

Vereine aufgepasst: Google-Schriftarten dynamisch einbinden birgt Risiken!

25.10.2022

Schriftarten können dynamisch über das Internet in eine Webseite eingebunden werden. Dafür bietet Google u. a. eine Schnittstelle (API) an, die über einen Codeschnitzel (ein Stück Programmiercode) eingebunden werden kann. Das ist einfach und bequem. Aber Vorsicht! Dies verstößt u. U. gegen die Regeln der DSGVO und kann abgemahnt werden.

Schriftarten können dynamisch in eine Webseite eingebunden werden. Dafür bietet Google – wie auch anderen Anbieter im Netz – eine Schnittstelle (API) an, die vom Webseitenbetreiber über einen Codeschnitzel (ein Stück Programmiercode) in die Internetseite eingebunden werden kann. Dadurch ist es möglich, einige tausend Schriftarten nutzen zu können. Allerdings werden dann, sobald ein User die Webseite besucht, Informationen über den Nutzer an den Schriftartenanbieter, in diesem Falle Google, weitergeleitet. Dieser Vorgang ist nicht [DSGVO-konform](#) und kann deshalb abgemahnt werden. Mögliche Abmahnkosten können dabei entstehen.

Dies gilt jedoch nur, wenn Schriftarten für die Webseite dynamisch, d. h. über eine Schnittstelle zu einem anderen Server eingebunden werden. Werden Schriftarten lokal auf dem eigenen Server verwendet, besteht die Gefahr der Abmahnung nicht. Es werden dann keine Informationen an den Schriftartenanbieter weitergereicht.

Daher sollte die eigene Webpräsenz dahingehend überprüft werden, ob Schriftarten dynamisch eingebunden werden. Dies kann u. a. bei Webbauxkästen im Internet der Fall sein. Wenn dies der Fall ist sollte geprüft werden, ob die Schriftarten nicht lokal eingebunden werden können. Google z. B. bietet diese Möglichkeit, seine Schriftarten auch lokal statt dynamisch einzubinden.

Weitere Informationen:

[Artikel "Abmahnwelle DSGVO-Verstoß wegen "Google-Fonts"](#)

Landgericht München - Urteil vom 20.01.2022:

[rewis.io/urteile/urteil/lhm-20-01-2022-3-o-1749320/](https://www.rewis.io/urteile/urteil/lhm-20-01-2022-3-o-1749320/)

[Zurück](#)



Achtung: Abmahnung