

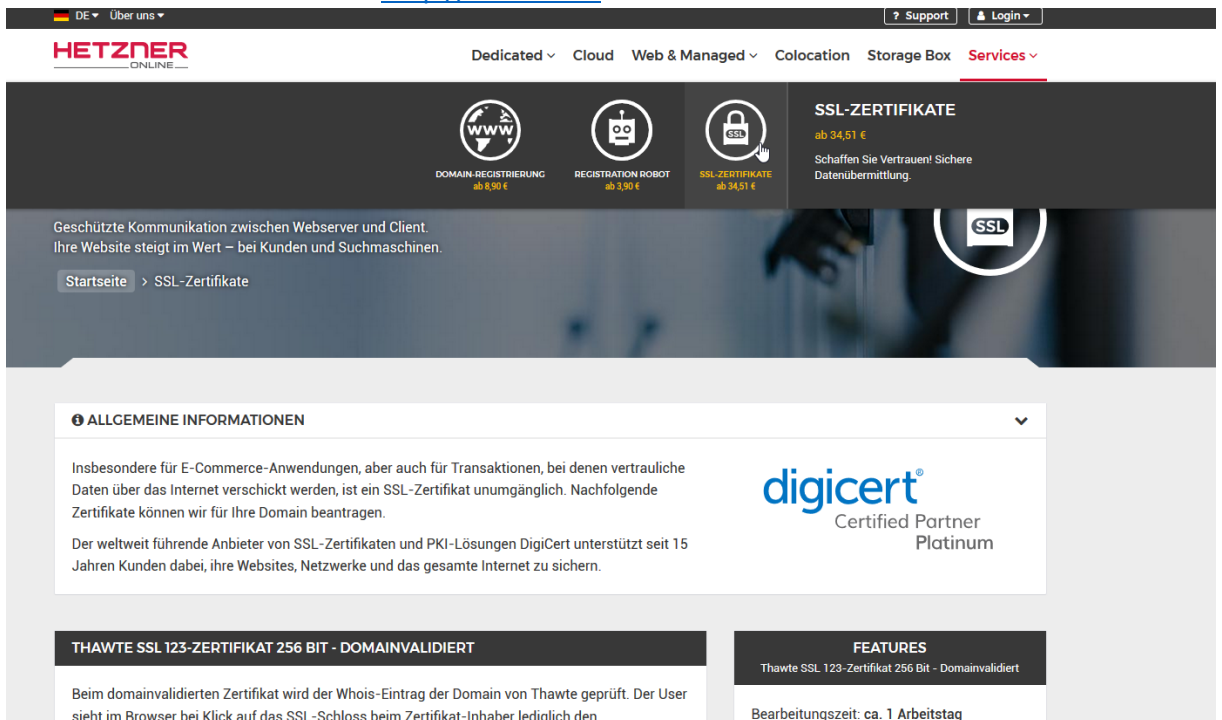
Erstellen und einbinden eines SSL Zertifikats in den Tomcat Server für den Betrieb von Webterminal oder Terminplaner

Inhalt

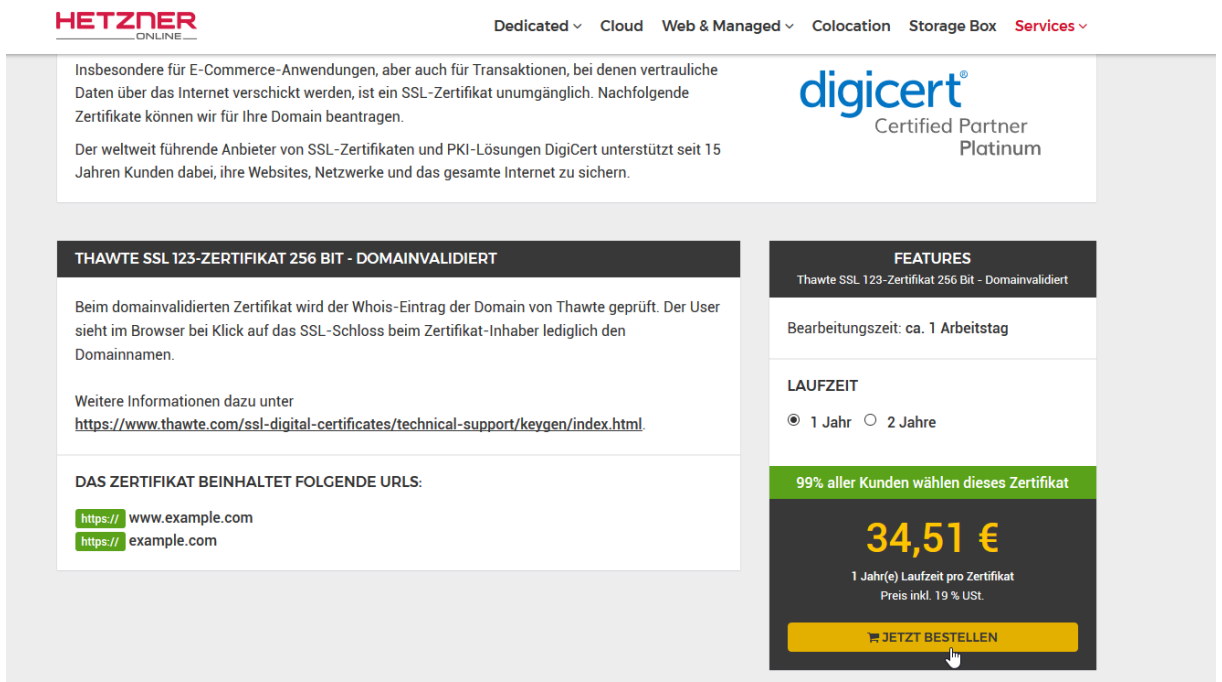
Konfigurationsbeispiel eines SSL Zertifikats von „Hetzner.de“	2
Konfigurationsbeispiel eines SSL Zertifikats von „Strato.de“	8

Konfigurationsbeispiel eines SSL Zertifikats von „Hetzner.de“

1. Erwerben Sie über die Website <http://Hetzner.de> ein SSL Zertifikat



The screenshot shows the Hetzner website's SSL certificate page. At the top, there are navigation menus for 'Dedicated', 'Cloud', 'Web & Managed', 'Colocation', 'Storage Box', and 'Services'. Below this, there are three service icons: 'DOMAIN-REGISTRIERUNG' (ab 8,90 €), 'REGISTRATION ROBOT' (ab 3,90 €), and 'SSL-ZERTIFIKATE' (ab 34,51 €). The main heading is 'SSL-ZERTIFIKATE' with a price of 'ab 34,51 €' and the text 'Schaffen Sie Vertrauen! Sichere Datenermittlung.' Below this, there is a section for 'ALLGEMEINE INFORMATIONEN' with a description of SSL certificates and a 'digicert® Certified Partner Platinum' logo. A 'THAWTE SSL 123-ZERTIFIKAT 256 BIT - DOMAINVALIDIERT' section explains that the domain is verified. A 'FEATURES' box states 'Thawte SSL 123-Zertifikat 256 Bit - Domainvalidiert' and 'Bearbeitungszeit: ca. 1 Arbeitstag'.



This screenshot provides a more detailed view of the SSL certificate purchase page. It includes the same navigation and service icons as the previous screenshot. The 'ALLGEMEINE INFORMATIONEN' section is expanded, showing a description of SSL certificates and the DigiCert logo. The 'THAWTE SSL 123-ZERTIFIKAT 256 BIT - DOMAINVALIDIERT' section provides more details, including a link to technical support: <https://www.thawte.com/ssl-digital-certificates/technical-support/keygen/index.html>. Below this, it lists 'DAS ZERTIFIKAT BEINHALTET FOLGENDE URLS:' with two example URLs: [https:// www.example.com](https://www.example.com) and [https:// example.com](https://example.com). The 'FEATURES' section is also expanded, showing 'Bearbeitungszeit: ca. 1 Arbeitstag' and 'LAUFZEIT' options: 1 Jahr and 2 Jahre. A green bar indicates '99% aller Kunden wählen dieses Zertifikat'. The price is prominently displayed as '34,51 €' for '1 Jahr(e) Laufzeit pro Zertifikat' (Preis inkl. 19 % USt.). A yellow button at the bottom says 'JETZT BESTELLEN'.

Hierfür müssen Sie einen Kundenaccount erstellen. Nach erfolgreicher Registrierung können Sie sich unter <https://konsoleh.your-server.de/> mit Ihren Zugangsdaten anmelden.

KONSOLE ^M

Welchen Zugang möchten Sie nutzen?



konsolEH
Kundenaccount



konsolEH
Domainzugang



Webmail-
zugang

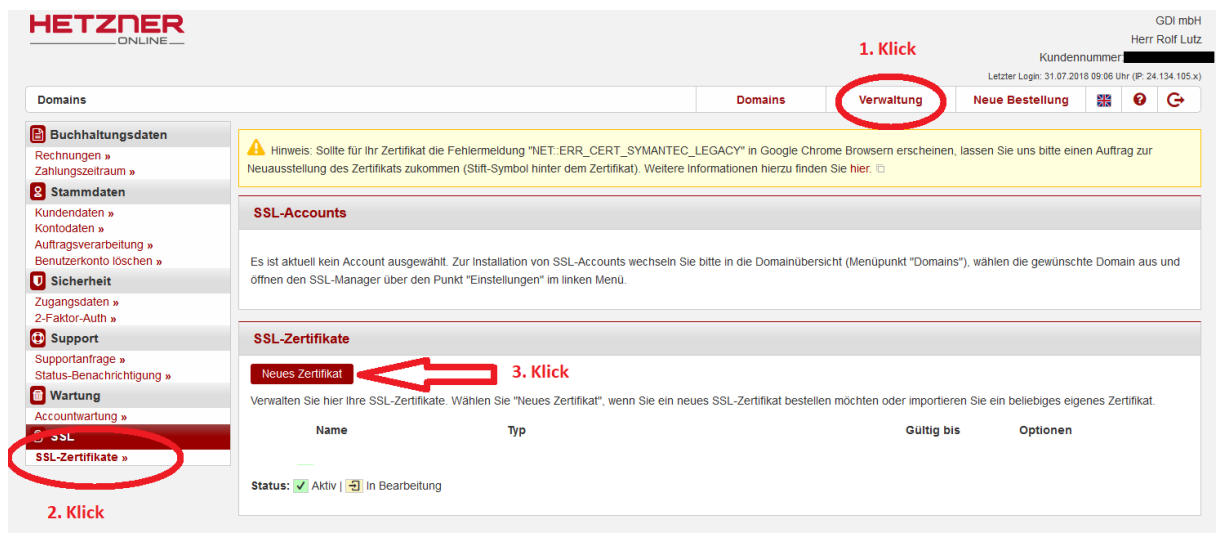
Login

Passwort

[Passwort vergessen?](#)

Sie haben noch keinen Kundenaccount? [Jetzt anmelden](#)

Über die Menüpunkte „Verwaltung=>SSL-Zertifikate=> Neues Zertifikat“ können Sie ein Zertifikat bestellen....



HETZNER ONLINE

GDI mbH
Herr Rolf Lutz
Kundennummer: [redacted]
Letzter Login: 31.07.2018 09:06 Uhr (IP: 24.134.105.x)

1. Klick

Domains Domains **Verwaltung** Neue Bestellung

2. Klick

SSL-Zertifikate

3. Klick

Neues Zertifikat

Verwalten Sie hier Ihre SSL-Zertifikate. Wählen Sie "Neues Zertifikat", wenn Sie ein neues SSL-Zertifikat bestellen möchten oder importieren Sie ein beliebiges eigenes Zertifikat.

Name	Typ	Gültig bis	Optionen
—	—	—	—

Status: Aktiv | In Bearbeitung

HETZNER
ONLINE


Domains Domains

Neue Bestellung

- »Zertifikat
- »Zertifikatsdaten
- »Kontakt & Authentifizierung
- »Zusammenfassung und Bestellung

Ein neues SSL-Zertifikat hinzufügen

Business Zertifikat



Bestellen Sie ein neues Business SSL-Zertifikat von Symantec (Thawte)

ab **34,51 €**

Bestellen

Zurück

Preise inkl. 19 % USt

Folgen Sie dem Bestellprozess wie auf den Bildern abgebildet....

Neue Bestellung

- »Zertifikat
- »Zertifikatsdaten
- »Kontakt & Authentifizierung
- »Zusammenfassung und Bestellung

Schritt 1 von 4 - Zertifikat auswählen

Zertifikat

- Thawte SSL 123-Zertifikat - Domainvalidiert
Dieses Zertifikat wird auf genau eine gültige (Sub-)Domain ausgestellt (zusätzl. mit www-Prefix). Als Aussteller wird "Thawte" angezeigt. Die Authentifizierung erfolgt per DNS, Datei (Webspace) oder Email.
Bearbeitungszeit ca. 1 Arbeitstag.
- Thawte SSL Zertifikat - Adressvalidiert
In dieses Zertifikat können bis zu 5 (Sub-)Domains eingetragen werden. Zur Validierung werden u.a. Handelsregisterauszug, Bankdaten und Telefondaten geprüft. Im Browser werden Zertifikat-Inhaber, Firmennamen und Ort angezeigt.
Bearbeitungszeit ca. 2-4 Wochen.
- Thawte Wildcard Zertifikat - Adressvalidiert
Sichern Sie alle Subdomains einer Domain (*.domain.com) und die Domain selbst (domain.com). Zur Authentifizierung werden u.a. Handelsregisterauszug, Bankdaten und Telefondaten geprüft. Der User sieht im Browser bei Klick auf das SSL-Schloss beim Zertifikat-Inhaber den Firmennamen und den Ort.
Bearbeitungszeit ca. 2-4 Wochen.
- Thawte SSL Webserver-Zertifikat mit EV
Das "Extended Validation" SSL-Zertifikat macht den Benutzern durch die grüne Adressleiste in den meisten Browsern deutlich, dass diese Website sicher ist und die Identifizierung nach den branchenweit höchsten Standards durchgeführt wurde. Im Browser werden Zertifikat-Inhaber, Firmennamen und Ort angezeigt.
Bearbeitungszeit ca. 2-4 Wochen.
- Thawte SSL 123 Wildcard Zertifikat - Domainvalidiert
Sichern Sie alle Subdomains einer Domain (*.domain.com) und die Domain selbst (domain.com). Als Aussteller wird "Thawte" angezeigt. Die Authentifizierung erfolgt per DNS, Datei (Webspace) oder Email.
Bearbeitungszeit ca. 1 Arbeitstag.

Laufzeit 1 Jahr(e)

Anzahl Domainnamen 1

Schlüssellänge 2048 bit

Zertifikatskette SHA-256 unter SHA-1-Rootzertifikat Vollständige SHA-256-Kette

Kosten für gewählte Laufzeit 34.51 Euro

(Preise inkl. 19 % USt)

Zurück Weiter

HETZNER ONLINE GDI mbH
Herr Rolf Lutz
Kundennummer: [REDACTED]
Letzter Login: 26.07.2018 15:37 Uhr (IP: 24.134.105.x)

Domains Domains Verwaltung **Neue Bestellung** [?]

Neue Bestellung

1. »Zertifikat
2. »Zertifikatsdaten
3. »Kontakt & Authentifizierung
4. »Zusammenfassung und Bestellung

Schritt 2 von 4 - Zertifikatsdaten

Während des Bestellprozesses werden alle nötigen Keys für Ihr Zertifikat automatisch mit den untenstehenden Angaben erstellt und für die Verwendung in konsoleH gespeichert. Sollten Sie dies nicht wünschen, haben Sie hier die Möglichkeit einen eigenen CSR-Key zu importieren. Das Zertifikat wird dann auf diesen CSR ausgestellt und kann anschließend für den externen Gebrauch heruntergeladen werden. Es kann jedoch **nicht** in konsoleH verwendet werden.

Eigenen CSR verwenden:

Domainnamen

Ausstellen für:

Domaineintrag:

Identität

Firma*

Abteilung*

Straße*

Postleitzahl*

Stadt*

Staat*

Bundesland*

E-Mailadresse*

Zurück **Weiter**

Wählen Sie eine für Sie geeignete Authentifizierungsmethode:

HETZNER ONLINE GDI mbH
Herr Rolf Lutz
Kundennummer: [REDACTED]
Letzter Login: 26.07.2018 15:37 Uhr (IP: 24.134.105.x)

Domains Domains Verwaltung **Neue Bestellung** [?]

Neue Bestellung

1. »Zertifikat
2. »Zertifikatsdaten
3. »Kontakt & Authentifizierung
4. »Zusammenfassung und Bestellung

Schritt 3 von 4 - Kontakt & Authentifizierung

Technischer Ansprechpartner

Vorname

Nachname

Telefon

E-Mailadresse

Position

Authentifizierung

Hinweis: Da der Domainname frei gewählt wurde, kann die Authentifizierung per DNS/Datei nicht automatisch erfolgen. Stellen Sie daher bitte sicher, dass Sie über Zugriff auf die DNS-Einstellungen bzw. FTP-Zugriff verfügen.

Approve E-Mail @

Hinweis: Bitte wählen Sie hier unabhängig von der gewünschten Authentifizierungsmethode eine funktionierende E-Mailadresse. Sollte die Authentifizierung fehlschlagen wird auf diese E-Mailadresse zurückgegriffen.

Authentifizierungsmethode

DNS-Authentifizierung

Bei dieser Option wird die Authentifizierung über einen Eintrag im Zonefile Ihrer Domain vorgenommen. Dieser Vorgang kann automatisch abgewickelt werden, sofern Ihre Domain die Nameserver der KonsolEH nutzt. In allen anderen Fällen prüfen Sie bitte, ob Ihr Provider das Bearbeiten des Zonefiles zulässt.

Dateibasierte Authentifizierung

Es muss eine Datei auf Ihrem Weospace angelegt werden, worüber die Authentifizierung stattfinden kann. Diese Option steht nicht für Domainregistrierungen zur Verfügung.

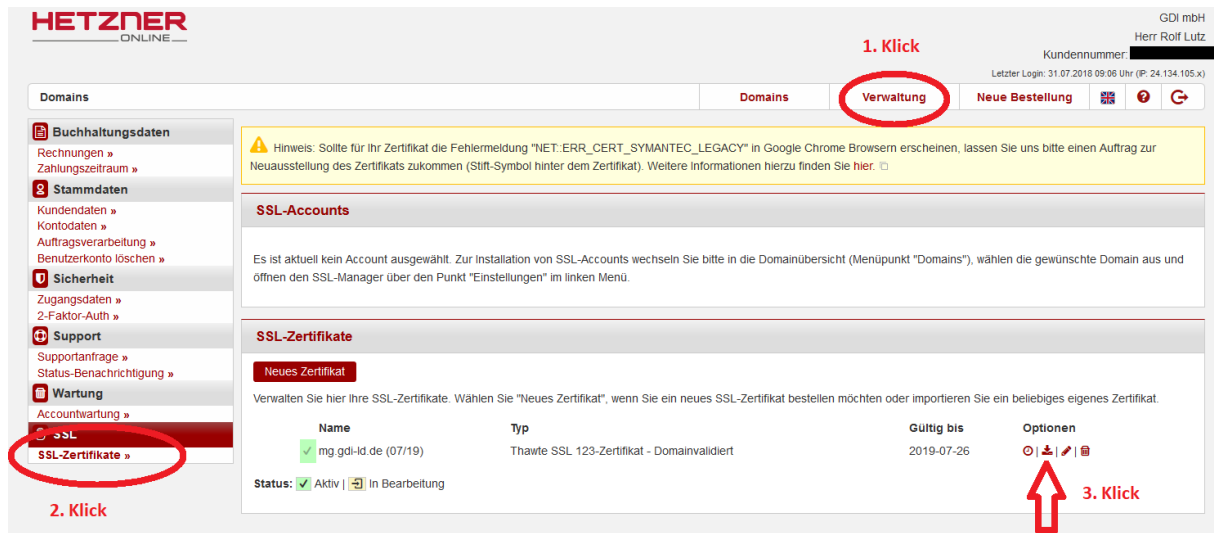
Approve-Email

Sie erhalten eine sogenannte "Approve-Email" an eine der unten ausgewählten E-Mailadressen unterhalb Ihrer Domain.

Zurück **Weiter**

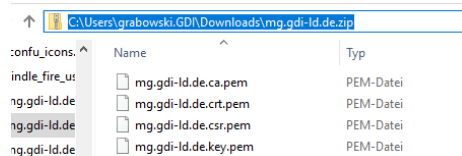
Schließen Sie den Bestellvorgang ab... Nach erfolgreicher Ausstellung des Zertifikats, werden Sie per Email benachrichtigt.

- Nachdem der Bestellprozess erfolgreich durchlaufen wurde, können Sie in Ihrer „Konsole“ das Zertifikatspaket herunterladen.



The screenshot shows the Hetzner Online control panel. In the top right corner, the user is identified as Herr Rolf Lutz. The main navigation bar includes 'Domains', 'Verwaltung' (circled in red), and 'Neue Bestellung'. A yellow warning box at the top contains a message about a certificate error. Below this, there are sections for 'SSL-Accounts' and 'SSL-Zertifikate'. The 'SSL-Zertifikate' section shows a table with one certificate entry: 'mg.gdi-ld.de (07/19)' with a status of 'Aktiv'. The 'Optionen' column for this entry contains a download icon, which is highlighted by a red arrow labeled '3. Klick'. The left sidebar contains various menu items, with 'SSL-Zertifikate' circled in red and labeled '2. Klick'. The top right corner has a '1. Klick' label pointing to the 'Verwaltung' menu item.

- Sie erhalten eine Zip Datei mit folgendem Inhalt. (Hierbei wurde ein SSL Zertifikat für die Subdomain mg.gdi-ld.de angefordert)



Entpacken Sie die „*.ca.pem“, „*.crt.pem“ sowie die *.key.pem“ Dateien in ein beliebiges Verzeichnis.

- Kopieren Sie den kompletten Inhalt der „*.ca.pem“ Datei und fügen Sie diesen an das Ende der „*.crt.pem“ Datei. (Bei diesen Dateien handelt es sich um Textdateien, die mit einem beliebigen Editor bearbeitet werden können.)
- Installieren Sie das beiliegende Programm „OpenSSL“ (<http://gdimbh.de/Setup/Terminplaner/OpenSSL/win32openssl.exe>)
- Im Installationsverzeichnis des Programms „OpenSSL“ finden Sie im Unterordner „bin“ die „openssl.exe“

Über die Kommandozeile muss folgender Befehl ausgeführt werden

```
openssl pkcs12 -export -in "<PFAD-UNTER-DEM-DIE-ZERTIFIKATE-ZU-FINDEN-SIND>\mg.gdi-ld.de.crt.pem" -inkey "<PFAD-UNTER-DEM-DIE-ZERTIFIKATE-ZU-FINDEN-SIND>\mg.gdi-ld.de.key.pem" -out "<PFAD-UNTER-DEM-DIE-ZERTIFIKATE-ZU-FINDEN-SIND>\myZert.p12" -name tomcat
```

Beim Ausführen dieses Befehls werden Sie aufgefordert ein neues Passwort zu hinterlegen.

Konnte der Prozess erfolgreich durchgeführt werden, so wird eine Datei namens „myZert.p12“ erstellt

Wichtig: Sollten Sie das „p12“ Zertifikat auf andere Art und Weise erzeugen, achten Sie bitte darauf, dass der interne Key-Name „tomcat“ lautet.

7. Diese Datei hinterlegen Sie bitte im „cert“ Ordner Ihres TomCat
z.B.: C:\GDI\WebAppsTomcat\apache-tomcat-9.0.0.M10\cert
8. Anschließend müssen Sie noch folgende Einstellungen in der Tomcat „server.xml“ tätigen.
Diese XML-Datei finden Sie im Installationsverzeichnis des Tomcat im Unterordner „conf“.
Folgende „Connector“ Einstellungen müssen angepasst werden:

```
<Service name="Catalina">
  <!-- The connectors can use a shared executor, you can define one or more named thread pools-->
  <!-- <Executor name="tomcatThreadPool" namePrefix="catalina-exec-" maxThreads="150" minSpareThreads="47"/> -->
  <!-- A "Connector" represents an endpoint by which requests are received and responses are returned. Documentation at : Java HTTP Connector: /docs/config/http.html (blocking & non-blocking) Java AJP Connector: /docs/config/ajp.html APR (HTTP/AJP)
  Connector: /docs/apr.html Define a non-SSL/TLS HTTP/1.1 Connector on port 8080 -->
  <Connector port="8080" redirectPort="8081" protocol="HTTP/1.1" connectionTimeout="20000"/>
  <!-- A "Connector" using the shared thread pool-->
  <!-- <Connector executor="tomcatThreadPool" port="8080" protocol="HTTP/1.1" connectionTimeout="20000" redirectPort="8443" /> -->
  <!-- Define a SSL/TLS HTTP/1.1 Connector on port 8443 This connector uses the NIO implementation that requires the JSSE style configuration. When using the APR/native implementation, the OpenSSL style configuration is required as described in the
  documentation at /docs/config/ssl.html -->
  <Connector port="8081" protocol="org.apache.coyote.http11.Http11NioProtocol" sslProtocol="TLS" secure="true" scheme="https" maxThreads="150" keystorePass="GDI0815aXa" keystoreFile="C:\GDI\WebAppsTomcat\apache-tomcat-9.0.0.M10\cert\keystore" clientAuth="false" SSLEnabled="true"/>
  <!-- <Connector port="8082" protocol="org.apache.coyote.http11.Http11NioProtocol" sslProtocol="TLS" clientAuth="false" secure="true" scheme="https" keystoreType="PKCS12" keystorePass="test" keystoreFile="C:\GDI\WebAppsTomcat\apache-tomcat-9.0.0.M10\cert\myZert.p12" SSLEnabled="true"
  maxThreads="150" protocol="org.apache.coyote.http11.Http11NioProtocol"/> -->
  <Connector port="8009" protocol="AJP/1.3" redirectPort="8443"/>
  <!-- An Engine represents the entry point (within Catalina) that processes every request. The Engine implementation for Tomcat stand alone analyzes the HTTP headers included with the request, and passes them on to the appropriate Host (virtual host).
  Documentation at /docs/config/engine.html -->
  <!-- You should set jvmRoute to support load-balancing via AJP ie : <Engine name="Catalina" defaultHost="localhost" jvmRoute="jvm1"> -->
  <Engine name="Catalina" defaultHost="localhost" jvmRoute="jvm1">
    <Host name="localhost" appBase="webapps" <!--
  </Host>
  </Engine>
</Service>
```

Ändern Sie bitte folgende Attribute ab:

keyStorePass: Das über OpenSSL neu definierte Passwort eintragen

KeystoreFile: Pfad zur neu erstellten „myZert.p12“ Datei an

Fügen Sie folgendes Attribut hinzu:

keystoreType="PKCS12"

Der Connectoreintrag sollte nun wie folgt aussehen=>

```
<Service name="Catalina">
  <!-- The connectors can use a shared executor, you can define one or more named thread pools-->
  <!-- <Executor name="tomcatThreadPool" namePrefix="catalina-exec-" maxThreads="150" minSpareThreads="47"/> -->
  <!-- A "Connector" represents an endpoint by which requests are received and responses are returned. Documentation at : Java HTTP Connector: /docs/config/http.html (blocking & non-blocking) Java AJP Connector: /docs/config/ajp.html APR (HTTP/AJP)
  Connector: /docs/apr.html Define a non-SSL/TLS HTTP/1.1 Connector on port 8080 -->
  <!-- A "Connector" using the shared thread pool-->
  <!-- <Connector executor="tomcatThreadPool" port="8080" protocol="HTTP/1.1" connectionTimeout="20000" redirectPort="8443" /> -->
  <!-- Define a SSL/TLS HTTP/1.1 Connector on port 8443 This connector uses the NIO implementation that requires the JSSE style configuration. When using the APR/native implementation, the OpenSSL style configuration is required as described in the APR/native
  documentation at /docs/config/ssl.html -->
  <Connector port="8081" protocol="org.apache.coyote.http11.Http11NioProtocol" sslProtocol="TLS" clientAuth="false" secure="true" scheme="https" keystoreType="PKCS12" keystorePass="test" keystoreFile="C:\GDI\WebAppsTomcat\apache-tomcat-9.0.0.M10\cert\myZert.p12" SSLEnabled="true"
  maxThreads="150" protocol="org.apache.coyote.http11.Http11NioProtocol"/>
  <!-- <Connector port="8082" protocol="org.apache.coyote.http11.Http11NioProtocol" sslProtocol="TLS" clientAuth="false" secure="true" scheme="https" keystoreType="PKCS12" keystorePass="test" keystoreFile="C:\GDI\WebAppsTomcat\apache-tomcat-9.0.0.M10\cert\myZert.p12" SSLEnabled="true"
  maxThreads="150" protocol="org.apache.coyote.http11.Http11NioProtocol"/> -->
  <Connector port="8009" protocol="AJP/1.3" redirectPort="8443"/>
  <!-- An Engine represents the entry point (within Catalina) that processes every request. The Engine implementation for Tomcat stand alone analyzes the HTTP headers included with the request, and passes them on to the appropriate Host (virtual host).
  Documentation at /docs/config/engine.html -->
  <!-- You should set jvmRoute to support load-balancing via AJP ie : <Engine name="Catalina" defaultHost="localhost" jvmRoute="jvm1"> -->
  <Engine name="Catalina" defaultHost="localhost" jvmRoute="jvm1">
    <Host name="localhost" appBase="webapps" <!--
  </Host>
  </Engine>
</Service>
```

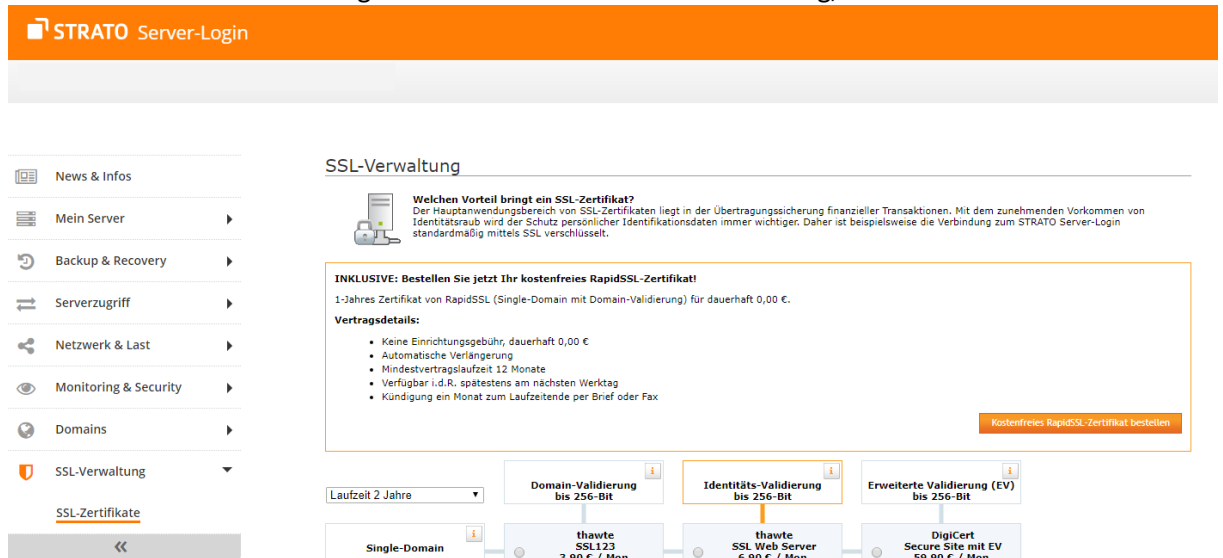
9. Starten Sie Ihren Tomcat-Dienst neu...Das Zertifikat sollte nun korrekt angezeigt werden

Konfigurationsbeispiel eines SSL Zertifikats von „Strato.de“

Strato bietet in verschiedenen Hosting und Server Paketen kostenlose SSL Zertifikate an. Diese können mit ein wenig Aufwand in den Tomcat Server eingebunden werden.

Hier wird am Beispiel eines Dedicated Servers erklärt wie man ein solches SSL Zertifikat anfordert und im Tomcat bereitstellt.

1. Melden Sie sich im Server-Login an und rufen Sie die SSL-Verwaltung, SSL-Zertifikate auf:



SSL-Verwaltung

Welchen Vorteil bringt ein SSL-Zertifikat?
Der Hauptwendungsbereich von SSL-Zertifikaten liegt in der Übertragungssicherung finanzieller Transaktionen. Mit dem zunehmenden Vorkommen von Identitätsraub wird der Schutz persönlicher Identifikationsdaten immer wichtiger. Daher ist beispielsweise die Verbindung zum STRATO Server-Login standardmäßig mittels SSL verschlüsselt.

INKLUSIVE: Bestellen Sie jetzt Ihr kostenfreies RapidSSL-Zertifikat!
1-Jahres Zertifikat von RapidSSL (Single-Domain mit Domain-Validierung) für dauerhaft 0,00 €.

Vertragsdetails:

- Keine Einrichtungsgebühr, dauerhaft 0,00 €
- Automatische Verlängerung
- Mindestvertragslaufzeit 12 Monate
- Verfügbar i.d.R. spätestens am nächsten Werktag
- Kündigung ein Monat zum Laufzeitende per Brief oder Fax

[Kostenfreies RapidSSL-Zertifikat bestellen](#)

Laufzeit 2 Jahre

Option	Preis / Monat	Dauer
Single-Domain	3,90 €	2 Jahre
thawte SSL123	6,90 €	2 Jahre
thawte SSL Web Server	59,90 €	2 Jahre
DigiCert Secure Site mit EV	59,90 €	2 Jahre

2. Klicken Sie dort auf „Kostenfreies RapidSSL Zertifikat bestellen“
3. Geben Sie Ihre Daten ein falls diese nicht schon automatisch ausgefüllt wurden.
4. Im Feld „Ihre SSL Software“ wählen Sie „Other“.
5. „Domain-Check E-Mail“: Hier muss eine E-Mail Adresse mit der Domain angegeben werden für die auch das Zertifikat ausgestellt wird.
6. Damit ein Zertifikat erstellt werden kann müssen Sie ein CSR (Certificate Signing Request) erstellen. In diesem CSR sind die Daten enthalten die am Ende im Zertifikat angezeigt werden. Z. B. um welche Domain es sich handelt und wem es gehört.
7. Installieren Sie das beiliegende Programm „OpenSSL“
(<http://gdimbh.de/Setup/Terminplaner/OpenSSL/win32openssl.exe>)
Im Installationsverzeichnis des Programms „OpenSSL“ finden Sie im Unterordner „bin“ die „openssl.exe“

Über die Kommandozeile muss folgender Befehl ausgeführt werden:

```
openssl genrsa -out www.Ihre-Domain.de.key 2048
```

8. Nun erhalten Sie eine Datei [www.Ihre-Domain.de.key](#). Dies ist ihr privater Schlüssel, diese Datei darf nicht weitergegeben werden.
9. Mit diesem Befehl generieren Sie aus dem privaten Schlüssel die CSR Datei:

```
openssl req -new -key www.Ihre-Domain.de.key -out www.Ihre-Domain.com.csr
```


Hier ist darauf zu achten bei Common Name den korrekten Domainnamen einzugeben, handelt es sich um eine Subdomain dann tragen Sie diese auch ein.

10. Öffnen Sie die Datei mit einem Texteditor, kopieren Sie den kompletten Text und fügen Sie ihn im Strato Formular hinter „Ihr CSR“ ein.

CSR-Eingabe

Erzeugen Sie ein CSR und fügen Sie die CSR-Daten in das folgende Formular ein. Stellen Sie sicher, dass diese Daten die „BEGIN“ und „END“ beginnende Zeilen) wie im folgenden Beispiel enthalten.

CSR Beispiel

```

-----BEGIN NEW CERTIFICATE REQUEST-----
MIIDCjCCARMQAwgTEZMBcGA1UEAxMQG9zdC5kb21haw4ubmFtZTEVMBWGA1UECMMTJ3nYw5pemF0aw9UMRUEwEYDVQQKEwxPcm9udm16YXRpb2
4MDALBgNVBACTBE9pdHkxZjAwBgNVBAGTBN0YXRIMQSwCQYDVQGEwJ
VUZCBnZANBgkqhkiG9w0BAQEFAADBgkqhkiG9w0BAQ0w
G3GYxjS4B837+y3A6xIM90VXV4ZnStIe9n0HgdksQJpwaQe0wneq1fte
hrJ/S55PVPxok+Tqq0t7BfMkUu1YnFduo10pPdw3ceAP9wSrduouI
Vnq2AWTDw2ykyxkg6neb2vYTRvbot7M578Vvh6P8CAwEAACAAMwGgY
KKwYBBAGCNwOCAzEMFgo1LJAuMjESNS4yMDUGC1sGAQQBggcCAQ4xJzA1
MA4GA1UdDwEB/wQEAwIE8DATBgNVHSUEDDAKBggrBgEFBQCDATCB/QYKK
wYBBAGCNwOCAjGB7jCB6wIBAR5aE0AAQBJAHIAbwSzAG8AZgB0ACAAUg
BTAEEAIBTAEWA8ABhAG4AbgB1AGwAIA8DAHIAeQ6wAHQAbwBnAHIAyQB
RQ/MD5Zy3b0zSRFO=
-----END NEW CERTIFICATE REQUEST-----

```

Ihr CSR:

11. Sie erhalten nach einiger Zeit eine E-Mail mit dem folgenden Betreff:
www.Ihre-Domain.de RapidSSL Bestellung / Order:Ihre Bestell Nummer Vollständig / Complete
 In der Email findet sich ein Link unter folgender Überschrift:
**** NUTZER VON MICROSOFT IIS und TOMCAT**
 Über diesen Link können Sie sich für die Geotrust Bestellinformationsseite authentifizieren. Sie erhalten dann in einer weiteren Mail den Link für die Bestellinformationsseite. In dieser Mail klicken Sie auf den Link hinter dem Text „Um fortzufahren, gehen Sie bitte auf“
12. Auf dieser Seite klicken Sie auf „Zertifikatsinfos anzeigen“:

RapidSSL.com®

<p>Informationen zur Bestellung anzeigen</p> <p style="border: 2px solid red; padding: 2px;">Zertifikatsinfos anzeigen</p> <p>Neuausgabe des Zertifikats</p> <p>Zertifikat widerrufen</p> <p>Kontaktinformationen aktualisieren</p> <p>Sprache</p>	<h2 style="margin: 0;">Zertifikatsdaten</h2> <hr/> <h3 style="margin: 0;">Zertifikatdetails</h3> <p>Common Name:</p> <p>Organisation:</p> <p>Organisationseinheit:</p> <p>Servertyp:</p> <p>Neu ausgeben:</p>
--	--

Nun können Sie das Zertifikat im x509 Zertifikat Format herunterladen, Sie sollten dann eine Datei mit diesem Namen erhalten: www_Ihre-Domain_de_ee.crt.

Zertifikat [Show Certificate](#)

Formate: X.509-Zertifikat ▼ Download

13. Diese Seite aufrufen <https://knowledge.digicert.com/generalinformation/INFO1548.html>
SHA-2 Intermediate CAs (under SHA-1 Root)
 Intermediate und Root Zertifikate über View betrachten und den Inhalt in eine Datei DigiCert_Global_Root_CA.pem einfügen. Diese Datei dient dazu, dass im Browser das Zertifikat anerkannt wird.

RSA SHA-2		Legacy
SHA-2 Intermediate CAs (under SHA-1 Root)		
Certificate Type	Intermediate CA	Root
<ul style="list-style-type: none"> RapidSSL Wildcard 	View Download	View Download

14. Nun muss aus dem x509 Zertifikat, der CA Chain und dem privaten Schlüssel eine pkcs12 Datei für den Tomcat Server erstellt werden. Öffnen Sie dazu Openssl.exe wie unter Punkt 7 beschrieben.

openssl.exe

```
pkcs12 -export -in "<PFAD-UNTER-DEM-DIE-ZERTIFIKATE-ZU-FINDEN-SIND>\www_Ihre-Domain_ee.crt" -inkey "<PFAD-UNTER-DEM-DIE-ZERTIFIKATE-ZU-FINDEN-SIND>\www.Ihre-Domain.de.key" -out "<PFAD-UNTER-DEM-DIE-ZERTIFIKATE-ZU-FINDEN-SIND>\myZert.p12" -name tomcat -CAfile "DigiCert_Global_Root_CA.pem" -caname root
```

15. Nach der Bestätigung des Befehls mit Enter werden Sie aufgefordert ein Keystore Passwort einzugeben.
 Konnte der Prozess erfolgreich durchgeführt werden, so wird eine Datei namens „myZert.p12“ erstellt
Wichtig: Sollten Sie das „p12“ Zertifikat auf andere Art und Weise erzeugen, achten Sie bitte darauf, dass der interne Key-Name „tomcat“ lautet.
10. Diese Datei hinterlegen Sie bitte im „cert“ Ordner Ihres TomCat
 z.B.: C:\GDI\WebAppsTomcat\apache-tomcat-9.0.0.M10\cert
16. Anschließend müssen Sie noch folgende Einstellungen in der Tomcat „server.xml“ tätigen. Diese XML-Datei finden Sie im Installationsverzeichnis des Tomcat im Unterordner „conf“. Folgende „Connector“ Einstellungen müssen angepasst werden:

```
<!-- The connectors can use a shared executor, you can define one or more named thread pools-->
<!-- Executor name="tomcatThreadPool" namePrefix="catalina-exec-" maxThreads="150" minSpareThreads="47"/> -->
<!-- A "Connector" represents an endpoint by which requests are received and responses are returned. Documentation at : Java HTTP Connector: /docs/config/http.html (blocking & non-blocking) Java AJP Connector: /docs/config/ajp.html APR (HTTP/AJP) Connector: /docs/apr.html Define a non-SSL/TLS HTTP/1.1 Connector on port 8080 -->
<Connector port="8080" redirectPort="8081" protocol="HTTP/1.1" connectionTimeout="20000"/>
<!-- A "Connector" using the shared thread pool-->
<!-- Connector executor="tomcatThreadPool" port="8080" protocol="HTTP/1.1" connectionTimeout="20000" redirectPort="8443" /> -->
<!-- Define a SSL/TLS HTTP/1.1 Connector on port 8443 This connector uses the NIO implementation that requires the JSSE style configuration. When using the APR/native implementation, the OpenSSL style configuration is required as described in the documentation at /docs/config/engine.html -->
<Connector port="8081" protocol="org.apache.coyote.http11.Http11NioProtocol" sslProtocol="TLS" secure="true" scheme="https" maxThreads="150" keystorePass="GDI0815sXa" keystoreFile="C:\GDI\WebAppsTomcat\apache-tomcat-9.0.0.M10\cert\keystore" clientAuth="false" SSLEnabled="true"/>
<!-- Connector port="8009" redirectPort="8443" protocol="AJP/1.3"/>
<!-- An Engine represents the entry point (within Catalina) that processes every request. The Engine implementation for Tomcat stand alone analyzes the HTTP headers included with the request, and passes them on to the appropriate Host (virtual host). Documentation at /docs/config/engine.html -->
<!-- You should set jvmRoute to support load-balancing via AJP ie : <Engine name="Catalina" defaultHost="localhost" jvmRoute="jvm1"/> -->
<Engine name="Catalina" defaultHost="localhost">
```

Ändern Sie bitte folgende Attribute ab:

keyStorePass: Das über OpenSSL neu definierte Passwort eintragen
 keystoreFile: Pfad zur neu erstellten „Ihre-Domain_de_certificate_Strato.p12“ Datei

Fügen Sie folgendes Attribut hinzu:

keystoreType="PKCS12"

Der Connector eintrag sollte nun wie folgt aussehen:

```

- <Service name="Catalina">
  <!-- The connectors can use a shared executor, you can define one or more named thread pools-->
  <!-- <Executor name="tomcatThreadPool" namePrefix="catalina-exec-" maxThreads="150" minSpareThreads="4"/> -->
  <!-- A "Connector" represents an endpoint by which requests are received and responses are returned. Documentation at : Java HTTP Connector: /docs/config/http.html (blocking & non-blocking) Java AJP Connector: /docs/config/ajp.html APR (HTTP/AJP
  Connector: /docs/ajp.html Define a non-SSL/TLS HTTP/1.1 Connector on port 8080 -->
  <!-- A "Connector" using the shared thread pool-->
  <!-- <Connector executor="tomcatThreadPool" port="8080" protocol="HTTP/1.1" connectionTimeout="20000" redirectPort="8443" /> -->
  <!-- Define a SSL/TLS HTTP/1.1 Connector on port 8443 This connector uses the NIO Implementation that requires the JSSE style configuration. When using the APR/native
  implementation, the OpenSSL style configuration is required as described in the APR/native
  documentation. -->
  <Connector port="8082" sslProtocol="TLS" clientAuth="false" secure="true" scheme="https" keystoreType="PKCS12" keystorePass="test" keystoreFile="C:\GDI\WebAppsTomcat\opache-tomcat-9.0.0.M10\cert\myZert.p12" SSLEnabled="true"
  maxThreads="150" protocol="org.apache.coyote.http11.Http11NioProtocol"/>
  <!-- <Connector port="8082" scheme="https" sslProtocol="TLS" /> -->
  <Connector port="8009" protocol="AJP/1.3" redirectPort="8443"/>
  <!-- An Engine represents the entry point (within Catalina) that processes every request. The Engine implementation for Tomcat stand alone analyzes the HTTP headers included with the request, and passes them on to the appropriate Host (virtual host).
  Documentation at /docs/config/engine.html -->
  <!-- You should set jvmRoute to support load-balancing via AJP ie : <Engine name="Catalina" defaultHost="localhost" jvmRoute="jvm1" /> -->

```

17. Starten Sie Ihren Tomcat-Dienst neu und überprüfen Sie mit dem Aufruf der URL ob das Zertifikat korrekt verarbeitet wird.