FREE eBook

LEARNING

https

Free unaffiliated eBook created from
Stack Overflow contributors.

#https

Table of Contents

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: https

It is an unofficial and free https ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official https.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with https

Remarks

This section provides an overview of what https is, and why a developer might want to use it.

It should also mention any large subjects within https, and link out to the related topics. Since the Documentation for https is new, you may need to create initial versions of those related topics.

Examples

Getting started with HTTPS

HTTPS *(Hypertext Transfer Protocol Secure)* is an encrypted version of HTTP protocol, most often used in connection with services where only the sender and receiver must know the message. It's required if you handle credit card information, and will improve your rank on Google.

To enable HTTPS you need to check if your web host supports it—if you don't know you can ask their support for help and information about it. Some web hosts may take some money for it.

> **Important!**: Your HTTPS needs to use sha2 or sha3 *(sha1 is blocked by **Chrome**, **Firefox**, **Edge** and **IE**)*

When you have enabled HTTPS on your web host, you can use HTTPS. But the browser does not use HTTPS as default; the best way to make sure that all traffic runs on HTTPS is by using a .htaccess file and adding it to the root of your website.

The .htaccess file

```
RewriteEngine On

# If we receive a forwarded http request from a proxy...
RewriteCond %{HTTP:X-Forwarded-Proto} =http [OR]

# ...or just a plain old http request directly from the client
RewriteCond %{HTTP:X-Forwarded-Proto} =""
RewriteCond %{HTTPS} !=on

# Redirect to https version
RewriteRule ^ https://%{HTTP_HOST}%{REQUEST_URI} [L,R=301]
```

This will change the http:// to https://.

> **NOTE**: .htaccess is a system file, and can't be seen by default. How to show .htaccess

Read Getting started with https online: https://riptutorial.com/https/topic/2910/getting-started-with-https

Chapter 2: HTTP Strict Transport Security (HSTS)

Parameters

Parameter	Details
`max-age=31536000`	Time in seconds. HSTS will be enforced for this future time period.
`includeSubDomains`	HSTS should be applied for this domain and all of its sub-domains.
`preload`	This domain agrees to be included in a HSTS pre-load list

Remarks

See also

- MDN - HTTP Strict Transport Security
- Wikipedia - HTTP Strict Transport Security
- HSTS preload list

Examples

HSTS Header

```
Strict-Transport-Security: max-age=31536000; includeSubDomains
```

`Strict-Transport-Security` is a promise to the browser that all future requests to this domain will be secure.
For the future time period `max-age`:

- All outgoing HTTP requests from the browser will be converted to HTTPS *on the client* (not an HTTP redirect).
- If the certificate is invalid (e.g. outdated or self-singed), the user will be unable to white-list it and the site will remain inaccessible.

HSTS behavior is meant to eliminate Man-in-the-Middle attacks that use HTTPS stripping, issuing of invalid certificates (and expecting the user to add and exception), and redirecting on HTTP requests to another destination.

HSTS preload list

```
Strict-Transport-Security: max-age=31536000; includeSubDomains; preload
```

HSTS is activated only after a successful HTTPS request to the server with a valid certificate.
There is still a risk of a first-time user accessing the site, at which point a Man-in-the-Middle attack is possible.
To make the site secure even before the first request the domain can be added to a preload list, already configured in browsers.
The `preload` parameter is not used by the browsers directly, but it an indiciation to the browser developers that the site developers really asked to be added to the preload list.

Read HTTP Strict Transport Security (HSTS) online: https://riptutorial.com/https/topic/3495/http-strict-transport-security--hsts-

Credits

S. No	Chapters	Contributors
1	Getting started with https	Community, finjo, RamenChef, raphinesse, TheCrazyProfessor
2	HTTP Strict Transport Security (HSTS)	Kobi