

# nginx page caching

Bevor das Caching im Nginx konfiguriert wird, müssen die Seiten, die zwischengespeichert werden sollen oder die Upstream-Webserver, zuerst die passenden Header ausliefern. Das kann z.B. so aussehen:

```
Cache-Control: public, max-age=3600, must-revalidate
```

Was bedeutet das nun?

- public: die Datei ist nicht auf einen bestimmten User zugeschnitten, sondern kann an alle ausgeliefert werden
- max-age: Maximale Lebensdauer in Sekunden
- der Webserver darf den Content nicht mehr ausliefern, bevor er nicht eine auf eine neue Version geprüft hat und ein „304 Not Modified“ erhalten hat

Des Weiteren sollte die Applikation nach Ablauf der Lebensdauer des gecachten Objekts korrekt antworten. D.h. es wird der Statuscode „304 Not Modified“ gesendet, wenn sich die Datei nicht geändert hat. Alternativ wird die Seite in der neuen Version geschickt (Code 200 OK) oder „404 File Not Found“ sollte es sie inzwischen nicht mehr geben.

Nginx wird jetzt so konfiguriert, dass er die Dateien in einem Filesystemcache zwischenspeichern kann. Idealerweise legt man den Cache in eine [Ramdisk](#).

```
http {
    # ...
    proxy_cache_path /ramdisk/nginx_cache levels=1:2 keys_zone=cache:5m max_size=100m;

    server {
        # ...

        proxy_pass          http://upstream;
        proxy_cache          cache;
        proxy_cache_key      "$host$request_uri";
        proxy_cache_revalidate on;
        # Optionally;
        # proxy_cache_use_stale error timeout invalid_header updating http_500 http_502 http_503
        http_504;
    }
}
```

Um so z.B. Bilder oder andere statische Dateien durch den Proxy-Nginx cachen zu lassen, kann im Upstream-Nginx diese Einstellung vorgenommen werden:

```
location ~* \.(js|css|png|jpg|jpeg|gif|ico)$ {  
    expires max;  
    log_not_found off;  
}
```

---

Revision #1

Created 2021-05-05 07:02:28 UTC by magenbrot

Updated 2021-05-05 07:03:03 UTC by magenbrot