HTTPS und Zertifikate in Velocity

Source file: doc/ger-de/funktionen/6.0/velocity-https-zertifikate.md Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/features/6.0/velocity-https-certificates.md

Diese Seite richtet sich an Administratoren, die eine Website mit Velocity über HTTPS ausliefern. Das HTTPS von Velocity kümmert sich um sich selbst. Ist kein Zertifikat konfiguriert, erzeugt der Server sein eigenes und liefert trotzdem HTTPS aus. Ist eine Quelle konfiguriert (Ihre eigenen Dateien, ein Archiv, ein PKCS#12-Bundle, Let's Encrypt oder eine andere ACME-CA), prüft er das Zertifikat, liefert es aus, überwacht es, erneuert es und tauscht das neue im laufenden Server aus, ohne Neustart und ohne eine Verbindung abzubrechen. Gilt für Exponential Velocity 0.0.4.27 und später.

Worauf Sie sich verlassen können

- Es gibt keinen Moment, in dem der Server unverschlüsseltes HTTP beantwortet, aber kein HTTPS (HTTPS startet zuerst).
- Ein defektes oder abgelaufenes Zertifikat erreicht nie einen Besucher: Das Paar wird vor der Verwendung geprüft (lässt sich parsen, ist nicht abgelaufen, der Schlüssel gehört dazu), und ein selbstsigniertes Zertifikat springt ein, bis das echte wieder verwendbar ist.
- Die Erneuerung braucht nichts von Ihnen: Let's-Encrypt-Zertifikate werden in einem Hintergrundjob erneuert, nach einem Fehlschlag mit wachsenden Pausen (5 Minuten, verdoppelt bis zu einem Tag), sodass die Ratenbegrenzungen einer CA nie aufgebraucht werden.
- HTTP/2 ist inbegriffen (HTTP/2 und Sicherheit).

Schnellstart

Alle Einstellungen liegen unter `Q.web.https` in der Site-Datei (zum Beispiel `/etc/qbix/sites-enabled/example.com.conf` oder `/etc/vc/...` für Velocity) oder im JSON von `--config`.

Let's Encrypt (Port 80 dieses Servers aus dem Internet erreichbar, DNS zeigt hierher):

JSON

```
{ "Q": { "web": { "https": {
  "port": 443,
  "mode": "letsencrypt",
  "acme": { "email": "your-contact-address", "domains": ["example.com", "www.example.com"] }
} } } }
```

Ihre eigenen Dateien:

JSON

```
{ "Q": { "web": { "https": {
  "port": 443,
  "mode": "files",
  "cert": "/etc/ssl/example.com/fullchain.pem",
  "key":  "/etc/ssl/example.com/privkey.pem"
} } } }
```

Entwicklung, ganz ohne Zertifikat:

JSON

```
{ "Q": { "web": { "https": { "port": 8443, "mode": "self-signed" } } } }
```

Starten Sie den Server. Die Konsole gibt eine Zeile wie `tls: certificate ready (EC, openssl-ecdsa, 0.1s, 397 days, localhost ...)` aus. Prüfen Sie jederzeit mit:

Shell

```
qbixconsole ssl:show
```

Probieren Sie immer zuerst das **Staging**-Verzeichnis von Let's Encrypt aus (`"directory": "letsencrypt-staging"`): Seine Zertifikate sind nicht vertrauenswürdig, aber seine Ratenbegrenzungen großzügig.

Einstellungen (unter `Q.web.https`)

Einstellung	Standard	Bedeutung
`port`	`443`	HTTPS-Port; `--https-port` überschreibt ihn
`mode`	`"manual"`	`files` (gleichbedeutend mit `manual`), `archive`, `pkcs12`, `letsencrypt` (gleichbedeutend mit `acme`), `certbot`, `remote`, `self-signed`
`fallback`	`"self-signed"`	Wenn die Quelle nichts Verwendbares hat: `"self-signed"` oder `"none"` (HTTPS aus)
`watchInterval`	`60`	Sekunden zwischen den Prüfungen der Zertifikatsdateien
`domain`	keiner	Hauptname des Hosts; Standard für die Domains von Let's Encrypt und certbot
`selfSigned.dir`	`<conf dir>/ssl`	Wo das selbstsignierte Paar, importierte Paare und ACME-Dateien liegen
`selfSigned.hosts`	automatisch	Namen und Adressen des selbstsignierten Zertifikats: `domain`, Namen und Aliasse der virtuellen Hosts, der Name dieses Rechners, `localhost`, `127.0.0.1`, `::1`, die gebundene Adresse
`sources.<mode>`	keiner	Eine Klasse, die `Q_WebServer_Certificate_Source` implementiert, um einen eigenen Modus hinzuzufügen

`files`: `cert` (Standard `config/certs/fullchain.pem`; Blattzertifikat allein oder mit Kette; PEM, DER oder PKCS#7), `key` (Standard `config/certs/privkey.pem`; darf dieselbe Datei sein), `chain` (optional), `keyPassword` / `keyPasswordFile` / `keyPasswordEnv`.

`archive`: `archive` (`.zip`, `.tar`, `.tar.gz`/`.tgz`, `.tar.bz2`, `.tar.xz`, `.rar`, `.7z`), `password*` für ein verschlüsseltes Archiv, `keyPassword*` für einen verschlüsselten Schlüssel.

`pkcs12`: `bundle` (`.p12` / `.pfx`) und sein `password`, `passwordFile`, `passwordEnv`.

`letsencrypt` / **ACME (`Q.web.https.acme`)**:

Einstellung	Standard	Bedeutung
`email`	keiner	Kontaktadresse für Ablaufhinweise (empfohlen)
`domains`	`[domain]`	Zu zertifizierende Namen; der erste benennt das Verzeichnis; `*.example.com` braucht `dns-01`

Einstellung	Standard	Bedeutung
`directory`	`"letsencrypt"`	`letsencrypt`, `letsencrypt-staging`, `zerossl`, `google`, `google-staging` oder eine beliebige URL eines ACME-Verzeichnisses
`challenge`	`"http-01"`	`http-01` oder `dns-01`
`webroot`	keiner	HTTP-01-Antworten zusätzlich nach `<webroot>/.well-known/acme-challenge/` schreiben, wenn ein anderer Server Port 80 belegt
`dnsHook`	keiner	`dns-01`: ein Programm, das als `HOOK add NAME VALUE` und `HOOK remove NAME VALUE` aufgerufen wird
`dnsWait`	`30`	Sekunden Wartezeit, nachdem der Hook den Eintrag hinzugefügt hat
`keyType`	`"ec256"`	`ec256`, `ec384`, `rsa2048`, `rsa3072`, `rsa4096`; ein neuer Schlüssel für jedes Zertifikat
`renewAt`	`0.33`	Erneuern, wenn dieser Anteil der Laufzeit übrig ist
`eab.kid`, `eab.hmacKey*`	keiner	External Account Binding (ZeroSSL, Google)
`timeout`	`300`	Sekunden Wartezeit auf Validierung und Ausstellung
`dir`, `challengeDir`	`<ssl dir>/acme`, `<acme dir>/challenges`	Wo die ACME-Dateien liegen

`certbot` und `remote` (eine URL, die alle `checkInterval` heruntergeladen wird, Standard `86400`) stehen ebenfalls zur Verfügung; siehe `docs/https.md` der Engine.

Wie es sich verhält

1. Eine Quelle liefert das Zertifikat; das Paar wird geprüft; HTTPS liest nur eine **private Kopie** (`<ssl dir>/active-<port>-<fingerprint>-<pid>.pem`), die diesem Serverprozess gehört, sodass eine Datei, die anderswo neu geschrieben wird, keinen Handshake stören kann. Ein zweiter Server im selben Verzeichnis und auf demselben Port schreibt seine eigene Kopie (behoben in 0.0.4.28: Zuvor konnte ein zweiter Server dem ersten das HTTPS wegnehmen).
2. Alle `watchInterval` Sekunden werden die Dateien der Quelle geprüft; ein geändertes, gültiges Paar wird der nächsten Verbindung präsentiert. Ein Zertifikat, das vor seinem Schlüssel geschrieben wird, bekommt ein Intervall Zeit, um zur Ruhe zu kommen.
3. **Selbstsigniert** wird in dieser Reihenfolge versucht: PHP openssl mit ECDSA P-256; PHP openssl mit RSA 2048, signiert mit SHA-256; das System-Programm `openssl`; das Snakeoil-Paar des Systems. Es gilt 397 Tage, wird 30 Tage vor Ablauf und bei jeder Änderung der Namen erneuert und liegt in `ssl/self-signed.*`. (Release 0.0.4.27 signiert mit SHA-256, weil Systeme, die SHA-1 ablehnen, das frühere Zertifikat abgelehnt haben.)
4. **Auswahl pro Domain (SNI)**: Das Zertifikat, das eine Domain abdeckt, wird pro Verbindung ausgewählt und im Reiter Domains des Panels angezeigt (Control Panel).

Was in `ssl/` liegt

```
ssl/
  self-signed.pem, self-signed.key      the self-signed pair
  active-443-<fp>-<pid>.pem/.key        the copy one server reads (do not edit)
  imported/                             pairs imported from DER, archives, bundles, remote
  acme/account-<ca>.pem                 one account key per CA: keep it
  acme/<first domain>/fullchain.pem, privkey.pem, state.json
  acme/challenges/<token>               pending HTTP-01 answers (short-lived)
```

Sichern Sie `ssl/`. Alles darin lässt sich neu erzeugen, außer den ACME-Kontoschlüsseln.

Betrieb

- `qbixconsole ssl:show` (und `--json`): das ausgelieferte Zertifikat und jedes bekannte Zertifikat nach Ablaufdatum. Achten Sie auf `daysLeft`: Unter 14 bei einem CA-Zertifikat bedeutet, dass die Erneuerung seit zwei Wochen fehlschlägt; `acme.lastError` sagt, warum.
- Der Reiter **SSL** des Control Panels zeigt dasselbe, bearbeitet die unbedenklichen Einstellungen (Modus, ACME-E-Mail und -Verzeichnis, `renewAt`, Hostnamen), erneuert und lädt neu ohne Neustart.
- Lassen Sie NTP laufen: Zertifikate werden gegen die Systemuhr geprüft.

Fehlerbehebung

Symptom	Ursache	Behebung
ACME-Fehler `Invalid response from http://.../.well-known/acme-challenge/...`	Port 80 erreicht diesen Server nicht	Öffnen Sie Port 80, leiten Sie `/.well-known/acme-challenge/` per Proxy dorthin weiter, oder setzen Sie `acme.webroot` für den Server, der Port 80 belegt
`the DNS hook failed`	Der Hook wurde mit einem Wert ungleich null beendet	Führen Sie ihn von Hand mit `add NAME VALUE` aus
`no private key found (is it encrypted? ...)`	Verschlüsselter Schlüssel, kein Passwort angegeben	Setzen Sie `keyPassword`, `keyPasswordFile` oder `keyPasswordEnv`
`none of the certificates belongs to the private key`	Schlüssel und Zertifikat stammen aus verschiedenen Anträgen	Verwenden Sie den Schlüssel, für den das Zertifikat ausgestellt wurde
`no tool here reads ...rar`	Kein `bsdtar`/`unrar`	Installieren Sie `libarchive-tools` oder packen Sie neu als `.zip`

Grenzen und Sicherheitshinweise

- Für `files`, `pkcs12`, `.zip`/`.tar*`, selbstsigniert oder Let's Encrypt ist kein externes Programm nötig; die openssl-Extension von PHP genügt.
- Private Schlüssel werden mit `0600` atomar geschrieben und nie protokolliert. Externe Programme laufen mit einer Argumentliste, nie über eine Shell. Der ACME-Client prüft das Zertifikat der CA (schalten Sie `verify` nur für eine Test-CA ab).

- Wildcards brauchen `dns-01` mit einem Hook-Programm.

Verwandte Seiten

- Control Panel (Reiter SSL und Domains), Scheduler (Erneuerungsjobs), Webserver Velocity
- Spezifikationen: HTTP/2 und Sicherheit, Engine-Einstellungen (Wiederaufnahme von TLS-Sitzungen, `--https-port`)
- Aktualisierung: Hinweise zur Aktualisierung der Velocity-Engine, Velocity-Engines (`[HTTPSSettings]`)
- Changelog: Engine Exponential Velocity
- Geschichte: 24. September (Zertifikatssubsystem), 25. bis 30. September (Reiter SSL, Zertifikate pro Domain)

> **Werden Sie Exponential-Partner**
>
> Partner teilen ihre Informationen, stellen Mittel bereit, geben Beiträge zurück und entwickeln die Produkte mit uns, und so wächst Exponential. Eine Partner-Website und ein Dashboard sind für Mitte 2027 geplant: Werden Sie noch heute Partner, damit wir schneller wachsen und neue Funktionen früher liefern.
>
> Partner: hier beginnen | Wie Sie sich bewerben

Exponential Copyright © 1998-2026 Exponential Foundation and 7x
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".
This documentation is free documentation, licensed under the GNU Free Documentation License 1.3 or later:
https://www.gnu.org/licenses/fdl-1.3.html.