellen Sie eine versteckte Passwortdatei und tragen Sie in diese Datei wenigstens zwei verschiedene Benutzer mit zugehörigem Passwort ein!
21. Informieren Sie sich über den Aufbau der Authentifizierungsabschnitts der Datei `.htaccess`:
http://httpd.apache.org/docs/2.4/mod/mod_auth_digest.html
Schauen Sie sich insbesondere die Bedeutung der Parameter **AuthType**, **AuthName**, **AuthUserFile** und **require** an!
22. Erstellen Sie eine eigenen Datei `.htaccess`! Verwenden Sie als Authentifizierungstyp **Digest**!
23. Informieren Sie sich über die Sicherheit der md5-Verschlüsselung und stellen Sie die wichtigsten Merkmale stichwortartig zusammen!
24. Erstellen Sie öffentliche und mehrere geschützte Ordner für unterschiedliche Zugangsdaten, kopieren Sie einige Dateien in die Ordner und testen die Funktionsweise Ihrer `.htaccess`-Konfigurationen!
25. Dokumentieren Sie Ihre Vorgehensweise in Form einer Checkliste!