HUBJECT

eRoaming Platform

Secure Connection Guide

intercharge everywhere

Content

Tables

Figures

Revision Overview

DATE	VERSION	NOTICE / REASON OF CHANGE	AUTHOR
08.01.2013	0.1	Creation	Daniel van Maren
11.01.2013	0.8	Completion of the first draft	Daniel van Maren
15.01.2013	1.0	Added usage of environment variables, under windows, corrected minor error.	Daniel van Maren
27.02.2013	1.1	Updated document according changes of the 26.02.2013	Nikolaj Langner
01.03.2013	1.2	Updated the document according to feedback of the 01.03.2013	Nikolaj Langner
21.03.2013	1.3	Updated the document according to feedback of the 01.03.2013	Nikolaj Langner
25.10.2013	1.4 DRAFT	Added chapters Abbreviations and Security as well as comments	Christian Pestel Michael Thiel
13.11.2013	1.4 DRAFT2	Checked Connectivity Guide for technical Errors and corrected them.	Daniel van Maren
17.06.2014	1.5	Updated pictures and links	Nikolaj Langner
26.11.2015	2.0	Overall layout and CSR generation	Jaime Brodhag
15.02.2019	2.1	Updated document according to feedback on the 07.12.2018	Yei Perez
08.08.2022	2.2	Added instruction for configuring SSL for Postman and removed chapters related to JKS and Soap UI	Maciej Dados

TABLE 1: REVISIONS OVERVIEW

Abbreviations

Abbreviations	Description
CA	Certificate authority
CSR	Certificate signing request
DER	Distinguished Encoding Rules
GNU	GNU's Not Unix!
GUI	Graphical User Interface
HBS	Hubject Brokering System
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IP	Internet Protocol
PEM	Privacy-enhanced Electronic Mail
PGP	Pretty Good Privacy
PKCS10	Public-Key Cryptography Standard #10
PKCS12	Public-Key Cryptography Standard #12
SSL	Secure Sockets Layer
TLS	Transport Layer Security
UI	User interface

TABLE 2: ABBREVIATIONS

Preconditions

OpenSSL

There are a lot of other tools for tasks regarding SSL – however we recommend the use of OpenSSL, therefore this guide uses OpenSSL for various instructions.

- If you are using Windows, OpenSSL is available for download from various sources. One possible source is available here:

  http://www.heise.de/download/win32-openssl.html

- If you are using Linux/Unix, install OpenSSL from your package repository
- If you are using MacOS, then OpenSSL is already preinstalled on your system

Requirements for your PKCS10 CSR

The PKCS10 Certificate Signing Request (CSR) contains different data: your Public Key and a set of Metadata. To receive a certificate signed by the HubjectCA it is necessary to fulfill some requirements for your CSR:

- Key algorithm: RSA
- Key length: 2048 bit
- Common Name (CN): <DNS of the Partner Backend>
- Organization (O): Hubject
- Organization Unit (OU): <Partner Name>

How to set these values will be covered in the chapter 5 “Creating your PKCS10 CSR”.

In case there is no “real” backend which establishes the connection, but for example a person accessing the web services via Postman, please use the DNS Name of the website of your company as Common Name, or if only one specific person uses the certificate, use the name of that person.

Accessing IP address

At this moment, accesses to all Hubject web services are restricted to specific IPs. Before you can connect to one of the restricted web services, you will need to communicate your public IP to your onboarding manager which supports you in establishing the connection. This can be for instance the static public IP of your partner backend, or the public IP of your company’s internet proxy.

For requests from the eRoaming Platform to your Partner Backend please check chapter 10 “Configure an outgoing partner communication”.

Establishing a connection

Hubject web services for backend communication are protected by using SSL/TLS Client Authentication. This implies that you will need a valid SSL/TLS configuration to be able to connect to Hubject web services.

The basic way of establishing a connection to these web services is as follows:

1. Private Key Generation
2. PKCS10 CSR Generation
3. Sending the CSR to your onboarding manager
4. Receiving the HubjectCA Signed Certificate + HubjectCA + Hubject Intermediate CA IT Certificates
5. Converting your Certificates & Private Key Pair into your desired format
6. Accessing the web services with Client Authentication

Creating your PKCS10 CSR

PLEASE NOTE THAT ALL COMMANDS HERE ARE EXAMPLES.

> Below each command there is a list explaining the different parameters and what should be the value of the parameter - **these values will probably be different** than the example.

During this process, you will create **two files: the private key and the PKCS10 Certificate Signing Request.** Please archive the private key in a secure way: to use your x509 Certificate which we will send you, you will need the private key. Without the private key it is impossible to decrypt messages which were encrypted to your public key (which is contained in your CSR and your Certificate).

Furthermore, this means that anyone with your private key may decrypt all messages which were sent to you (also responses on web service requests!) and are able to identify themselves with your client certificate. Therefore, please make sure that only authorized personal can access the private key.

CAUTION:

> Do not send Hubject your private key. If you send Hubject your private key, we will be unable to sign your CSR. Only send the CSR when creating a certificate signing request

Create a directory where the CSR and Private Key should be saved

STEP 1

Open your command line

ON WINDOWS 10

For instance, you can do this by **typing "cmd" into the search within the start menu and click on the "cmd.exe"** within the search results.

ON MacOS

terminal can be started either by opening Spotlight search by pressing Command+Space keys on your keyboard and then by searching for Terminal, or through **Finder > Applications > Utilities > Terminal**

CAUTION:

Be careful with copying the listed commands below and pasting it into the command line window. It can cause execution problems.

STEP 2

Change environment variables

ON WINDOWS

```
set RANDFILE=C:\Users\userABC\hubject_ssl\.rnd
set PATH=%PATH%;C:\OpenSSL-Win32\bin
```

Please adjust the yellow selection to your preconditions. This means typing in your username and adjusting whether you use Open SSL in Win32 or Win64 bit version.

Make sure that you have Read & Write permissions to the file path you have set in RANDFILE.

CAUTION:

Please note that setting the environment variables will only apply to your current session (Windows: Terminal Window). If the Terminal Window was closed during the process, you will need to repeat this step.
Also please note that the above values are examples which may differ from your environment.

LINUS / UNIX / MacOS

Under Linux/Unix/MacOS Environment variables should already be set correctly.

STEP 3

Change to the directory you created before

ON WINDOWS

```
cd C:\Users\userABC\hubject_ssl
```

LINUS / UNIX / MacOS

```
cd ~/hubject_ssl
```

STEP 4

Use OpenSSL to create the private key and the CSR.

ON WINDOWS

```
openssl req -config C:\OpenSSL-Win32\bin\openssl.cfg -newkey rsa:2048 -
keyout my_private.key -out my_pkcs10.csr -nodes
```

Please adjust the yellow selection to your preconditions. This means adjusting whether you use OpenSSL in Win32 or Win64 bit version.

LINUS / UNIX / MacOS

```
openssl req -newkey rsa:2048 -keyout my_private.key -out my_pkcs10.csr -
nodes
```

See a listing of all parameters in the examples here:

PARAMETER	EXAMPLE VALUE	DESCRIPTION
req	-	Openssl command to manipulate Certificate Signing Requests
-newkey	RSA:2048	Create a new Key with RSA Algorithm and 2048 bits
-keyout	CompanyName_private.key	Filename of the new key to generate
-out	CompanyName_pkcs10.csr	Filename of the PKCS10 CSR to generate
-nodes	-	Private key will be saved without encryption
-config	C:\OpenSSL-Win32\bin\ openssl.cfg	-

TABLE 3. CREATING YOUR PKCS10 CSR

OpenSSL will now require some user input - there are some requirements which your CSR needs to match - to fill these fields correctly please refer to the table of parameters below.

EXAMPLE

```
Country Name (2 letter code) [AU]: DE
  State or Province Name (full name) [Some-State]: Sample State
  Locality Name (eg, city) []: Sample city
  Organization Name (eg, company) [Internet Widgits Pty Ltd]: Hubject
  Organizational Unit Name (eg, section) []: Example GmbH
  Common Name (e.g. server FQDN or YOUR name) []: example.example.com
  Email Address []: max.muster@example.com
```

You will be asked to fill in now 'extra' attributes to be sent with your certificate request. For this Hubject related request it should be left empty (just press enter).

```
A challenge password [ ]:
  An optional company name [ ]:
```

See a listing of all parameters in the examples here:

PARAMETER	EXAMPLE VALUE	DESCRIPTION
Country Name	DE	Two-letter code of your country name
State or Province Name	Sample province or federal state	Your province or state name
Locality Name	Sample city	City where your company is located
Organization Name (O)	Hubject	Organization name, this value **must be Hubject**
Organizational Unit Name (OU)	Example GmbH	Insert your company name here
Common Name (CN)	example.com	The DNS of your partner backend, or your company's website DNS name in case there is no "real" backend. Alternatively, the name of the person using the certificate can be used.
Email Address	daniel.muster@example.com	The email address of the person responsible for your CSR
Challenge Password		This password is used by some CAs to check for revocation.
An optional company name		As the parameter describes, optional

TABLE 4 REQUIREMENTS OF THE CSR

Now you should have two files within your directory – the example commands create a **YourCompany_private.key** file, and a **YourCompany_pkcs10.csr** file.

If you have set your **RANDFILE environment variable**, you will additionally have generated a .rnd file.

Checking and sending the CSR

Please check the *.csr file before you send it to your onboarding manager using the following command in your command line. Please note that you maybe need to set your environment variables again because they do not persist unless you set them permanently in your system variables.

```
openssl req -noout -text -in my_pkcs10.csr
```

Your command line will show you the complete *.csr. Please check if **the values are set correctly for O and OU as shown above.**

Now you should send only the my_pkcs10.csr file to onboarding@hubject.com. Please create a **password secured ZIP file containing the CSR.** The password associated with the zip file must be sent to onboarding@hubject.com in a separate email. Please only include the .csr file in the password secured ZIP file.

Receiving the certificate

After you sent your PKCS10 CSR via ticket, your CSR will be signed by your Onboarding Manager. Once it has been signed, you will receive access to the HBS QA environment and a ticket with the following attachments:

- Your x509 Certificate, signed by the HubjectCA, in the PEM format
- The trusted CA certificates (HubjectCA) - to be able to verify the Backend certificate, in the PEM format

The CA certificates should be used for establishing the TLS trust between the HBS and the partner backend, as the certificate for the service will be renewed in a regular interval (every two years), which would cause the connection to be untrusted if only the service certificate is directly trusted. The HubjectCA certificate will be renewed only every 10 years. The decision if only the sub-certificate or also the root certificate (or even only the chain of both certificates) need to be trusted needs to be taken by you, as it largely depends on the security obligations and technologies used for TLS.

Now you can convert your certificates and your private key into the format you need.

Converting your certificate

This section describes how to convert your certificate for various uses. All certificates sent to you by Hubject will be plain x.509 Certificates, formatted in the PEM Format (Base64 encoded DER).

Make sure that you have opened your command line and changed to the directory where the certificates and your private key are located. Instructions how to do this are contained in chapter 4 “Establishing a connection”.

Please make sure to have the Hubject certificate on hand (sent to you via ticket by your onboarding manager) before you proceed with this manual. The Root certificate needs to be placed in the same folder you are using to generate your own certificate request.

You may also use a user certificate signed by a certificate authority of your choice – please be sure to follow the instructions in the Help Center for Certificate Management and upload your User Certificate in this section to proceed.

PEM to PKCS12 conversion

PLEASE NOTE THAT ALL COMMANDS HERE ARE EXAMPLES.

Below each command there is a list explaining the different parameters and what should be the value of the parameter - these values will probably be different than the example.

A PKCS12 file contains your **private key**, **your certificate** and the **certificate chain** which belongs to your certificate.

As the Hubject CA directly signs Partner Backend Certificates, the certificate chain only contains this (Hubject CA) certificate.

CAUTION:

Sometimes it can happen that the execution of the command below generates the following error “unable to write "random state". Please make sure you have set your RANDFILE Environment variable as described in chapter 5 “Creating your PKCS10 CSR”, and you have read & write permissions to the file.

WINDOWS

```
openssl pkcs12 -export -in my_x509.crt -inkey my_private.key -out
My_pkcs12.p12
```

LINUS / UNIX / MacOS

```
openssl pkcs12 -export -in my_x509.crt -inkey my_private.key -out
My_pkcs12.p12
```

PARAMETER	EXAMPLE VALUE	DESCRIPTION
pkcs12	-	OpenSSL utility for manipulating PKCS12 files
-export	RSA:2048	You want to export a PKCS12 file
-in	my_x509.crt	Your x.509, PEM formatted certificate you want to convert
-key	my_private.key	The private key you created when creating the CSR for your certificate
-out	My_pkcs12.p12	The file where the new PKCS12 file will be written
-config	C:\OpenSSL-Win32\bin\ openssl.cfg (Windows only)	Choose the openssl config file, for troubleshooting with windows installations of openssl, may be optional

TABLE 5: PARAMETERS FOR CONVERTING TO A PKCS12 FILE

Configure an outgoing partner communication

For each partner, an outgoing communication can be established to an external server. The connection needs to be SSL encrypted. Currently only an outgoing SSL connection with server certificate verification can be supported. To establish an outgoing connection to your desired system Hubject needs to import the Root certificate of the CA signing your server certificate (e.g., Verisign). The Root certificate can be downloaded from the website of the company providing the signed certificate.

Please upload this user certificate in the Certificate Management section of the HBS portal.

If your partner backend is located behind a firewall, you also should allow incoming connections from the Hubject Platform IP Addresses to your Hubject web services. These IP Addresses differ depending on the environment (QA or PROD) you want to use, or you want to be reached by.

Configure Postman for SSL usage

Postman is a free tool used for accessing APIs. It can be downloaded here: **go.postman.co/home**

After installing and opening Postman you are presented with the following screen:

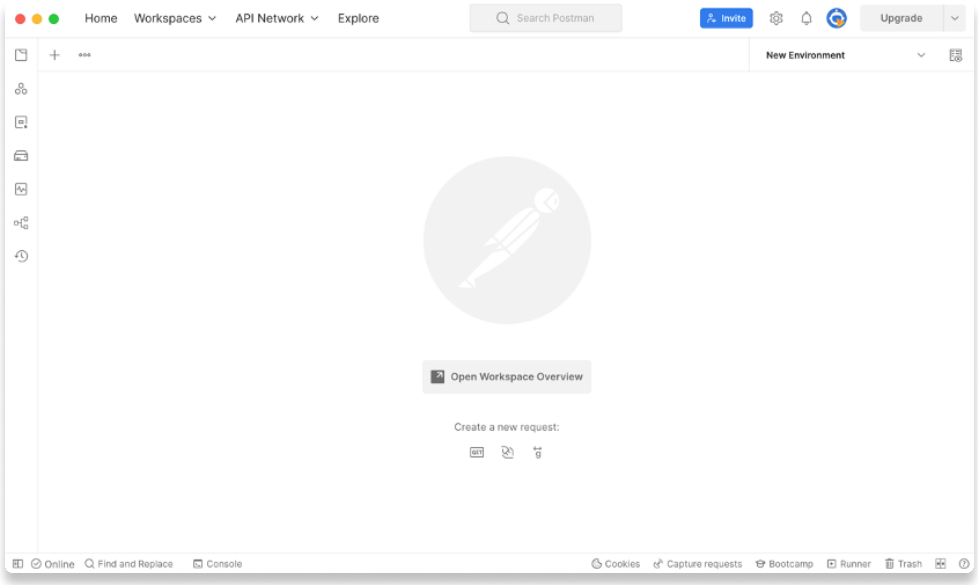

FIGURE 1: POSTMAN START SCREEN

In order to configure SSL, please click on the Settings icon and then select the Settings option.

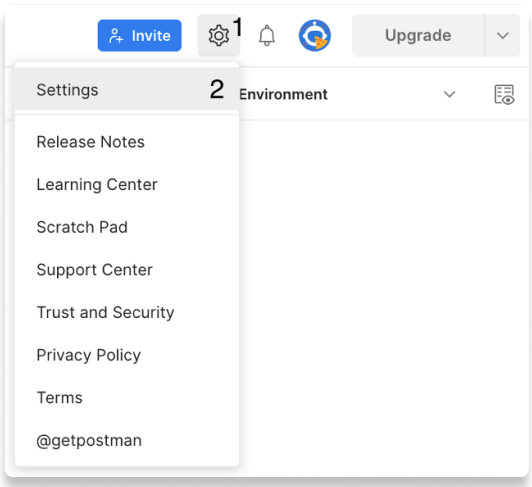

FIGURE 2: POSTMAN SETTINGS MENU

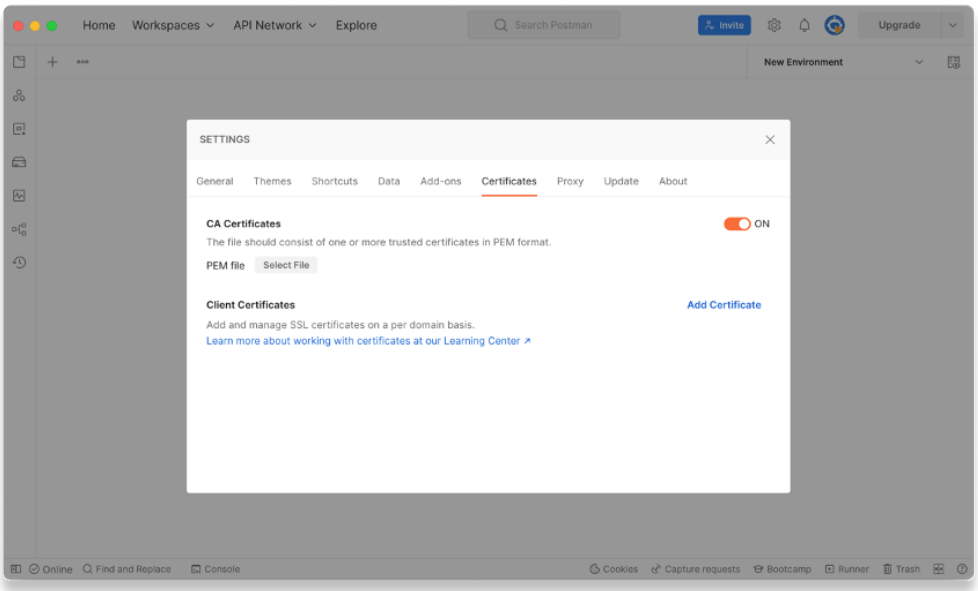

FIGURE 3: POSTMAN CERTIFICATE SETTINGS

After opening the settings, please navigate to the Certificates tab. In the CA Certificate section click on Select File and select your PEM certificate. Next click the Add Certificate button.

You will be presented with the following view:

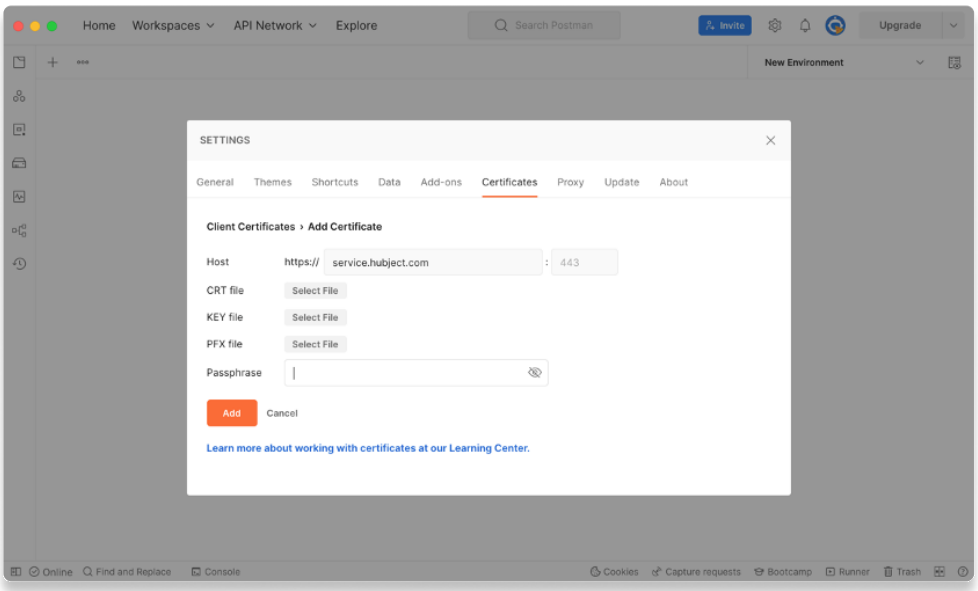

FIGURE 4: POSTMAN CLIENT CERTIFICATE SETTINGS

Please fill the form as follows:

1. **HOST**
    a. If you use the same certificate for production and QA environment you can specify *.hubject.com as host, however this is not recommended for security reasons. Separate certificates should be used for Production and for QA.

b. If you use separate certificates, please use the appropriate host address.

i. QA (acceptance): **service-qa.hubject.com**

ii. PROD (live): **service.hubject.com**

2. PFX File - press Select File and select the PFX file. This is the file that you have created in section 8.1 PEM to PKCS12 conversion, the name of the file is CompanyName_pkcs12.p12. This file contains both private and public keys.

3. Passphrase - enter your passphrase if you have entered it when converting PEM to PKCS12 file

4. CRT file and KEY file should not be selected

5. Click the Add button to save your entry

That is all for the configuration. Postman will use certificates when making the connection to **service.hubject.com** and **service-qa.hubject.com.** Please do not hesitate to contact your onboarding manager at **onboarding@hubject.com** should you have any questions