Certum
by asseco

API User Guide

Version 5.13

1. Spis treści

2. Overview

Certum Partner Program offers a flexible and efficient solution based on SOAP (Simple Object Access Protocol) that allows to place orders for certificates, check the status of their issuance, and at a later stage also manage certificates directly from the partner's system.

Certum Partner API allows to place an order for a certificate of any type (in accordance with the signed partnership agreement) and to monitor the status of the order as it is processed. Certum handles the domain and email address verification process and may contact the Partner's customer if it is necessary to provide additional documents.

In partnership agreement shall be determined such matters as:

- products that the partner may order,
- in the case of personalized certificates – dedicated policies for the partner,
- the content of emails sent automatically by the system in the process of issuing certificates,
- Certum's contact rules with the partner's customers.

2.1. API repository

This documentation is constantly being developed and supplemented with new information and methods added to the API.

The latest version of the documentation and library is always available at:

http://repository.certum.pl/API/

API WSDL:

https://gs.test.certum.pl/service/PartnerApi.wsdl for test environment

https://gs.certum.pl/service/PartnerApi.wsdl for production environment

Additionally, Certum provides a www interface **CertManager** available at the addresses:

https://certmanager.test.certum.pl?language=en for test environment

https://certmanager.certum.pl?language=en for production environment

3. Products available via API

3.1. SSL Certificates

3.1.1. Commercial SSL

Commercial SSL Certificates are certificates offered for 1 year, in the SSL, MultiDomain SSL and Wildcard SSL variants. The issue of the Commercial SSL certificate requires domain access verification. The result of a positive verification will be the automatic issuance of a certificate.

3.1.2. Trusted SSL

Trusted SSL Certificates are certificates offered for 1 year, in the SSL, MultiDomain SSL and Wildcard SSL variants. The issue of the Commercial SSL certificate requires domain access verification and additional verification of the subscriber and the organization.

3.1.3. Premium EV SSL

Premium EV SSL Certificates are certificates offered for 1 year, in the SSL and MultiDomain SSL variants, and do not have Wildcard option available. The issuance of the Premium EV SSL certificate requires domain access verification and additional verification of the subscriber and the organization.

3.2. Certum S/MIME Certificates

3.2.1. Certum S/MIME Mailbox

Certum S/MIME Mailbox certificates are certificates offered in the 1-2 year variant and are always issued for a single email address. They allow to sign and encrypt email. The issuance of the Certum S/MIME Mailbox certificate requires verification of the email address. The result of a positive verification will be the automatic issuance of a certificate.

3.2.2. Certum S/MIME Individual

Certum S/MIME Individual certificates are certificates offered in the 1-2 year variant and are always issued for a single email address. They allow to sign and encrypt email. The issuance of the Certum S/MIME Individual certificate requires verification of the email address and additional verification of the subscriber.

3.2.3. Certum S/MIME Sponsor

Certum S/MIME Sponsor certificates are certificates offered in the 1-2 year variant and are always issued for a single email address. They allow to sign and encrypt email. The issuance of the Certum S/MIME Sponsor certificate requires verification of the email address and additional verification of the subscriber and the organization.

3.2.4. Certum S/MIME Organization

Certum S/MIME Organization certificates are certificates offered in the 1-2 year variant and are always issued for a single email address. They allow to sign and encrypt email. The issuance of the Certum S/MIME Organization certificate requires verification of the email address and additional verification of the organization.

3.3. Code Signing Certificates

3.3.1. Standard Code Signing in the Cloud

Standard Code Signing in the Cloud are certificates offered in the 1-3 year variant. They allow developers to sign executables and recipients to verify the integrity of the software and the identity of the signer. The issuance of the Standard Code Signing in the Cloud certificate requires additional verification of the subscriber and the organization.

3.3.2. EV Code Signing in the Cloud

EV Code Signing in the Cloud are certificates offered in the 1-3 year variant. They allow developers to sign executables and recipients to verify the integrity of the software and the identity of the signer. EV Code Signing Certificates are required to access the Windows Hardware Developer Center Dashboard Portal. The issuance of the EV Code Signing in the Cloud certificate requires strict verification of the subscriber and the organization.

3.4. Document Signing in the Cloud Certificates

Document Signing in the Cloud certificates are certificates offered in the 1-3 year variant. They allow to sign PDF documents. The issuance of the Document Signing in the Cloud certificate requires verification of the email address and additional verification of the subscriber and the organization. In accordance with the applicable AATL regulations, the subscriber's verification must be performed using the F2F or an equivalent method.

3.5. List of product codes

The list of codes available for a given partner is configured individually and depends on the scope of the partnership agreement.

Product name	Validity days	Issuance	Renewal
Commercial SSL	365	601	606
Commercial Wildcard SSL	365	741	746
Commercial MultiDomain SSL 300 Domains	365	931	936
Commercial MultiDomain Wildcard SSL 300 Domains	365	961	966
Trusted SSL	365	631	636
Trusted Wildcard SSL	365	681	686
Trusted MultiDomain SSL 300 Domains	365	921	926
Trusted MultiDomain Wildcard SSL 300 Domains	365	971	976
Premium EV SSL	365	641	646
Premium EV MultiDomain SSL 300 Domains	365	981	986
Certum S/MIME Mailbox	365	501	506
Certum S/MIME Mailbox	730	502	507
Certum S/MIME Individual	365	511	516
Certum S/MIME Individual	730	512	517
Certum S/MIME Sponsor	365	521	526
Certum S/MIME Sponsor	730	522	527
Certum S/MIME Organization	365	531	536
Certum S/MIME Organization	730	532	537
Standard Code Signing in the Cloud	365	831	836
Standard Code Signing in the Cloud	730	832	837
Standard Code Signing in the Cloud	1095	833	838
EV Code Signing in the Cloud	365	316	321
EV Code Signing in the Cloud	730	317	322
EV Code Signing in the Cloud	1095	318	323
Document Signing in the Cloud	365	281	286
Document Signing in the Cloud	730	282	287
Document Signing in the Cloud	1095	283	288

4. Ordering and issuing certificates process

4.1. Information regarding issuing certificates

Certificates that can be ordered via API are divided into several types: DV certificates issued automatically, IV, OV and EV certificates. The types of products are distinguished according to:

- financial guarantees,
- data included in the certificates,
- issuance process,
- requirements to provide additional information on Web Service in order request.

This chapter contains examples of the process of ordering and issuing certificates, as well as other information on placing orders using the API.

4.2. Fields required in certificates

Depending on the type of certificate, different data is required which must be provided in the CSR or the data section for the certificate. The table below lists the required and optional fields for each type of certificate.

R – required
R* – always one of two is required
O – optional
A – filled automatically

Certificate type	CN	GN	SN	O	L	SP	C	E	Org. identifier	SN	BC	ST	P	JoILN	JoISoPN	JoISoCN	SAN
Commercial SSL Commercial Wildcard SSL Commercial MultiDomain SSL Commercial MultiDomain Wildcard SSL	R																R
Trusted SSL Trusted Wildcard SSL Trusted MultiDomain SSL Trusted MultiDomain Wildcard SSL	R			R	R*	R*	R										R
Premium EV SSL Premium EV MultiDomain SSL	R			R	R*	R*	R			R	R	O	O	R*	R*	R	R
Certum S/MIME Mailbox	A							R									A
Certum S/MIME Individual	R	R	R					R									A
Certum S/MIME Sponsor	R	R	R	R	R	O	R	R	A								A
Certum S/MIME Organization	R			R	R	O	R	R	A								A
Standard Code Signing in the Cloud	R			R	R*	R*	R										
EV Code Signing in the Cloud	R			R	R*	R*	R			R	R	O	O	R*	R*	R	
Document Signing in the Cloud	R			R	R*	R*	R	R									

The content of individual fields

Field	Name	Description
CN	commonName	For SSL certificates, it is the first of the domains included in the certificate. For other certificates, it is most often the name and surname of the certificate owner or organization name.
GN	givenName	Given name of the subscriber – first and middle name if applies.
SN	surname	Surname of the subscriber.
O	organization	Name of the organization.
L	locality	Location of the entity for which the certificate is issued.
SP	state	State or province name – location of the entity for which the certificate is issued. The field is additionally validated when C has the value PL.
C	country	Country – location of the entity for which the certificate is issued. The field is validated against a list of allowed values for two-letter ISO codes.
E	email	Email address
Org. Id.	organizationIdentifier	Registration number of the organization or the entity, in accordance with the record required by Baseline, used only in S/MIME certificates.
SN	serialNumber	Registration number of the organization, used only in EV certificates.
BC	businessCategory	Depending on the business category of the entity for which the certificate is issued, used only in EV certificates. The field is validated against a list of allowed values.
ST	streetAddress	Location of the entity for which the certificate is issued, used only in EV certificates.
P	postalCode	Location of the entity for which the certificate is issued, used only in EV certificates.
JoILN	Jurisdiction of Incorporation Locality Name	Location of registration place of the entity for which the certificate is issued, used only in EV certificates.
JoISoPN	Jurisdiction of Incorporation State or Province Name	Location of registration place of the entity for which the certificate is issued, used only in EV certificates. The field is additionally validated when C has the value PL.
JoISoCN	Jurisdiction of Incorporation Country Name	Location of registration place of the entity for which the certificate is issued, used only in EV certificates. The field is validated against a list of allowed values for two-letter ISO codes.
SAN	Subject Alternative Name	All domains included in the certificate are added in this field.

If the PL value is given in the C field or the joISoCN field, the SP and joISoPN fields, respectively, are validated with the following dictionary:

- dolnośląskie
- kujawsko-pomorskie
- lubelskie
- lubuskie
- łódzkie
- małopolskie
- mazowieckie
- opolskie
- podkarpackie
- podlaskie
- pomorskie
- śląskie

- świętokrzyskie
- warmińsko-mazurskie
- wielkopolskie
- zachodniopomorskie

For other values, the C and joISoCN voivodeship fields are not additionally validated.

4.2.1. The uniqueness of the customer field in the order

Certum distinguishes end users by the **customer** field. This means that each order from a different end user should be identified with a different value of the **customer** field. The field does not have to contain the actual login, it can be an order number from the partner's external system, or another identifier, unique for a given end user.

In the case of in the Cloud products, the **customer** field must contain the email address that is the user's login to the SimplySign service.

The field **customer** cannot be the same as partner login.

Note: If one value is used for the **customer** field, all orders will be treated as orders placed by the same user, which may affect the certificate issuance process.

4.2.2. The validation of commonName field in the order

Certificates in the **commonName** field must match the other fields in **quickOrder**:

- All SSL products: CN=one of **SANEntries** values,
- S/MIME Individual: CN = **givenName+" "+surname**,
- S/MIME Sponsor: CN = **givenName+" "+surname**,
- S/MIME Organization: CN=**organization**,
- Standard Code Signing in the Cloud: CN= **organization**,
- Standard Code Signing in the Cloud: CN=**firstName+" "+lastName**
- EV Code Signing in the Cloud: CN = **organization**,
- Document Signing in the Cloud: CN = **firstName+" "+lastName** or CN = **organization**.

4.2.3. Domain verification methods for SSL certificates

Certum provides three methods of automatic domain verification from an SSL certificate:

1. Verification of the domain name with the email address of the domain administrator
   - Method name in API: ADMIN
   - Allowed email addresses to which the verification link can be sent are: admin@*domain.com*, administrator@*domain.com*, hostmaster@*domain.com*, webmaster@*domain.com*, postmaster@*domain.com*. Please ensure that one of these email addresses is created for each domain you are verifying.
   - E-mails are sent to addresses created on the basis of the prefix specified in the **approverEmailPrefix** element and the list of domains in the order.
   - As many verification emails as the domains specified in the order will be sent.
   - Sending verification emails for the ADMIN method can not be disabled.
   - To complete the verification using the domain administrator's email address the link in the email should be used.

2. Verification of the domain name with file placed on the server
   - Method name in API: FILE
   - Limitations: must not be used to issue Wildcard SSL certificates and must not be used as Authorization Domain Names for subordinate FQDNs of the validated FQDN

- Email is sent to the address that specified in **approverEmail**.
- One email with one verification code will be sent for all domains in the order.
- In order to FILE verification messages not be sent, sending verification emails can be disabled in the partner's account.
- When email sending is disabled, **quickOrder** returns a verification code.
- The email contains the name of the file, the verification code to be included in the file, and the link to be clicked after performing the described steps.
- To complete the verification, the file with the specified name, the content of which must contain the verification code with **-certum.pl** suffix should be placed in the directory: **/.well-known/pki-validation** of the verified domain, and then link from the email should be used or call **perfomSanVerification** should be performed.

3. Verification of the domain name with record placed in DNS record
    - Method name in API:
        - DNS_TXT – place the code in the TXT record for the domain name
        - DNS_CNAME – place the code with **.certum.pl** suffix in the CNAME record for the domain name
        - DNS_TXT_PREFIX – place the code in the TXT record for the domain name preceded by the prefix **_certum** (e.g. _certum.yourdomain.pl)
        - DNS_CNAME_PREFIX – place the code with **.certum.pl** suffix in the CNAME record for the domain name preceded by the prefix **_certum** (e.g. _certum.yourdomain.pl)
    - Limitations: must not be used to verify IP addresses.
    - Email is sent to the address that specified in **approverEmail**.
    - One email with one verification code will be sent for all domains in the order.
    - In order to DNS verification messages not be sent, sending verification emails can be disabled in the partner's account.
    - When email sending is disabled, **quickOrder** returns a verification code.
    - The received email contains a verification code and a link to be clicked after creating the DSN record.
    - To complete the verification, the verification code should be placed in DSN record according to the selected method name, and then link from the email should be used or call **perfomSanVerification** should be performed.
    - Please note that propagation time for DNS changes may take up to 24 hours.

Note: In some cases, e.g. in the case of popular domains or domains for institutions such as a bank, Certum may request additional documents for a more complete verification of the order.

4.2.4. The subscriber and the organization verification methods

After placing an order for the certificate, the system sends an order confirming notifications with information on the next steps according to the product for which the order was placed – the notifications is sent to the email address provided in **orderParameters/email** or **requestorInfo/email**. In order to confirming notifications not be sent, sending informational emails can be disabled in the partner's account.

Based on the data in the **organizationInfo** section, the organization's data included in the certificate will be verified. Information about the organization is verified in publicly available registers, eg KRS, GUS, CEiDG, DUNS. This step is performed by Certum and does not require the subscriber's involvement. If the organization is not listed, a valid company registration document must be provided using **verifyOrder**, specifying **ORGANIZATION** as the document type.

Based on the data in the **requestorInfo** section, the subscriber's identity will be confirmed. Confirmation is carried out using identity documents or the ARIADNEXT system. In the case of ARIADNEXT, an email message will be sent to the address provided in **requestorInfo** with a link allowing to start automatic identity confirmation. In case of document-based verification, the valid document confirming the subscriber's identity should be provided using **verifyOrder**, specifying **APPLICANT** as the document type.

Additionally, if the person applying for the certificate is not authorized to represent the institution on his own, a valid work certificate or authorization should be provided using **verifyOrder**, specifying **AUTHORIZATION** as the document type.

In justified cases, the Certum team may ask for additional documents necessary for proper verification. Documents should be provided using **verifyOrder**, specifying **ADDITIONAL** as the document type.

To check documents verification status, the **getDocumentsList** method is available, which returns a list of documents with their statuses and their expiry dates.

To check the subscriber and the organization verification status, the **getOrderState** method is available, which returns the statuses related to the verification of documents, respectively **ORGANIZATION**, **APPLICANT** and **AUTHORIZATION**. If verification has been made based on the documents provided, the status will change from **REQUIRED** to **VERIFIED**.

4.2.5. Additional configuration options

In order for a partner to be able to use the API, it is necessary to configure the following account data in the Certum system:

- The IP address from which the partner connects.
- Products codes that the partner may order.
- Default language for sending notifications.
- Configuration of sending verification emails for DNS and FILE verification methods – all emails are digitally signed.
    - It is possible to disable sending verification emails for DNS and FILE verifications of the subscriber's control over the domain for individual orders, using the **verificationNotificationEnabled** parameter in the **SanApprovers** element, but this parameter must be set for each order separately.
- Configuration of sending informational emails – all emails are digitally signed.
    - Order confirmation – sent after the order is placed in the Certum system.
    - Incomplete order verification – email reminding the end user about the pending order verification, sent once 23 days after placing the order. Certification Practice Statement specifies the time after which the order will be rejected if the subscriber has not complied with the formalities.
    - Certificate issuance – sent after certificate is issued.

- Certificate revocation– sent after certificate is revoked.
- Certificate expiration – email reminding the end user about certificate expiration date (it is sent 30, 14, 7 and 1 day before certificate expiration). Certificate renewal disables the sending of emails reminding about the expiry.
- Certificate expired – sent after certificate is expired.

Note: If the sending of emails by Certum is disabled, the partner is obliged to inform the subscriber about the above activities on his own, but remember that if the partner sends emails to its subscribers, these messages, in accordance with the requirements of WebTrustSM / TM 2.0, **must be digitally signed.**

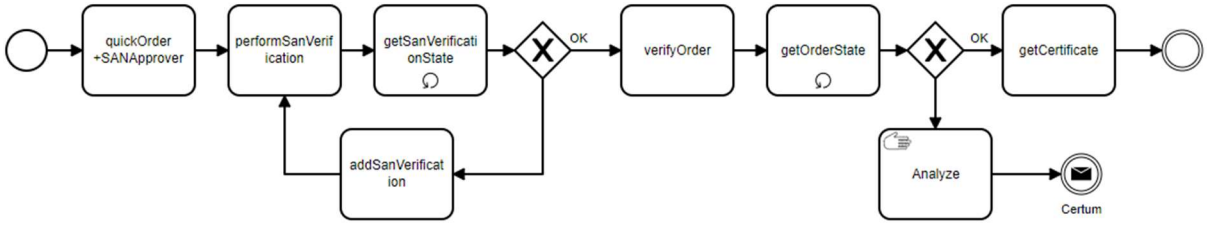
Additional elements, the configuration of which is optional (if they are not configured standard Certum templates are used)

- Dedicated header and footer of email messages sent by the Certum system.
- Dedicated content of email messages sent by the Certum system.

4.3. Trusted SSL and Premium EV SSL certificates ordering process

4.3.1. Placing an order

For Trusted SSL and Premium EV SSL certificates, the **`SanApprover`** and **`SANEntries`** as well as **`requestorInfo`** and **`organizationInfo`** sections must be present in the **`quickOrder`** request.



4.3.2. Completing verifications

After placing an order for a certificate, the system sends two types of emails:

- order confirming notification with information on the next steps in the verification of the subscriber and the organization,
- verification email for the selected method: ADMIN, DNS_TXT, DNS_CNAME, DNS_TXT_PREFIX, DNS_CNAME_PREFIX or FILE as described in the domain verification for SSL certificates.

For DNS and FILE methods, the obtained code should be placed in the DNS configuration or in a file on the server, respectively, and then the verification should be initiated by calling **`perfomSanVerification`**. For the ADMIN method, link in the email should be used.

The domain verification status for an order can be checked by calling **`getSanVerificationState`**. The method returns not only the verification status for each domain, but also information about what problems have occurred. If the response contains verification errors, their cause must be removed and then **`getSanVerificationState`** can be called again.

In case of verification problems, code expiration, or a verification method change, new code can be generated by calling **`addSanVerification`**.

The required documents should be provided with **`verifyOrder`** method.

4.3.3. Obtaining a certificate

Using the **`getOrderState`** method, verifications required for the order and their status can be monitored. When the order status changes to **`ENROLLED`**, the issued certificate can be obtained.

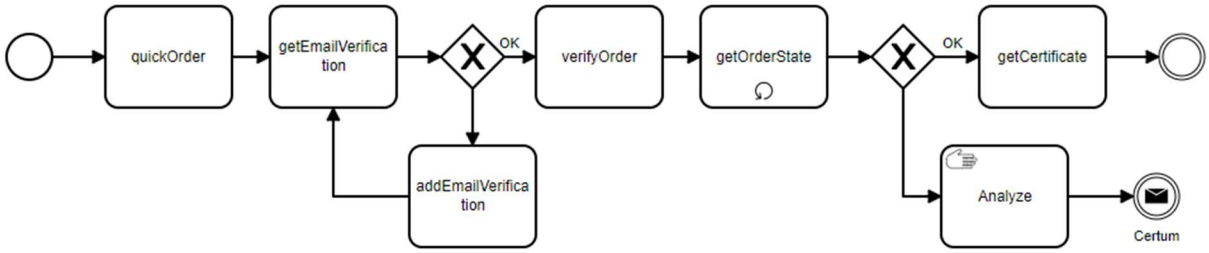
In the case of SSL certificates, domain verification will be required, separated into two verifications: **`SYSTEM`** and **`DOMAIN`**. **`SYSTEM`** verification is carried out automatically by Certum systems and does not need to be called via API. **`DOMAIN`** verification is the verification of domains in the certificate discussed earlier. With the **`getSanVerificationState`** method, problems that occur during both verifications can be diagnosed.

Apart from the certificate issued for the domain, remember to install the root certificates (rootCA) and intermediate certificates (subCA) on the server – using **`getCertificate`** it is possible to obtain all these certificates in PEM format.

4.4. Certum S/MIME Sponsor certificates ordering process

4.4.1. Placing an order

For Certum S/MIME Sponsor certificates, the **requestorInfo** and **organizationInfo** sections must be present in the **quickOrder** request. The presence of **SanApprover** and **SANEntries** sections will result in an error.



4.4.2. Completing verifications

After placing an order for a certificate, the system sends two types of emails:

- order confirming notification with information on the next steps in the verification of the subscriber and the organization,
- verification email for the address provided in the **orderParameters/email** field.

To complete the verification of the email address link in the email should be used. It is not possible to disable the sending of verification emails for the E (email) field.

The email verification status for an order can be checked by calling **getEmailVerification.**

In case of problems with verification, the email can be sent again by calling **addEmailVerification**.

The required documents should be provided with **verifyOrder** method.

4.4.3. Obtaining a certificate

Using the **getOrderState** method, verifications required for the order and their status can be monitored. When the order status changes to **ENROLLED**, the issued certificate can be obtained.

The issued certificate can be obtained using **getCertificate**.

4.5. Standard Code Signing in the Cloud certificates ordering process

The process is similar to the process for Certum S/MIME Sponsor, with the difference that in the Standard Code Signing in the Cloud certificate there is no email field, so its verification will not be needed.

This is in the Cloud product, so the **`customer`** field must contain the email address that is the user's login to the SimplySign service.

Using the **`getOrderState`** method, verifications required for the order and their status can be monitored. When the order status changes to **`ENROLLED`**, the issued certificate can be obtained. In the case of Standard Code Signing in the Cloud certificates, if **`EMAIL`** verification is not required, the status **`NOT_REQUIRED`** will be returned for this field.

Standard Code Signing in the Cloud certificates in can not be issued automatically.

4.6. Document Signing in the Cloud certificates ordering process

The process is similar to the process for Certum S/MIME Sponsor.

This is in the Cloud product, so the **`customer`** field must contain the email address that is the user's login to the SimplySign service.

Using the **`getOrderState`** method, verifications required for the order and their status can be monitored. When the order status changes to **`ENROLLED`**, the issued certificate can be obtained. For Document Signing in the Cloud certificates **`EMAIL`** verification is required.

Document Signing in the Cloud certificates in can not be issued automatically.

In the case of Document Signing in the Cloud certificates, the identity of the subscriber must be verified by the F2F method or an equivalent method, it is not possible to verify the subscriber's identity on the basis of documents.

5. WebService overview

5.1. Placing an order

5.1.1. Retrieving the list of available products

Use **`getProductList`** to retrieve a list of available product codes, along with detailed product information.

5.1.2. Ordering new certificate

The **`quickOrder`** provides all the information necessary to place an order:

- customer identifier,
- product code,
- CSR and data for the certificate
- data allowing verification of the subscriber and the organization in the case of IV, OV and EV certificates.
- in the SAN (Subject Alternative Name) extension, it is possible to include multiple domains in single certificate (available multidomain certificates), and the www option for certificates securing one domain.
- one verification method that applies to all domains included in the order.

In the case of all multidomain certificates, all domains to be included in the SANEntry extension must be explicitly provided, the system for such certificates does not automatically add any additional entries.

5.1.3. Verification of the correctness of data in the order

Use **`validateOrderParameters`** to validate all data included in the order. The subject to verification are the completeness of the data contained in the CSR with the certificate type and the scope of the data provided. In particular, the following are verified:

Partner login:

- correct login and password,
- account activity.

Order:

- whether the order ID is unique in the database and correctly constructed,
- whether the customer ID is provided,
- whether the given product is available for the partner,
- whether the expiry date of the certificate is within the number of days assigned to a given product code, if no date range is given, the certificate will be issued with a starting date equal to the issue date and the maximum end date for a given product.

CSR:

- whether the key does not appear on the blacklist or on the list of used keys,
- whether all required fields are filled – the required fields are defined within the certificate type,,
- whether the fields are in the allowed format,
- whether the CN field matches one of the allowed CN values for the product,
- verification of the correctness of data related to the SAN extension,
- in response, the method returns the data that will be placed in the certificate, taking into account both the data from CSR and the data from the request overwriting the data from CSR.

5.1.4. Reissuing the certificate

Reissuing is available for all certificates during their validity period. Use **`reissueCertificate`** to reissue the certificate with new keys and the same data with the end date of the original certificate. As part of the reissue, a new domain can be added, with all new domains requiring re-verification. If no new domains have been added, the reissue process is automatic.

The issue of a new certificate as a result of a reissue will automatically revoke the original certificate 14 days after issue. Consequently, the subscriber always has one valid certificate.

5.1.5. Renewing the certificate

Use **`renewCertificate`** to place an order to renew, that is. to issue a new certificate with new keys and the same data and with the new validity period. The data in the new certificate will come from the renewed certificate and cannot be changed while placing an order for renewal.

5.1.6. Retrieving the order verification status

Use **`getOrderState`** to retrieve verification details for the provided order ID. Based on the order ID provided, the following can be determined:

- whether the verification of all domains was successfully completed,
- whether it is necessary to verify the email address,
- whether the data of the subscriber and the organization are subject to verification in a given order,
- whether the certificate has been issued.

5.1.7. Retrieving order data

Use **`getOrderByOrderID`** to retrieve a single order for a given order ID. Optionally additional data about the order and related certificates can be retrieved.

5.1.8. Cancelling order

Use **`cancelOrder`** to cancel a placer order. Only an order placed from the same partner account can be canceled. The order will be canceled if the certificate has not been issued. If the certificate has been issued, the order status cannot be changed, the issued certificate should be revoked. Certificate revocation does not cancel the order.

5.2. Domain verification – SSL certificates

5.2.1. Retrieving the detailed domain verification status

Use **`getSanVerificationState`** to retrieve verification statuses of all domains for a given order along with information about errors on end user-side verification, and additional problems with the domain that may block the issuance of a certificate, such as an incorrect CAA entry or the presence of a domain on Phishing lists. In case of positive verification, additional information will not be returned.

5.2.2. Generating new domain verifications

Use **`addSanVerification`** to generate new verification codes. The method allows to create any number of verifications of the same type, as well as create a new type of verifications for an order. Creating a new verification code will not deactivate the previous codes.

5.2.3. Initiating domain verification for the order

Use **`performSanVerification`** to start the process of asynchronous verification of all domains from the order for the DNS and FILE methods. To retrieve information about the verification result, use the **`getSanVerificationState`** method.

5.3. E (email) field verification – S/MIME and Document Signing in the Cloud certificates

5.3.1. Sending a verification email for the E (email) field

Use **`addEmailVerification`** to send an email with a verification link to the address from the E (email) field in the certificate. The method allows sending any number of verifications. The email field verification applies to S/MIME and Document Signing in the Cloud certificates.

5.3.2. Retrieving the E (email) field verification status

Use **`getEmailVerification`** to retrieve verification status for the E (email) field placed in the certificate. The email field verification applies to S/MIME and Document Signing in the Cloud certificates.

5.4. Subscriber and organization verification – IV, OV and EV certificates

5.4.1. Adding documents

Use **`verifyOrder`** to add to the order documents that will enable the order to be verified and the certificate to be issued. Documents may be required when placing an order for an IV, OV or EV certificate.

The document adding method is limited by several conditions:

- Documents can be added only to a placed order, cannot add a document not related to any order.
- During one method call, it is possible to add document consisting of several files (e.g. each page of a document scanned in a separate file).
- It is not possible to remove or change previously added documents and files.
- For new orders and renewed certificates, documents and information from public registers cannot be older than 13 months.
- Authorization documents, whether timely or not, remain valid for 13 months from the date of issue.

5.5. Status of the issued certificate

5.5.1. Obtaining certificate

Use **`getCertificate`** to obtain the certificate in PEM format. In addition to the user certificate, all intermediate certificates (subCA) and the root certificate (rootCA) are also returned in PEM format. If reissue certificates exist, the most recent active certificate for the given order ID is returned.

5.5.2. Revoking certificate

Use **`revokeCertificate`** to revoke a certificate. Only certificates generated from the same partner account can be revoked. The certificate may be revoked during its validity period.

5.6. Reports

5.6.1. Report of orders placed in a given period of time

Use **`getOrdersByDateRange`** to retrieve information about orders and certificates (if issued) placed in a given period of time. Optionally additional data about the order and related certificates can be retrieved. The results are paged, information on a maximum of 100 orders is returned on one page.

5.6.2. Report of orders modified in a given period of time

Use **`getModifiedOrders`** to retrieve information about orders and certificates (if issued), the status of which has changed in a given period. Optionally additional data about the order and related certificates can be retrieved. The results are paged, information on a maximum of 100 orders is returned on one page.

5.6.3. Report of expiring certificates

Use **`getExpiringCertificates`** to retrieve a list of certificates expiring in the near future – certificates expiring in the range of 1-30 days.

6. WebService structure

The following designation is used in the documentation below:

```
Required fields - the minimum data set for the request and response
Optional fields
Fields added in the current version
Fields that will be removed in future versions
~~Fields that have been removed in the current release~~
```

6.1. Requests headers

Each request sent to the API requires authorization data such as login and password.

```
<requestHeader>
  <authToken>
    <userName>255 String
    <password>255 String
  </authToken>
</requestHeader>
```

Field name	Req.	Type	Description
userName	YES	255 String	Partner's identifier agreed with Certum, SSO account.
password	YES	255 String	The password to match the SSO password.

Each response returns a header with a code confirming the execution of the operation or error codes allowing to identify the problem. Results paging data only applies to requests that return reports.

In case of errors, the data described in individual Response for API methods will not be returned. The list of error codes is in a separate chapter

```
<responseHeader>
  <successCode>3
  <errors>
    <error>
      <errorCode>
    </error>
  </errors>
  <currentPage>1..100
  <pagesCount>1..100
  <returnCount>5
  <timestamp>YYYY-MM-DDTHH:MM:SS.000Z
</responseHeader>
```

Field name	Req.	Type	Description
successCode	YES	3	Code 0 means that the order has been accepted correctly. Code 1 and 3 indicate an error.
errorCode	NO	5 String	Error codes that occurred while executing the request.
currentPage	NO	1..100	Information on which page of results is being returned.
pagesCount	NO	1..100	Information on the total number of pages.
returnCount	NO	5	Information about the total number of records if more than one record is returned as a result.
timestamp	YES	Timestamp	The date and time of the response.

6.2. getProductListRequest

The request allows to download a list of products configured for a given account.

```
<getProductList>
  <requestHeader>
    <authToken>
      <userName>255 String
      <password>255 String
    </authToken>
  </requestHeader>
  <hashAlgorithm>true, false
</getProductList>
```

Field name	Req.	Type	Description
hashAlgorithm	NO	true, false	No value is equivalent to setting false. Returns additional information about the available hash functions for the product.

6.3. getProductListResponse

The response returns information about the available products and their configuration details.

```
<getProductListResponse>
  <responseHeader>
    <successCode>3
    <errors>
      <error>
        <errorCode>5
      </error>
    </errors>
    <timestamp>YYYY-MM-DDTHH:MM:SS.000Z
  </responseHeader>
  <products>
    <product>
      <code>3 String
      <type>ISSUE, RENEWAL
      <validityPeriod>4
      <certificateNotificationEnabled>true, false
      <verificationNotificationEnabled>true, false
      <supportedHashAlgorithms>
        <hashAlgorithm>RSA-SHA256, ECC-SHA256
      </supportedHashAlgorithms>
    </product>
  </products>
</getProductListResponse>
```

Field name	Req.	Type	Description
code	YES	3 String	3-digit product code.
type	YES	List	ISSUE – products used in quickOrder, RENEWAL – products used in renewCertificate .
validityPeriod	YES	Liczba	certificate validity period in days
certificateNotificationEnabled	YES	true/false	Information whether information emails are sent.
verificationNotificationEnabled	YES	true/false	Information whether verification emails for DNS and FILE methods are sent.
hashAlgorithm	YES	Timestamp	RSA-SHA256, ECC-SHA256 – hash functions available for the product.

6.4. quickOrderRequest

The request should contain all the data needed to place the order, such as CSR in the form of PKCS#10, product code, data of the person ordering the certificate and other data required by a specific type of certificate, such as verification data for SSL certificates.

```
<quickOrder>
  <requestHeader>
    <authToken>
      <userName>255 String
      <password>255 String
    </authToken>
  </requestHeader>
  <orderParameters>
    <customer>64 String
    <orderID>50 String
    <userAgent>255 String
    <language>2 String
    <revocationContactEmail>255 String
    <productCode>3 String
    <CSR>4000 String
    <hashAlgorithm> RSA-SHA256, ECC-SHA256
    <shortenedValidityPeriod>25 YYYY-MM-DD
    <email>64 String
    <commonName>64 String
    <givenName>16 String
    <surname>40 String
    <organization>64 String
    <organizationalUnit>64 String
    <locality>128 String
    <state>128 String
    <country>2 String
    <serialNumber>64 String
    <businessCategory>Private Organization, Business Entity, Non-Commercial Entity,
Government Entity
    <streetAddress>64 String
    <postalCode>40 String
    <joILN>128 String
    <joISoPN>128 String
    <joISoCN>2 String
  </orderParameters>
  <SANEntries>
    <SANEntry>
      <DNSName>230 String
    </SANEntry>
  </SANEntries>
  <SANApprover>
    <approverMethod>ADMIN, FILE, DNS_TXT, DNS_CNAME, DNS_TXT_PREFIX, DNS_CNAME_PREFIX
    <approverEmail>255 String
    <approverEmailPrefix>ADMIN, ADMINISTRATOR, POSTMASTER, HOSTMASTER, WEBMASTER
    <verificationNotificationEnabled>true, false
  </SANApprover>
  <requestorInfo>
    <email>255 String
    <firstName>16 String
    <lastName>40 String
    <phone>32 String
  </requestorInfo>
  <organizationInfo>
    <taxIdentificationNumber>32 String
  </organizationInfo>
</quickOrder>
```

Product and customer data – this is the customer identifier and product code identifier, always one product and additional data.

```
  <orderParameters>
    <customer>64 String
    <orderID>50 String
    <userAgent>255 String
    <language>2 String
    <revocationContactEmail>255 String
    <productCode>3 String
  </orderParameters>
```

Field name	Req.	Type	Description
customer	YES	64 String	Customer ID or customer login for SimplySign service.
orderID	NO	50 String	The unique identifier of the order that will be used by the partner. If not provided, it will be automatically assigned by API.
userAgent	NO	255 String	Browser and operating system information.
language	NO	2 String	Language for mailings, if different from the default language set for the partner.
revocationContactEmail	NO	255 String	Email address that will be used to notify the client only in the event of certificate revocation as a result of reporting non-compliance. The given address cannot be the partner's login.
productCode	YES	3 String	3-digit product code.

Data for the certificate – data to be included in the certificate. The required fields depend on the selected product. When specifying CSR, the missing fields can be completed and incorrect data overwritten with additional fields included in the request.

```
  <orderParameters>
    <CSR>4000 String
    <hashAlgorithm> RSA-SHA256, ECC-SHA256
    <shortenedValidityPeriod>25 YYYY-MM-DD
    <email>255 String
    <commonName>64 String
    <givenName>16 String
    <surname>40 String
    <organization>64 String
    <organizationalUnit>64 String
    <locality>128 String
    <state>128 String
    <country>2 String
    <serialNumber>64 String
    <businessCategory>Private Organization, Business Entity, Non-Commercial Entity,
Government Entity
    <streetAddress>64 String
    <postalCode>40 String
    <joILN>128 String
    <joISoPN>128 String
    <joISoCN>2 String
  </orderParameters>
```

Field name	Req.	Type	Description
CSR	YES	4000 String	Certification request in PKCS#10 format.
hashAlgorithm	NO	List	RSA-SHA256, ECC-SHA256 – If not provided, the default value configured for the partner product will be used. The available hash functions can be obtained by calling the getProductList method

shortenedValidityPeriod	NO	YYYY-MM-DD	Certificate expiration date, must be less than the current date + the number of days resulting from the product (e.g. current date + 364 days for a 365 certificate). Set this value to expire the certificate on a specific date.
email	NO	64 String	Overwriting or filling in the field E from CSR.
commonName	NO	64 String	Overwriting or filling in the field CN from CSR.
givenName	NO	16 String	Overwriting or filling in the field GN from CSR.
surname	NO	40 String	Overwriting or filling in the field SN from CSR.
organization	NO	64 String	Overwriting or filling in the field O from CSR.
locality	NO	128 String	Overwriting or filling in the field L from CSR.
state	NO	128 String	Overwriting or filling in the field SP from CSR.
country	NO	2 String	Overwriting or filling in the field C from CSR.
serialNumber	NO	64 String	Overwriting or filling in the field SN from CSR.
businessCategory	NO	List	Private Organization, Business Entity, Non-Commercial Entity, Government Entity – Overwriting or filling in the fieldBC from CSR.
streetAddress	NO	64 String	Overwriting or filling in the field ST from CSR.
postalCode	NO	40 String	Overwriting or filling in the field P from CSR.
joILN	NO	128 String	Overwriting or filling in the field JoILN from CSR.
joISoPN	NO	128 String	Overwriting or filling in the field JoISoPN from CSR.
joISoCN	NO	2 String	Overwriting or filling in the field JoISoCN from CSR.

Note: There is a validation for the **commonName** field. The field must have values that comply with the retention rules in section 4.2.2. If the data does not match, the API will return an error.

Data for SSL certificate – a list of domains along with the selected verification method. Domains are not loaded from CSR, they must be specified separately in the request. These data are required when placing an order for an SSL certificate.

```
<SANEntries>
  <SANEntry>
    <DNSName>230 String
  </SANEntry>
</SANEntries>
<SANApprover>
  <approverMethod>ADMIN, FILE, DNS_TXT, DNS_CNAME, DNS_TXT_PREFIX, DNS_CNAME_PREFIX
  <approverEmail>255 String
  <approverEmailPrefix>ADMIN, ADMINISTRATOR, POSTMASTER, HOSTMASTER, WEBMASTER
  <verificationNotificationEnabled>true, false
</SANApprover>
```

Field name	Req.	Type	Description
DNSName	YES	230 String	Any domain that is to be included in the certificate.
approverMethod	YES	List	ADMIN, FILE, DNS_TXT, DNS_CNAME, DNS_TXT_PREFIX, DNS_CNAME_PREFIX – one of the available verification methods.
approverEmail	NO	255 String	Email to which the verification code will be sent – if the DNS or FILE method is selected.
approverEmailPrefix	NO	List	ADMIN, ADMINISTRATOR, POSTMASTER, HOSTMASTER, WEBMASTER – if the ADMIN method has been selected, select the prefix to which the verification emails will be sent.
verificationNotificationEnabled	NO	true/false	FALSE – Parameter that disables sending verification emails, if the DNS or FILE method is selected. For the ADMIN method, verification emails are always sent. No value means that the

	default value configured for the partner is accepted. The current configuration can be obtained by calling the getProductList method

Subscriber's data – required to verify the person placing the order. These data are required when placing an order for an OV or EV certificate.

```
<requestorInfo>
  <email>230 String
  <firstName>16 String
  <lastName>40 String
  <phone>32 String
</requestorInfo>
```

Field name	Req.	Type	Description
email	YES	230 String	Subscriber's email address.
firstName	YES	16 String	Subscriber's first name.
lastName	YES	40 String	Subscriber's last name.
phone	NO	32 String	Subscriber's phone number.

Organization data – required to verify the organization whose data is in the certificate. These data are required when placing an order for an OV or EV certificate.

```
<organizationInfo>
  <taxIdentificationNumber>32 String
</organizationInfo>
```

Field name	Req.	Type	Description
taxIdentificationNumber	YES	32 String	Tax identification number or company identifier, such as DUNS.

6.5. quickOrderResponse

The response returns the confirmation of the order with its number. If domain verification is required, it returns additional information about that verification.

```
<quickOrderResponse>
  <responseHeader>
    <successCode>3
    <errors>
      <error>
        <errorCode>5
      </error>
    </errors>
    <timestamp>YYYY-MM-DDTHH:MM:SS.000Z
  </responseHeader>
  <orderID>50 String
  <SANVerification>
    <approverMethod>FILE, DNS_TXT, DNS_CNAME, DNS_TXT_PREFIX, DNS_CNAME_PREFIX
    <code>50 String
    <approverEmail>255 String
    <FQDNs>
      <FQDN>230 String
    </FQDNs>
  </SanVerification>
</quickOrderResponse>
```

Field name	Req.	Type	Description
orderID	YES	50 String	The unique identifier of the order that will be used by the partner. If not provided, it will be automatically assigned by API.
approverMethod	NO	List	FILE, DNS_TXT, DNS_CNAME, DNS_TXT_PREFIX, DNS_CNAME_PREFIX – selected verification method. The verification data is not returned if the ADMIN method is selected.
code	NO	50 String	Verification code.
approverEmail	NO	255 String	Email to which the verification code will be sent – if the DNS or FILE method is selected.
FQDN	NO	230 String	Domain for which verification is required.

6.6. validateOrderParametersRequest

The request allows the data to be submitted for validation. The structure of the entire request is the same as in the **`quickOrder`**. The compliance of the data contained in the CSR with the certificate profile and the scope of the data provided are checked. The same validators are used when placing an order using the **`quickOrder`** method.

```
<validateOrderParameters>
  <requestHeader>
    <authToken>
      <userName>255 String
      <password>255 String
    </authToken>
  </requestHeader>
  <orderParameters>
    <customer>64 String
    <orderID>50 String
    <userAgent>255 String
    <language>2 String
    <revocationContactEmail>255 String
    <productCode>3 String
    <CSR>4000 String
    <hashAlgorithm>RSA-SHA256, ECC-SHA256
    <shortenedValidityPeriod>25 YYYY-MM-DD
    <email>64 String
    <commonName>64 String
    <givenName>16 String
    <surname>40 String
    <organization>64 String
    <organizationalUnit>64 String
    <locality>128 String
    <state>128 String
    <country>2 String
    <serialNumber>64 String
    <businessCategory>Private Organization, Business Entity, Non-Commercial Entity,
Government Entity
    <streetAddress>64 String
    <postalCode>40 String
    <joILN>128 String
    <joISoPN>128 String
    <joISoCN>2 String
  </orderParameters>
  <SANEntries>
    <SANEntry>
      <DNSName>230 String
    </SANEntry>
  </SANEntries>
  <SANApprover>
    <approverMethod>ADMIN, FILE, DNS_TXT, DNS_CNAME, DNS_TXT_PREFIX, DNS_CNAME_PREFIX
    <approverEmail>255 String
```

```
    <approverEmailPrefix>ADMIN, ADMINISTRATOR, POSTMASTER, HOSTMASTER, WEBMASTER
    <verificationNotificationEnabled>true, false
  </SANApprover>
  <requestorInfo>
    <email>255 String
    <firstName>16 String
    <lastName>40 String
    <phone>32 String
  </requestorInfo>
  <organizationInfo>
    <taxIdentificationNumber>32 String
  </organizationInfo>
</validateOrderParameters>
```

6.7. validateOrderParametersResponse

The response returns the results of validation of the request. The correct answer returns the data retrieved from the CSR and the fields in the request in that will be added to the certificate, the remaining fields are omitted.

```
<validateOrderParametersResponse>
  <responseHeader>
    <successCode>3
    <errors>
      <error>
        <errorCode>5
      </error>
    </errors>
    <timestamp>YYYY-MM-DDTHH:MM:SS.000Z
  </responseHeader>
  <parsedCSR>
    <email>64 String
    <commonName>64 String
    <givenName>16 String
    <surname>40 String
    <organization>64 String
    <organizationalUnit>64 String
    <locality>128 String
    <state>128 String
    <country>2 String
    <serialNumber>64 String
    <businessCategory>Private Organization, Business Entity, Non-Commercial Entity,
Government Entity
    <streetAddress>64 String
    <postalCode>40 String
    <joILN>128 String
    <joISoPN>128 String
    <joISoCN>2 String
  </parsedCSR>
</validateOrderParametersResponse>
```

6.8. reissueCertificateRequest

The request should contain all the data needed to reissue the certificate such as CSR with new keys in the form of PKCS#10 and optional new domains. To reissue the latest valid certificate must be provided, either by attaching its file in PEM format or by specifying its serial number. The data in the new certificate will come from the original certificate, it will not be loaded from the CSR.

```
<reissueCertificate>
  <requestHeader>
    <authToken>
      <userName>255 String
      <password>255 String
    </authToken>
```

```
  </requestHeader>
  <userAgent>255 String
  <CSR>4000 String
  <X509Cert>4000 String
  <serialNumber>32 String
  <hashAlgorithm>RSA-SHA256, ECC-SHA256
  <SANEntries>
    <SANEntry>
      <DNSName>230 String
    </SANEntry>
  </SANEntries>
  <SANApprover>
    <approverMethod>ADMIN, FILE, DNS_TXT, DNS_CNAME, DNS_TXT_PREFIX, DNS_CNAME_PREFIX
    <approverEmail>255 String
    <approverEmailPrefix>ADMIN, ADMINISTRATOR, POSTMASTER, HOSTMASTER, WEBMASTER
    <verificationNotificationEnabled>true, false
  </SANApprover>
</reissueCertificate>
```

Field name	Req.	Type	Description
userAgent	NO	255 String	Browser and operating system information.
CSR	YES	4000 String	Certification request in PKCS#10 format.
X509Cert	YES	4000 String	Reissued certificate in PEM format (Base64)
serialNumber	YES	32 String	The serial number of the certificate being reissued in HEX format.
hashAlgorithm	NO	List	RSA-SHA256, ECC-SHA256 – If not provided, the default value configured for the partner product will be used. The available hash functions can be obtained by calling the getProductList method

Data for SSL certificate – the list of domains will come from the previous certificate. As part of the reissue, new domain can be added, with all new domains requiring verification. These data are required when submitting reissue for SSL certificates with new domains.

```
  <SANEntries>
    <SANEntry>
      <DNSName>230 String
    </SANEntry>
  </SANEntries>
  <SANApprover>
    <approverMethod>ADMIN, FILE, DNS_TXT, DNS_CNAME, DNS_TXT_PREFIX, DNS_CNAME_PREFIX
    <approverEmail>255 String
    <approverEmailPrefix>ADMIN, ADMINISTRATOR, POSTMASTER, HOSTMASTER, WEBMASTER
    <verificationNotificationEnabled>true, false
  </SANApprover>
```

Field name	Req.	Type	Description
DNSName	YES	230 String	New domain to be certified.
approverMethod	YES	List	ADMIN, FILE, DNS_TXT, DNS_CNAME, DNS_TXT_PREFIX, DNS_CNAME_PREFIX – one of the available verification methods.
approverEmail	NO	255 String	Email to which the verification code will be sent – if the DNS or FILE method is selected.
approverEmailPrefix	NO	List	ADMIN, ADMINISTRATOR, POSTMASTER, HOSTMASTER, WEBMASTER – if the ADMIN method has been selected, select the prefix to which the verification emails will be sent.
verificationNotificationEnabled	NO	true/false	FALSE – Parameter that disables sending verification emails, if the DNS or FILE method is

	selected. For the ADMIN method, verification emails are always sent. No value means that the default value configured for the partner is accepted. The current configuration can be obtained by calling the getProductList method

6.9. reissueCertificateResponse

The response returns the confirmation of the order with its number. If domain verification is required, it returns additional information about that verification. The structure of the entire request is the same as in the **quickOrder**.

```
<reissueCertificateResponse>
  <responseHeader>
    <successCode>3
    <errors>
      <error>
        <errorCode>5
      </error>
    </errors>
    <timestamp>YYYY-MM-DDTHH:MM:SS.000Z
  </responseHeader>
  <orderID>50 String
  <SANVerification>
    <approverMethod>FILE, DNS_TXT, DNS_CNAME, DNS_TXT_PREFIX, DNS_CNAME_PREFIX
    <code>50 String
    <approverEmail>255 String
    <FQDNs>
      <FQDN>230 String
    </FQDNs>
  </SanVerification>
</reissueCertificateResponse>
```

Field name	Req.	Type	Description
orderID	YES	50 String	The unique identifier of the order that will be used by the partner. If not provided, it will be automatically assigned by API.
approverMethod	NO	List	FILE, DNS_TXT, DNS_CNAME, DNS_TXT_PREFIX, DNS_CNAME_PREFIX – selected verification method. The verification data is not returned if the ADMIN method is selected.
code	NO	50 String	Verification code.
approverEmail	NO	255 String	Email to which the verification code will be sent – if the DNS or FILE method is selected.
FQDN	NO	230 String	Domain for which verification is required.

6.10. renewCertificateRequest

The request should contain all the data needed to renew the certificate, such as CSR with new keys in the form of PKCS#10 and the product code. To renew the original certificate must be provided, either by attaching its file in PEM format or by specifying its serial number. The data in the new certificate will come from the renewed certificate, it will not be loaded from the CSR.

```
<renewCertificate>
  <requestHeader>
    <authToken>
      <userName>255 String
      <password>255 String
```

```
    </authToken>
  </requestHeader>
  <customer>64 String
  <userAgent>255 String
  <revocationContactEmail>255 String
  <productCode>3 String
  <CSR>4000 String
  <X509Cert>4000 String
  <serialNumber>32 String
  <hashAlgorithm> RSA-SHA256, ECC-SHA256
  <shortenedValidityPeriod>25 YYYY-MM-DD
  <SANApprover>
    <approverMethod>ADMIN, FILE, DNS_TXT, DNS_CNAME, DNS_TXT_PREFIX, DNS_CNAME_PREFIX
    <approverEmail>255 String
    <approverEmailPrefix>ADMIN, ADMINISTRATOR, POSTMASTER, HOSTMASTER, WEBMASTER
    <verificationNotificationEnabled>true, false
  </SANApprover>
</renewCertificate>
```

Field name	Req.	Type	Description
customer	YES	64 String	Customer ID or customer login for SimplySign service.
userAgent	NO	255 String	Browser and operating system information.
revocationContactEmail	NO	255 String	Email address that will be used to notify the client only in the event of certificate revocation as a result of reporting non-compliance. The given address cannot be the partner's login.
productCode	YES	3 String	3-digit product code.
CSR	YES	4000 String	Certification request in PKCS#10 format.
X509Cert	YES	4000 String	Renewed certificate in PEM format (Base64)
serialNumber	YES	32 String	The serial number of the certificate being renewed in HEX format.
hashAlgorithm	NO	List	RSA-SHA256, ECC-SHA256 – If not provided, the default value configured for the partner product will be used. The available hash functions can be obtained by calling the getProductList method
shortenedValidityPeriod	NO	YYYY-MM-DD	Certificate expiration date, must be less than the current date + the number of days resulting from the product (e.g. current date + 364 days for a 365 certificate). Set this value to expire the certificate on a specific date.

Data for SSL certificate – the list of domains will come from the renewed certificate, but all domains require re-verification. These data are required when placing an order for an SSL certificate.

```
  <SANApprover>
    <approverMethod>ADMIN, FILE, DNS_TXT, DNS_CNAME, DNS_TXT_PREFIX, DNS_CNAME_PREFIX
    <approverEmail>255 String
    <approverEmailPrefix>ADMIN, ADMINISTRATOR, POSTMASTER, HOSTMASTER, WEBMASTER
    <verificationNotificationEnabled>true, false
  </SANApprover>
```

Field name	Req.	Type	Description
approverMethod	YES	List	ADMIN, FILE, DNS_TXT, DNS_CNAME, DNS_TXT_PREFIX, DNS_CNAME_PREFIX – one of the available verification methods.

approverEmail	NO	255 String	Email to which the verification code will be sent – if the DNS or FILE method is selected.
approverEmailPrefix	NO	List	ADMIN, ADMINISTRATOR, POSTMASTER, HOSTMASTER, WEBMASTER – if the ADMIN method has been selected, select the prefix to which the verification emails will be sent.
verificationNotificationEnabled	NO	true/false	FALSE – Parameter that disables sending verification emails, if the DNS or FILE method is selected. For the ADMIN method, verification emails are always sent. No value means that the default value configured for the partner is accepted. The current configuration can be obtained by calling the getProductList method

6.11. renewCertificateResponse

The response returns the confirmation of the order with its number. If domain verification is required, it returns additional information about that verification. The structure of the entire request is the same as in the **`quickOrder`**.

```
<renewCertificateResponse>
  <responseHeader>
    <successCode>3
    <errors>
      <error>
        <errorCode>5
      </error>
    </errors>
    <timestamp>YYYY-MM-DDTHH:MM:SS.000Z
  </responseHeader>
  <orderID>50 String
  <SANVerification>
    <approverMethod>FILE, DNS_TXT, DNS_CNAME, DNS_TXT_PREFIX, DNS_CNAME_PREFIX
    <code>50 String
    <approverEmail>255 String
    <FQDNs>
      <FQDN>230 String
    </FQDNs>
  </SanVerification>
</renewCertificateResponse>
```

Field name	Req.	Type	Description
orderID	YES	50 String	The unique identifier of the order that will be used by the partner. If not provided, it will be automatically assigned by API.
approverMethod	NO	List	FILE, DNS_TXT, DNS_CNAME, DNS_TXT_PREFIX, DNS_CNAME_PREFIX – selected verification method. The verification data is not returned if the ADMIN method is selected.
code	NO	50 String	Verification code.
approverEmail	NO	255 String	Email to which the verification code will be sent – if the DNS or FILE method is selected.
FQDN	NO	230 String	Domain for which verification is required.

6.12. getOrderStateRequest

The request allows to retrieve the status and verification details for the provided order ID.

```
<getOrderState>
  <requestHeader>
    <authToken>
      <userName>255 String
      <password>255 String
    </authToken>
  </requestHeader>
  <orderID>50 String
</getOrderState>
```

Field name	Req.	Type	Description
orderID	YES	50 String	The unique identifier of the order

6.13. getOrderStateResponse

The response returns information about the status of the order and each of the verifications that are carried out for the order.

```
<getOrderStateResponse>
  <responseHeader>
    <successCode>3
    <errors>
      <error>
        <errorCode>5
      </error>
    </errors>
    <timestamp>YYYY-MM-DDTHH:MM:SS.000Z
  </responseHeader>
  <orderStatus>AWAITING, VERIFICATION, ACCEPTED, ENROLLED, REJECTED
  <lastUpdateDate>YYYY-MM-DDTHH:MM:SS.000Z
  <verifications>
    <verification>
      <type>PRODUCT, APPLICANT, ORGANIZATION, AUTHORIZATION, SYSTEM, DOMAIN, EMAIL,
EXTENDED_VALIDATION
      <state>NOT_REQUIRED, REQUIRED, FAILED, VERIFIED
      <expireDate>YYYY-MM-DDTHH:MM:SS.000Z
    </verification>
  </verifications>
</getOrderStateResponse>
```

Field name	Req.	Type	Description
orderStatus	YES	List	AWAITING – new order awaiting verification, VERIFICATION – the order is being verified, ACCEPTED – order verified, ENROLLED – certificate issued, REJECTED – an order canceled by cancelOrder or rejected by Certum
lastUpdateDate	YES	Timestamp	Date of the last update of the order.
type	YES	List	PRODUCT – verification that the product is not deprecated APPLICANT – verification of the subscriber from requestorInfo ORGANIZATION – verification of the organization's data placed in the certificate AUTHORIZATION – verification of the subscriber's authorization to represent the organization SYSTEM – CAA and blacklist verification for domains from the order

			DOMAIN – verification of the subscriber's control over domains from the order EMAIL – email address verification for non-SSL certificates EXTENDED_VALIDATION – additional verification for EV
state	YES	List	NOT_REQUIRED – this type of verification is not required, REQUIRED – this type of verification is required, VERIFIED – verification is complete, FAILED – verification failed.
expireDate	NO	Timestamp	The date by which a given verification, if completed, will remain valid.

6.14. getOrderByOrderIDRequest

The request allows to retrieve an order for the provided order ID. All other parameters are set to false by default.

```
<getOrderByOrderID>
  <requestHeader>
    <authToken>
      <userName>255 String
      <password>255 String
    </authToken>
  </requestHeader>
  <orderID> 50 String
  <orderOption>
    <orderStatus>true, false
    <orderDetails>true, false
    <certificateDetails>true, false
  </orderOption>
</getOrderByOrderID>
```

Field name	Req.	Type	Description
orderID	YES	50 String	The unique identifier of the order
orderStatus	NO	true/false	TRUE – returns basic information about the order, including the processing status.
orderDetails	NO	true/false	TRUE – returns the order details.
certificateDetails	NO	true/false	TRUE – returns details of the certificate if issued

6.15. getOrderByOrderIDResponse

The response returns the information specified in the request.

```
<getOrderByOrderIDResponse>
  <responseHeader>
    <successCode>3
    <errors>
      <error>
        <errorCode>5
      </error>
    </errors>
    <returnCount>5
    <timestamp>YYYY-MM-DDTHH:MM:SS.000Z
  </responseHeader>
  <orders>
    <Order reissue="true">
      <orderStatus>
        <orderID>50 String
        <orderStatus>AWAITING, VERIFICATION, ACCEPTED, ENROLLED, REJECTED
        <orderDate>YYYY-MM-DDTHH:MM:SS.000Z
        <productCode>3 String
        <customer>64 String
        <serialNumber>32 String
```

```
      </orderStatus>
      <orderDetails>
        <requestorInfo>
          <email>255 String
          <firstName>16 String
          <lastName>40 String
          <phone>32 String
        </requestorInfo>
        <organizationInfo>
          <taxIdentificationNumber>32 String
        </organizationInfo>
      </orderDetails>
      <certificateDetails>
        <certificateStatus>VALID, REVOKING, REVOKED
        <startDate>25 YYYY-MM-DDTHH:MM:SS.000Z
        <endDate>25 YYYY-MM-DDTHH:MM:SS.000Z
        <commonName>64 String
        <serialNumber>32 String
        <subjectName>3000 String
        <DNSNames>300 String
        <revokedDate>25 YYYY-MM-DDTHH:MM:SS.000Z
        <X509Cert>4000 String
      </certificateDetails>
    </Order>
  <orders>
</getOrderByOrderIDResponse>
```

Basic order information, if returned in response.

```
    <Order reissue="true">
      <orderStatus>
        <orderID>50 String
        <orderStatus>AWAITING, VERIFICATION, ACCEPTED, ENROLLED, REJECTED
        <orderDate>YYYY-MM-DDTHH:MM:SS.000Z
        <productCode>3 String
        <customer>64 String
        <serialNumber>32 String
      </orderStatus>
    </Order>
```

Field name	Req.	Type	Description
reissue="true"	NO	true	Indicates a reissue certificate.
orderID	YES	50 String	The unique identifier of the order .
orderStatus	YES	List	AWAITING – new order awaiting verification, VERIFICATION – the order is being verified, ACCEPTED – order verified, ENROLLED – certificate issued, REJECTED – an order canceled by cancelOrder or rejected by Certum
orderDate	YES	Timestamp	The date the order was placed.
productCode	YES	3 String	3-digit product code.
customer	YES	64 String	Customer ID.
serialNumber	NO	32 String	Certificate serial number, returned only if the certificate exists, certificate number in HEX format.

Extended order information, if returned in response.

```
<orderDetails>
  <requestorInfo>
    <email>255 String
    <firstName>16 String
    <lastName>40 String
    <phone>32 String
  </requestorInfo>
  <organizationInfo>
    <taxIdentificationNumber>32 String
  </organizationInfo>
</orderDetails>
```

Field name	Req.	Type	Description
email	YES	3 String	Subscriber's email address.
firstName	YES	16 String	Subscriber's first name.
lastName	YES	40 String	Subscriber's last name.
phone	YES	32 String	Subscriber's phone number.
taxIdentificationNumber	YES	64 String	Tax identification number or company identifier, such as DUNS.

Extended certificate information, if returned in response.

```
<certificateDetails>
  <certificateStatus>VALID, REVOKING, REVOKED
  <startDate>25 YYYY-MM-DDTHH:MM:SS.000Z
  <endDate>25 YYYY-MM-DDTHH:MM:SS.000Z
  <commonName>64 String
  <serialNumber>32 String
  <subjectName>3000 String
  <DNSNames>300 String
  <revokedDate>25 YYYY-MM-DDTHH:MM:SS.000Z
  <X509Cert>4000 String
</certificateDetails>
```

Field name	Req.	Type	Description
certificateStatus	YES	List	VALID – valid certificate, REVOKING – certificate in the process of revocation, such a status may be given to certificates pending revocation after reissue, REVOKED – certificate revoked.
startDate	YES	Timestamp	Certificate validity start date.
endDate	YES	Timestamp	Certificate expiration date.
commonName	YES	64 String	The common name may contain the subscriber's first and last name for the ID certificate, or the domain name for the SSL certificate.
serialNumber	YES	32 String	Certificate serial number in HEX format.
subjectName	YES	3000 String	The content of the Subject field.
DNSNames	NO	300 String	The content of the SAN field, returned only for SSL certificates.
revokedDate	NO	Timestamp	Revocation date, returned only if the certificate status is REVOKED.
X509Cert	YES	4000 String	Certificate in PEM format (Base64)

6.16. cancelOrderRequest

The request allows the order to be canceled if the certificate has not been issued.

```
<cancelOrder>
  <requestHeader>
    <authToken>
      <userName>255 String
      <password>255 String
    </authToken>
  </requestHeader>
  <cancelParameters>
    <orderID>50 String
    <note>255 String
  </cancelParameters>
</cancelOrder>
```

Field name	Req.	Type	Description
orderID	YES	50 String	The unique identifier of the order .
note	NO	255 String	Reason for canceling the order.

6.17. cancelOrderResponse

The response does not return any data. If a certificate has been issued for a given request, an error and the serial number of the certificate will be returned.

```
<cancelOrderResponse>
  <responseHeader>
    <successCode>3
    <errors>
      <error>
        <errorCode>5
        <value>32 String
      </error>
    </errors>
    <timestamp>YYYY-MM-DDTHH:MM:SS.000Z
  </responseHeader>
</cancelOrderResponse>
```

Field name	Req.	Type	Description
value	NO	32 String	Certificate serial number in the HEX format.

6.18. getSanVerificationStateRequest

The request allows to retrieve information about the domain verifications for the order. Domain verification only applies to SSL certificates.

```
<getSanVerificationState>
  <requestHeader>
    <authToken>
      <userName>255 String
      <password>255 String
    </authToken>
  </requestHeader>
  <orderID>50 String
</getSanVerificationState>
```

Field name	Req.	Type	Description
orderID	NO	50 String	The unique identifier of the order .

6.19. getSanVerificationStateResponse

The response returns information about the verification status of all domains for a given order, along with information about verification problems.

```
<getSanVerificationStateResponse>
  <responseHeader>
    <successCode>3
    <errors>
      <error>
        <errorCode>5
      </error>
    </errors>
    <timestamp>YYYY-MM-DDTHH:MM:SS.000Z
  </responseHeader>
  <sanVerifications>
    <sanVerification>
      <FQDN> 255 String
      <manualVerification>
        <state>REQUIRED, VERIFIED, FAILED
        <expireDate>YYYY-MM-DDTHH:MM:SS.000Z
        <info>ALREADY_VERIFIED,  LINK_EXPIRED,  OTHER_ERROR,  FILE_INVALID_CONTENT,
FILE_CONNECTION_ERROR, FILE_HTTP_ERROR, DNS_NO_RECORDS, DNS_NO_PROPER_RECORDS
      </manualVerification>
      <systemVerification>
        <method> CAA, PHISHTANK, GOOGLE_SAFE_BROWSING, TOP_SITES, REVOKED_CERTIFICATE
      </systemVerification>
    </sanVerification>
  </sanVerifications>
</getSanVerificationStateResponse>
```

Field name	Req.	Type	Description
FQDN	YES	255 String	Domain for which verification status is being checked.
state	YES	List	REQUIRED – verification is required, VERIFIED – verification is completed, FAILED – verification failed.
expireDate	NO	Timestamp	Verification expiry date.
info	NO	List	ALREADY_VERIFIED – the verification has already been completed, LINK_EXPIRED – verification link has expired, OTHER_ERROR – unknown error cause, FILE_INVALID_CONTENT – incorrect content of the verification file, FILE_CONNECTION_ERROR – the verification file could not be found or the webpage does not exist, FILE_HTTP_ERROR – could not connect to the server, DNS_NO_RECORDS – no TXT records on the DNS server, DNS_NO_PROPER_RECORDS – no relevant TXT records on the DNS server
method	NO	List	CAA – incorrect CAA record, the record should be corrected, PHISHTANK – domain on the Phishtank phishing list, GOOGLE_SAFE_BROWSING – domain on the Google phishing list, TOP_SITES – popular domain, contact Certum, REVOKED_CERTIFICATE – domain on the certificate revoked list, contact Certum

6.20. addSanVerificationRequest

The request allows to generate new verification codes for the domains in the order. Domain verification only applies to SSL certificates.

```
<addSanVerification>
  <requestHeader>
    <authToken>
      <userName>255 String
      <password>255 String
    </authToken>
  </requestHeader>
  <orderID>50 String
  <SANApprover>
    <approverMethod>ADMIN, FILE, DNS_TXT, DNS_CNAME, DNS_TXT_PREFIX, DNS_CNAME_PREFIX
    <approverEmail>255 String
    <approverEmailPrefix>ADMIN, ADMINISTRATOR, POSTMASTER, HOSTMASTER, WEBMASTER
    <verificationNotificationEnabled>true, false
  </SANApprover>
</addSanVerification>
```

Field name	Req.	Type	Description
orderID	YES	50 String	The unique identifier of the order .
approverMethod	YES	List	ADMIN, FILE, DNS_TXT, DNS_CNAME, DNS_TXT_PREFIX, DNS_CNAME_PREFIX – one of the available verification methods.
approverEmail	NO	255 String	Email to which the verification code will be sent – if the DNS or FILE method is selected.
approverEmailPrefix	NO	List	ADMIN, ADMINISTRATOR, POSTMASTER, HOSTMASTER, WEBMASTER – if the ADMIN method has been selected, select the prefix to which the verification emails will be sent.
verificationNotificationEnabled	NO	true/false	FALSE – Parameter that disables sending verification emails, if the DNS or FILE method is selected. For the ADMIN method, verification emails are always sent. No value means that the default value configured for the partner is accepted. The current configuration can be obtained by calling the getProductList method

6.21. addSanVerificationResponse

The response does not return any data if sending email has been enabled, but emails for missing verifications are sent. If mailing is disabled, a verification code is returned

```
<getSanVerificationStateResponse>
  <responseHeader>
    <successCode>3
    <errors>
      <error>
        <errorCode>5
      </error>
    </errors>
    <timestamp>YYYY-MM-DDTHH:MM:SS.000Z
  </responseHeader>
  <SANVerification>
    <approverMethod>FILE, DNS_TXT, DNS_CNAME, DNS_TXT_PREFIX, DNS_CNAME_PREFIX
    <code>50 String
    <approverEmail>255 String
    <FQDNs>
      <FQDN>230 String
    </FQDNs>
  </SanVerification>
</getSanVerificationStateResponse>
```

Field name	Req.	Type	Description
approverMethod	NO	List	FILE, DNS_TXT, DNS_CNAME, DNS_TXT_PREFIX, DNS_CNAME_PREFIX – selected verification method. The verification data is not returned if the ADMIN method is selected.
code	NO	50 String	Verification code.
approverEmail	NO	255 String	Email to which the verification code will be sent – if the DNS or FILE method is selected.
FQDN	NO	230 String	Domain for which verification is required.

6.22. performSanVerificationRequest

The request allows to initiate the verification of domains from the order. Domain verification only applies to SSL certificates.

```
<performSansVerification>
  <requestHeader>
    <authToken>
      <userName>255 String
      <password>255 String
    </authToken>
  </requestHeader>
  <code>50 String
</ performSansVerification>
```

Field name	Req.	Type	Description
code	NO	50 String	Verification code.

6.23. performSanVerificationResponse

The response does not return any data. To check the verification status, use the method **`getSanVerificationState.`**

```
<performSanVerificationResponse>
  <responseHeader>
    <successCode>3
    <errors>
      <error>
        <errorCode>5
      </error>
    </errors>
    <timestamp>YYYY-MM-DDTHH:MM:SS.000Z
  </responseHeader>
</performSanVerificationResponse>
```

6.24. addEmailVerificationRequest

The request allows to create a new verification of the E (email) field placed in the certificate for a given order. The email field verification applies to S/MIME and Document Signing in the Cloud certificates

```
<addEmailVerification>
  <requestHeader>
    <authToken>
      <userName>255 String
      <password>255 String
    </authToken>
```

```
  </requestHeader>
  <orderID>50 String
</addEmailVerification>
```

Field name	Req.	Type	Description
orderID	YES	50 String	The unique identifier of the order

6.25. addEmailVerificationResponse

The response does not return any data, but emails for missing verifications are sent.

```
<adddEmailVerificationResponse>
  <responseHeader>
    <successCode>3
    <errors>
      <error>
        <errorCode>5
      </error>
    </errors>
    <timestamp>YYYY-MM-DDTHH:MM:SS.000Z
  </responseHeader>
</adddEmailVerificationResponse>
```

6.26. getEmailVerificationRequest

The request allows to retrieve information about the verification of the E (email) field placed in the certificate for a given order. The email field verification applies to S/MIME and Document Signing in the Cloud certificates

```
<getEmailVerification>
  <requestHeader>
    <authToken>
      <userName>255 String
      <password>255 String
    </authToken>
  </requestHeader>
  <orderID>50 String
</getEmailVerification>
```

Field name	Req.	Type	Description
orderID	YES	50 String	The unique identifier of the order

6.27. getEmailVerificationResponse

The response returns information about the verification status of the email field placed in the certificate.

```
<getEmailVerificationResponse>
  <responseHeader>
    <successCode>3
    <errors>
      <error>
        <errorCode>5
      </error>
    </errors>
    <timestamp>YYYY-MM-DDTHH:MM:SS.000Z
  </responseHeader>
  <emailVerification>
    <email>64 String
    <verified>true, false
    <sendDate>YYYY-MM-DDTHH:MM:SS.000Z
```

```
    <verificationLinkValidityDate>YYYY-MM-DDTHH:MM:SS.000Z
    <verificationDate>YYYY-MM-DDTHH:MM:SS.000Z
    <verificationValidity>YYYY-MM-DDTHH:MM:SS.000Z
  </emailVerification>
</getEmailVerificationResponse>
```

Field name	Req.	Type	Description
email	YES	64 String	The email from the E (email) field that is subject to verification.
verified	YES	true/false	Verification status.
sendDate	YES	Timestamp	Date of sending the email with the link for verification.
verificationLinkValidityDate	YES	Timestamp	Verification link expiry date.
verificationDate	NO	Timestamp	Date of the verification.
verificationValidity	NO	Timestamp	Verification expiry date..

6.28. verifyOrderRequest

The request allows to add documents. Multiple documents containing many files can be added in one method call (eg Agreement scanned in several separate files). Documents may be required when placing an order for an IV, OV or EV certificate.

```
<verifyOrder>
  <requestHeader>
    <authToken>
      <userName>255 String
      <password>255 String
    </authToken>
  </requestHeader>
  <orderID>50 String
  <note>200 String
  <documents>
    <document>
      <type>APPLICANT, ORGANIZATION, AUTHORIZATION, ADDITIONAL, VERIFICATION_REPORT,
ATTESTATION_LETTER
      <description>255 String
      <files>
        <file>
          <fileName>255 String
          <content>Base64
        </file>
      </files>
    </document>
  </documents>
</verifyOrder>
```

Field name	Req.	Type	Description
orderID	YES	50 String	The unique identifier of the order .
note	YES	200 String	A note that will be added to the order.
type	YES	List	APPLICANT – confirmation of the subscriber's identity, it can be identity card, passport, permanent residence card, driving license. ORGANIZATION – confirmation of company existence, address, official representatives, if that information is available. It can be company establishment document, printout or extract from the official online registration agency or government registry. AUTHORIZATION – confirmation of the subscriber's right to apply for a certificate on behalf of organization. It can be employment certificate or authorization letter (power of attorney).

			ADDITIONAL – document needed for verification purpose, it can be invoice or another document confirming domain ownership, invoice confirming company's current address, statement, contract, order information etc. VERIFICATION_REPORT – verification report, if the agreement with Certum covers such a report. ATTESTATION_LETTER – attestation letter, if the agreement with Certum covers such a report.
description	YES	255 String	Description of the added document. The description should include the document number and the name and surname of the person to whom the document relates or the name of the organization.
fileName	YES	255 String	Name of the added file.
content	YES	Base64	File content in Base64 format

6.29. verifyOrderResponse

The response does not return any data.

```
<verifyOrderResponse>
  <responseHeader>
    <successCode>3
    <errors>
      <error>
        <errorCode>5
      </error>
    </errors>
    <timestamp>YYYY-MM-DDTHH:MM:SS.000Z
  </responseHeader>
</verifyOrderResponse>
```

6.30. getDocumentsListRequest

The request allows to retrieve list of documents for the provided order ID. Documents can be added with verifyOrder or by Certum, based on previous verifications.

```
<getDocumentsList>
  <requestHeader>
    <authToken>
      <userName>255 String
      <password>255 String
    </authToken>
  </requestHeader>
  <orderID>50 String
</getDocumentsList>
```

Field name	Req.	Type	Description
orderID	YES	50 String	The unique identifier of the order .

6.31. getDocumentsListResponse

The response returns the details of the document's status on the system, but does not return the document file.

```
<getDocumentsListResponse >
  <responseHeader>
    <successCode>3
    <errors>
      <error>
        <errorCode>5
```

```
      </error>
    </errors>
    <timestamp>YYYY-MM-DDTHH:MM:SS.000Z
  </responseHeader>
  <documentsInfo>
    <documentInfo>
      <state> NEW, ACCEPTED, REJECTED
      <type> APPLICANT, ORGANIZATION, AUTHORIZATION, ADDITIONAL, VERIFICATION_REPORT,
ATTESTATION_LETTER
      <createDate> YYYY-MM-DDTHH:MM:SS.000Z
      <expireDate> YYYY-MM-DDTHH:MM:SS.000Z
    </documentInfo>
  </documentsInfo>
</getDocumentsListResponse>
```

Field name	Req.	Type	Description
state	YES	List	NEW – new document awaiting verification, ACCEPTED – document verified, REJECTED – document rejected
type	YES	List	APPLICANT – confirmation of the subscriber's identity, it can be identity card, passport, permanent residence card, driving license. ORGANIZATION – confirmation of company existence, address, official representatives, if that information is available. It can be company establishment document, printout or extract from the official online registration agency or government registry. AUTHORIZATION – confirmation of the subscriber's right to apply for a certificate on behalf of organization. It can be employment certificate or authorization letter (power of attorney). ADDITIONAL – document needed for verification purpose, it can be invoice or another document confirming domain ownership, invoice confirming company's current address, statement, contract, order information etc. VERIFICATION_REPORT – verification report, if the agreement with Certum covers such a report. ATTESTATION_LETTER – attestation letter, if the agreement with Certum covers such a report.
createDate	YES	Timestamp	The date by which a given document was added to system.
expireDate	NES	Timestamp	The date by which a given document, if ACCEPTED, will remain valid.

6.32. getCertificateRequest

The request allows to obtain the certificate based on the order number or the serial number of the certificate.

```
<getCertificate>
  <requestHeader>
    <authToken>
      <userName>255 String
      <password>255 String
    </authToken>
  </requestHeader>
  <orderID>50 String
  <serialNumber>32 String
</getCertificate>
```

Field name	Req.	Type	Description
orderID	NO	50 String	The unique identifier of the order .
serialNumber	NO	32 String	Certificate serial number in the HEX format.

6.33. getCertificateResponse

The response returns the certificate file and caBundle, that is. all intermediate certificates (subCA) and the root certificate (rootCA).

```
<getCertificateResponse>
  <responseHeader>
    <successCode>3
    <errors>
      <error>
        <errorCode>5
      </error>
    </errors>
    <timestamp>YYYY-MM-DDTHH:MM:SS.000Z
  </responseHeader>
  <certificateDetails>
    <startDate>25 YYYY-MM-DDTHH:MM:SS.000Z
    <endDate>25 YYYY-MM-DDTHH:MM:SS.000Z
    <X509Cert>4000 String
  </certificateDetails>
  <caBundle>
    <X509Cert>4000 String
  </caBundle>
</getCertificateResponse>
```

Field name	Req.	Type	Description
startDate	YES	Timestamp	Certificate validity start date.
endDate	YES	Timestamp	Certificate expiration date.
X509Cert	YES	4000 String	Certificate in PEM format (Base64)

6.34. revokeCertificateRequest

The request allows to revoke certificate.

```
<revokeCertificate>
  <requestHeader>
    <authToken>
      <userName>255 String
      <password>255 String
    </authToken>
  </requestHeader>
  <revokeCertificateParameters>
    <serialNumber>32 String
    <revocationReason>KEYCOMPROMISE,    AFFILIATIONCHANGED,    CESSATIONOFOPERATION,
UNSPECIFIED, SUPERSEDED
    <keyCompromitationDate>YYYY-MM-DD
    <note>200 String
  </revokeCertificateParameters>
</revokeCertificate>
```

Field name	Req.	Type	Description
serialNumber	YES	32 String	Certificate serial number in the HEX format.
revocationReason	NO	List	KEYCOMPROMISE - private key has been compromised, e.g. certificate stolen or cryptographic card with certificate lost, AFFILIATIONCHANGED - subject's name, organization name or address included in the certificate has changed, employee dismissed,

			CESSATIONOFOPERATION - subscriber no longer controls all domains listed in certificate, e.g. domain permission expired, site closed, SUPERSEDED - certificate has been replaced with new certificate, e.g. test by the production or by higher level of validation, UNSPECIFIED - if other reasons do not apply.
keyCompromitationDate	NO	YYYY-MM-DD	Key compromise date in case of KEYCOMPROMISE revocation reason
note	NO	255 String	Note for the CA added to the revocation.

6.35. revokeCertificateResponse

The response does not return any data.

```
<revokeCertificateResponse>
  <responseHeader>
    <successCode>3
    <errors>
      <error>
        <errorCode>5
      </error>
    </errors>
    <timestamp>YYYY-MM-DDTHH:MM:SS.000Z
  </responseHeader>
</revokeCertificateResponse>
```

6.36. getOrdersByDateRangeRequest

The request allows to retrieve orders placed within the given date range report. All other parameters are set to false by default.

```
<getOrderByDateRange>
  <requestHeader>
    <authToken>
      <userName>255 String
      <password>255 String
    </authToken>
  </requestHeader>
  <fromDate>YYYY-MM-DD
  <toDate>YYYY-MM-DD
  <orderOption>
    <orderStatus>true, false
    <orderDetails>true, false
    <certificateDetails>true, false
  </orderOption>
  <pageNumber>1..100
</getOrderByDateRange>
```

Field name	Req.	Type	Description
fromDate	YES	YYYY-MM-DD	Date range for placed orders, search parameter.
toDate	YES	YYYY-MM-DD	Date range for placed orders, search parameter.
orderStatus	NO	true/false	TRUE – returns basic information about the order, including the processing status.
orderDetails	NO	true/false	TRUE – returns the order details.
certificateDetails	NO	true/false	TRUE – returns details of the certificate if issued
pageNumber	NO	1…100	Results page number. It takes values from 1 to 100. A maximum of 100 results are returned in one request. If the number of returned records is greater than 100, they are

	paged. If a value is missing, the first page of results is returned.

6.37. getOrdersByDateRangeResponse

The response returns the information specified in the request. If all parameters were set to false, it only returns the sum of the records matching the search criteria.

```
<getOrderByDateRangeResponse>
  <responseHeader>
    <successCode>3
    <errors>
      <error>
        <errorCode>5
      </error>
    </errors>
    <currentPage>1..100
    <pagesCount>1..100
    <returnCount>5
    <timestamp>YYYY-MM-DDTHH:MM:SS.000Z
  </responseHeader>
  <orders>
    <Order reissue="true">
      <orderStatus>
        <orderID>50 String
        <orderStatus>AWAITING, VERIFICATION, ACCEPTED, ENROLLED, REJECTED
        <orderDate>YYYY-MM-DDTHH:MM:SS.000Z
        <productCode>3 String
        <customer>64 String
        <serialNumber>32 String
      </orderStatus>
      <orderDetails>
        <requestorInfo>
          <email>255 String
          <firstName>16 String
          <lastName>40 String
          <phone>32 String
        </requestorInfo>
        <organizationInfo>
          <taxIdentificationNumber>32 String
        </organizationInfo>
      </orderDetails>
      <certificateDetails>
        <certificateStatus>VALID, REVOKING, REVOKED
        <startDate>25 YYYY-MM-DDTHH:MM:SS.000Z
        <endDate>25 YYYY-MM-DDTHH:MM:SS.000Z
        <commonName>64 String
        <serialNumber>32 String
        <subjectName>3000 String
        <DNSNames>300 String
        <revokedDate>25 YYYY-MM-DDTHH:MM:SS.000Z
        <X509Cert>4000 String
      </certificateDetails>
    </Order>
  <orders>
</getOrderByDateRangeResponse>
```

Basic order information, if returned in response.

```
      <orderStatus>
        <orderID>50 String
        <orderStatus>AWAITING, VERIFICATION, ACCEPTED, ENROLLED, REJECTED
        <orderDate>YYYY-MM-DDTHH:MM:SS.000Z
```

```
        <productCode>3 String
        <customer>64 String
        <serialNumber>32 String
      </orderStatus>
```

Field name	Req.	Type	Description
reissue="true"	NO	true	Indicates a reissue certificate.
orderID	YES	50 String	The unique identifier of the order .
orderStatus	YES	List	AWAITING – new order awaiting verification, VERIFICATION – the order is being verified, ACCEPTED – order verified, ENROLLED – certificate issued, REJECTED – an order canceled by cancelOrder or rejected by Certum
orderDate	YES	Timestamp	The date the order was placed.
productCode	YES	3 String	3-digit product code.
customer	YES	64 String	Customer ID.
serialNumber	NO	32 String	Certificate serial number, returned only if the certificate exists, number in the HEX format.

Extended order information, if returned in response.

```
      <orderDetails>
        <requestorInfo>
          <email>255 String
          <firstName>16 String
          <lastName>40 String
          <phone>32 String
        </requestorInfo>
        <organizationInfo>
          <taxIdentificationNumber>32 String
        </organizationInfo>
      </orderDetails>
```

Field name	Req.	Type	Description
email	YES	3 String	Subscriber's email address.
firstName	YES	16 String	Subscriber's first name.
lastName	YES	40 String	Subscriber's last name.
phone	NO	32 String	Subscriber's phone number.
taxIdentificationNumber	YES	64 String	Tax identification number or company identifier, such as DUNS..

Extended certificate information, if returned in response

```
      <certificateDetails>
        <certificateStatus>VALID, REVOKING, REVOKED
        <startDate>25 YYYY-MM-DDTHH:MM:SS.000Z
        <endDate>25 YYYY-MM-DDTHH:MM:SS.000Z
        <commonName>64 String
        <serialNumber>32 String
        <subjectName>3000 String
        <DNSNames>300 String
        <revokedDate>25 YYYY-MM-DDTHH:MM:SS.000Z
        <X509Cert>4000 String
      </certificateDetails>
```

Field name	Req.	Type	Description
certificateStatus	YES	List	VALID – valid certificate, REVOKING – certificate in the process of revocation, such a status may be given to certificates pending revocation after reissue, REVOKED – certificate revoked.
startDate	YES	Timestamp	Certificate validity start date.
endDate	YES	Timestamp	Certificate expiration date.
commonName	YES	64 String	The common name may contain the subscriber's first and last name for the ID certificate, or the domain name for the SSL certificate.
serialNumber	YES	32 String	Certificate serial number in HEX format.
subjectName	YES	3000 String	The content of the Subject field.
DNSNames	NO	300 String	The content of the SAN field, returned only for SSL certificates.
revokedDate	NO	Timestamp	Revocation date, returned only if the certificate status is REVOKED.
X509Cert	YES	4000 String	Certificate in PEM format (Base64)

6.38. getModifiedOrdersRequest

The request allows to retrieve modified orders for which there was a status change within the given date range report. All other parameters are set to false by default. The structure of the request is the same as for **`getOrderByDateRange`**.

```
<getModifiedOrders>
  <requestHeader>
    <authToken>
      <userName>255 String
      <password>255 String
    </authToken>
  </requestHeader>
  <fromDate>YYYY-MM-DD
  <toDate>YYYY-MM-DD
  <orderOption>
    <orderStatus>true, false
    <orderDetails>true, false
    <certificateDetails>true, false
  </orderOption>
  <pageNumber>1..100
</getModifiedOrders>
```

6.39. getModifiedOrdersResponse

The response returns the information specified in the request. If all parameters were set to false, it only returns the sum of the records matching the search criteria. The structure of the response is the same as for **`getOrderByDateRange`**.

```
<getModifiedOrdersResponse>
  <responseHeader>
    <successCode>3
    <errors>
      <error>
        <errorCode>5
      </error>
    </errors>
    <currentPage>1..100
    <pagesCount>1..100
    <returnCount>5
    <timestamp>YYYY-MM-DDTHH:MM:SS.000Z
```

```
  </responseHeader>
  <orders>
    <Order reissue="true">
      <orderStatus>
        <orderID>50 String
        <orderStatus>AWAITING, VERIFICATION, ACCEPTED, ENROLLED, REJECTED
        <orderDate>YYYY-MM-DDTHH:MM:SS.000Z
        <productCode>3 String
        <customer>64 String
        <serialNumber>32 String
      </orderStatus>
      <orderDetails>
        <requestorInfo>
          <email>255 String
          <firstName>16 String
          <lastName>40 String
          <phone>32 String
        </requestorInfo>
        <organizationInfo>
          <taxIdentificationNumber>32 String
        </organizationInfo>
      </orderDetails>
      <certificateDetails>
        <certificateStatus>VALID, REVOKING, REVOKED
        <startDate>25 YYYY-MM-DDTHH:MM:SS.000Z
        <endDate>25 YYYY-MM-DDTHH:MM:SS.000Z
        <commonName>64 String
        <serialNumber>32 String
        <subjectName>3000 String
        <DNSNames>300 String
        <revokedDate>25 YYYY-MM-DDTHH:MM:SS.000Z
        <X509Cert>4000 String
      </certificateDetails>
    </Order>
  <orders>
</getModifiedOrdersResponse>
```

6.40. getExpiringCertificatesRequest

The request allows to retrieve expiring certificates report.

```
<getExpiringCertificates>
  <requestHeader>
    <authToken>
      <userName>255 String
      <password>255 String
    </authToken>
  </requestHeader>
  <validityDaysLeft>2
  <pageNumber>1..100
</getExpiringCertificates>
```

Field name	Req.	Type	Description
validityDaysLeft	YES	2	Number of days specifying the period for which expiring certificates are to be searched for – the limit is up to 30 days.
pageNumber	NO	1…100	Results page number. It takes values from 1 to 100. A maximum of 100 results are returned in one request. If the number of returned records is greater than 100, they are paged. If a value is missing, the first page of results is returned.

6.41. getExpiringCertificatesResponse

The response returns the information specified in the request.

```
<getExpiringCertificatesResponse>
  <responseHeader>
    <successCode>3
    <errors>
      <error>
        <errorCode>5
      </error>
    </errors>
    <currentPage>1..100
    <pagesCount>1..100
    <returnCount>5
    <timestamp>YYYY-MM-DDTHH:MM:SS.000Z
  </responseHeader>
  <expiringCertificates>
    <orderID>50 String
    <serialNumber>32 String
    <expiringDate> YYYY-MM-DDTHH:MM:SS.000Z
    <validityDaysLeft>4
  <expiringCertificates>
</getExpiringCertificatesResponse>
```

Field name	Req.	Type	Description
orderID	YES	50 String	The unique identifier of the order .
serialNumber	NO	32 String	Certificate serial number, returned only if the certificate exists, certificate number in HEX format.
expiringDate	YES	Timestamp	Certificate expiry date.
validityDaysLeft	YES	4	The number of days until the certificate expires.

7. Error codes

Error code	Description
0	Request processed successfully.
1	Error occurred during processing.
3	Incorrect authentication data in requestHeader/authToken element.
1001	Public key algorithm from CSR is not supported.
1002	Element orderParameters/CSR not fund or empty.
1006	The productCode element contains the code of the discontinued product.
1007	Element SANEntries not found.
1008	Element SANEntries/SANEntry/DNSName has wrong domain name.
1009	Element requestorInfo not found.
1010	Element orderParameters not found.
1012	Order ID is already taken.
1013	Wrong product code.
1014	Element orderParameters/productCode not found.
1015	Attribute CommonName not found.
1016	Attribute Organization not found.
1017	Attribute OrganizationUnit in CSR not found.
1018	Attribute Locality in CSR not found.
1019	Attribute State in CSR not found.
1020	Attribute Country not found.
1021	Attribute EmailAddress not found.
1022	Attribute EmailAddress has errors.
1023	Value of customer element cannot be the same as requestHeader/authToken/username element.
1024	Common name attribute must be encoded as UTF8String.
1025	Attribute EmailAddress has to be encoded as IA5String.
1026	Attribute Country has to be encoded as PrintableString.
1027	Attribute State has to be encoded as UTF8String.
1028	Attribute Locality has to be encoded as UTF8String.
1029	Attribute OrganizationUnit has to be encoded as UTF8String.
1030	Attribute Organization has to be encoded as UTF8String.
1031	Common name attribute must be encoded as PrintableString.
1032	Public key from CSR has already been used.
1033	Order ID does not exist.
1037	Address email in element approverEmail has errors.
1042	Domain verification method not supported.
1043	Order ID contains characters that are not allowed: "&'<>.
1045	Domain in CommonName is not present in SANEntries.
1046	Only one domain in SANEntries element May contain asterix “*” in domain name.
1048	Element taxIdentificationNumber not found or has incorrect value.
1049	Cannot read public key from CSR.
1053	Element requestorInfo/email not found.
1054	Element requestorInfo/firstName not found.
1055	Element requestorInfo/lastName not found.
1059	The Country attribute contains an unsupported country code.
1060	Email address from requestorInfo/email element is incorrect.
1063	Public key from CSR is blacklisted.
1065	Value from requestorInfo/lastName element exceeded 40 characters.
1066	Value from requestorInfo/phone element exceeded 32 characters.
1072	Value from requestorInfo/firstName element exceeded 16 characters.
1075	Partner is not allowed to order this product.
1076	Element customer not found or is empty.

1077	Value from customer element exceeded 64 characters.
1079	Value from orderID element exceeded 50 characters.
1080	Value from productCode element exceeded 3 characters.
1081	Value from email element exceeded 255 characters.
1083	Value from taxIdentificationNumber element exceeded 32 characters.
1085	Value from SANEntries/SANEntry/DNSName element exceeded the number of characters allowed.
1087	Value from approverEmail element exceeded 255 characters.
1088	Value from orderParameters/language element exceeded 2 characters.
1092	Attribute JoISoPN not found.
1093	Attribute JoISoCN not found.
1094	Attribute JoILN not found.
1095	Attribute SerialNumber not found.
1096	Attribute JoISoCN exceeded 2 characters.
1097	Attribute JoISoPN exceeded 128 characters.
1098	Attribute JoILN exceeded 128 characters.
1099	Attribute JoISoCN contains unsupported country code.
1100	Attribute SerialNumber exceeded 64 characters.
1101	Attribute CommonName exceeded 64 characters.
1102	Attribute Organization exceeded 64 characters.
1103	Attribute OrganizationalUnit exceeded 64 characters.
1104	Attribute Locality exceeded 128 characters.
1105	Attribute State exceeded 128 characters.
1106	Attribute EmailAddress exceeded 64 characters.
1107	Number of domains in SANEntries element exceeded allowed value for product.
1108	The value in approverEmailPrefix does not match the prefix list.
1110	Date in fromDate is incorrect.
1111	Date in toDate is incorrect.
1113	Order status prevents sending email verification. The order is being processed or has been canceled.
1114	All verification messages are still valid. Verification emails hasn't been sent.
1115	SANEntries/SANEntry/DNSName Element contains Widlcard domain which is not allowed for selected product.
1116	CommonName contains Widlcard domain which is not allowed for selected product.
1117	CommonName doesn't contain Widlcard domain which is mandatory for selected product.
1118	SANEntries/SANEntry/DNSName element doesn't contain Widlcard domain which is mandatory for selected product.
1126	SerialNumber attribute has to be UTF8String encoded.
1127	JoILN attribute has to be UTF8String encoded.
1128	JoISoCN attribute has to be PrintableString encoded.
1129	JoISoPN attribute has to be UTF8String encoded.
1131	Element revokeCertificate/RevokeCertificateParameters not found.
1133	No certificate at given serial number.
1135	Incorrect value in revokeCertificate/RevokeCertificateParameters/revocationReason.
1138	Request cannot be proceded. Please contact CERTUM.
1139	It is impossible to cancel an order due to its status. Please try again later. If problem repeats please contact CERTUM.
1140	Date in keyCompromitationDate element must be in format YYYY-MM-DD
1141	Date in the keyCompromitationDate element must be between the certificate valid from date and current date.
1142	Value of revokeCertificate/RevokeCertificateParameters/note element exceeded 250 characters.
1143	Cannot revoke the certificate because certificate is during revocation process.
1144	Cannot revoke the certificate because it has expired.
1145	Cannot revoke the certificate because it has already been revoked.

1148	Cannot place an order with private IP address.
1151	Order has already been cancelled.
1153	Page number Has to be from range [1 – 100].
1154	Too many records were returned from the query.
1155	Page not fund.
1156	Customer filed in orderParameters contains disallowed characters: "&'<>.
1157	SANEntries/SANEntry/DNSName element contains gTLD.
1159	Certificate not found.
1160	Certificate serial number is missing.
1161	Serial number exceeded 64 characters.
1162	Required parameters are missing (serial number or order identifier).
1163	Request contains parameters that cannot occur simultaneously (certificate serial number and order identifier).
1164	Certificate cannot be identified properly. Provide certificate serial number OR certificate in PEM form.
1165	Certificate format is incorrect (certificate should be in PEM form).
1166	Provided product code cannot be use as a renewal code.
1167	Profile of the original certificate and renewal profile mismatch.
1168	CN mismatch.
1169	Customer mismatch.
1170	Certificate issued from different account.
1172	Request contains additional elements. Please verify request with documentation.
1176	Order status prevents the addition of new documents and domain verification.
1181	Product code is incorrect for certificate issuance.
1182	Code element not found.
1183	Value in Code element exceeded 255 characters.
1184	Verification code has expired.
1186	Verification code is incorrect.
2005	None of provided certificate contain domain names.
2049	verifyOrder/verifyOrderParameters/note not found or empty.
2050	Value from verifyOrderParameters/note element exceeded 227 characters.
2051	Incorrect Valu In CommonName.
2052	getExpiringCertificates/validityDaysLeft element not found or has incorrect value.
2053	cancelParameters element not found.
2055	orderID element not found.
2056	Order status don't allow to perform the action.
2057	Value from toDate element cannot be emalier then value from fromDate element.
2058	Attribute Locality or StateOfProvince not found.
2059	Incorrect value in verificationNotificationEnabled element.
2063	verifyOrder/verifyOrderParameters/documents/ document element not found.
2064	verifyOrder/verifyOrderParameters/documents/document/type element not found.
2065	verifyOrder/verifyOrderParameters/documents/document/description element not found.
2066	verifyOrder/verifyOrderParameters/documents/document/files element not found.
2067	verifyOrder/verifyOrderParameters/documents/document/files/file element not found.
2068	verifyOrder/verifyOrderParameters/documents/document/files/file/filename element not found.
2069	verifyOrder/verifyOrderParameters/documents/document/files/file/content element not found.
2070	verifyOrder/verifyOrderParameters/documents/document/files/file/content element should be in base64 form.
2080	Request size limit has been exceeded.
2081	fileName element cannot contain following characters: \ / : * ? " < > |
2082	Value in fileName element has incorrect length. Allowed length are from 3 to 255 characters.
2083	Incorrect type of document.

2088	Part of domain name in SANEntries/SANEntry/DNSName element exceeded the number of characters allowed.
2089	Algorithm has not been configured
2090	Incorrect hashAlgorithm value
2091	Incorrect value in getProductList/HashAlgorithm element
2092	Cannot reissue, reissued certificate. Please contact CERTUM for more information.
2093	Certificate validity date is from the past
2094	Certificate cannot be reissued. Please contact CERTUM for more information.
2095	SANEntries/SANEntry/DNSName element cannot be empty.
2096	serialNumber has to be in HEX form.
2097	CSR element is invalid.
2104	Key compromise date may be used only when revocation reason is KEYCOMPROMISE and certificate is not CodeSigning.
2109	Not allowed public key length.
2111	Element ProductCode is empty or has not been found.
2122	Reissue for products for the SimplySign service is unavailable.
2124	Unsupported hash algorithm.
2125	Incorrect signature in CSR.
2126	Certificate reissue is not possible. Cannot reissue the same certificate again.
2127	Incorrect value in certificateDetails element.
2128	Incorrect value in certificateOrder element.
2129	Incorrect value in orderStatus element.
2130	Element SANEntries/SANEntry/DNSName contains domain existing already in issued certificate.
2131	Certification request not found
2137	Incorrect value in businessCategory element.
2138	Attribute businessCategory not found.
2139	Attribute streetAddress (ST) not found.
2140	Attribute streetAddress (ST) exceeded 64 characters.
2141	Attribute streetAddress (ST) has to be UTF8String encoded.
2142	PostalCode attribute not found.
2143	Attribute postalCode (P) exceeded 40 characters.
2144	Attribute postalCode (P) has to be PrintableString.
2153	Order with given number was not requested via API.
2154	Incorrect value in postalCode (P) element.
2155	Incorrect value in Jurisdiction of Incorporation State or Province Name JoISoPN element.
2156	Incorrect value in stateOrProvinceName (SP).
2157	Element approverMethod not found.
2158	Too many elements. Only one approverEmail or approverEmailPrefix can be used.
2159	Required approverEmailPrefix element not found.
2160	Required approverEmail element not found.
2162	Element SANApprover not found.
2163	The method does not support this product.
2164	SerialNumber attribute is invalid.
2165	Joiln attribute is invalid.
2166	Locality attribute is invalid.
2167	OrganizationUnit attribute is invalid.
2168	Organization attribute is invalid.
2169	StreetAddress attribute is invalid.
2170	The value of Description element exceeded the max size of 1000 characters.
2171	Verification method not supported for IP address
2172	Value of revocationContactEmail cannot be the same as requestHeader/authToken/username element.
2173	Address email in element revocationContactEmail is invalid.
2185	Date in shortenedValidityPeriod or ValidityPeriod/NotAfter is from the past
2186	The verification method is not supported for orders with Wildcard domains.

2188	Reissue request requires re-verification of domains. Due to regulatory changes on verification methods, the existing verifications cannot be reused. Reissue cannot be issued, please place a new order.
2189	Documents limit for order has been exceeded.
2190	Revocation reason Data changed is not allowed for DV certificates.
2191	Contents of at least two attachments are the same. Cannot add the same attachment to the same order.
2192	The certificate you are trying to perform the operation for contains invalid domains. The operation cannot be performed, please place a new order.
2193	IP address is not allowed in EV SSL certificates.
2194	Attribute givenName not found.
2195	Attribute givenName has to be encoded as UTF8String.
2196	Attribute givenName exceeded 16 characters.
2197	Attribute givenName has an invalid value.
2198	Attribute surname not found.
2199	Attribute surname has to be encoded as UTF8String.
2200	Attribute surname exceeded 40 characters.
2201	Attribute surname has an invalid value.
2204	Verification of the control over the domains is already in progress. The time of verification depends of the number of the domains. Check the domains verification status.
2205	Too many attempts to perform the domain verification. Please wait a while, check the verification status and try again if necessary.

8. Change log

Date	Version	Description
2021-08-01	5.0	New version of the document. Chapters reorganization. Removal of typos. Fixes in API method descriptions. New diagrams and descriptions of the processes.
2021-08-09	5.1	Added change markings to API methods: • green – fields added to the API • red – fields that are deprecated and will be removed in the next release Added fields: • userAgent for the renewCertificate, renewCertificate methods • shortenedValidityPeriod for the quickOrder, validateOrderParameters, renewCertificate methods Described changes in the validityPeriod section – fields are deprecated and will be removed in the next versions, now its adapted to shortenedValidityPeriod. Removed unsupported RSA-SHA1 value from the hashAlgorithm list. Removed error codes 1044, 1070, 1071, 1109. Added error code 2185.
2021-09-10	5.2	Removed fields from requests marked as deprecated: • validityPeriod section of quickOrder, validateOrderParameters • verificationPhoneNumber field from quickOrder getOrderByOrderID • id field from verifyOrder • dictionary values from the businessCategory field: PRIVATE_ORGANIZATION, BUSINESS_ENTITY, NON_COMMERCIAL_ENTITY, GOVERNMENT_ENTITY Removed methods marked as deprecated: • getDomainVerification • changeApprovers • verifyDomain • sendNotifications • getApproverList • updateDocuments The error codes related to the removed methods have been removed. A new type has been added for the approverMethod dictionary, the ADMIN value will replace EMAIL. A new verification type has been added to getOrderState: AUTHORIZATION. New document types have been added to verifyOrder: APPLICANT, ORGANIZATION, AUTHORIZATION, ADDITIONAL, VERIFICATION_REPORT to replace existing list. Additional notes on changes to FILE and ADMIN verifications have been added in chapter 3.2.2. Corrected punctuation and typos.
2021-10-31	5.3	A new information e-mail Incomplete order verification has been added. Error codes description updated: 1108, 1113, 1176, 1191 Error codes added: 2186, 2187, 2188 Error codes removed: 1034, 1035, 1036, 1039, 1041, 1086, 1112, 1130, 1158, 1171, 1173, 1174, 1175, 1177, 1178, 1179, 1180, 2060, 2062, 2071, 2072, 2073, 2074, 2075, 2076, 2077, 2078, 2079, 2084, 2086, 2087, 2121, 2135, 2145, 2161
2022-02-01	5.4	Removed EMAIL value from the approverMethod dictionary, the ADMIN value must be used. Removed document types KRS, NIP, CEIDG, DUNS, OTHERREG, EMPLOY, CONTRACT, ORDER, NAMELIST, INVOICE, OTHER. In chapter 3.2.3. additional information has been added regarding the verification of the Subscriber and the organization with dedicated types of documents. Error codes description updated: 1006, 1021, 1076, 1077, 1140, 1141, 1154, 2080, 2144, 2168, 2169, 2186 Error codes removed: 1149, 1185, 1187

2022-03-17	5.5	Updated getCertificateResponse documentation error by removing fields that were not in the API response. Added getDocumentsListRequest method
2022-07-15	5.6	In chapter 3.2. the OU field was removed from Trusted SSL and Premium EV SSL specification. In chapter 3.2.2. DNS verification methond name was replaced with DNS_TXT and new method names: DNS_CNAME, DNS_TXT_PREFIX and DNS_CNAME_PREFIX were added. In chapter 4.1.4. and 4.1.5. note about the change to the OU field in the reissue and renewal process were added. New DNS_TXT, DNS_CNAME, DNS_TXT_PREFIX, DNS_CNAME_PREFIX values were added. The DNS value was marked as deprecated. Changed description in revokeCertificate method and error code 2104: for Code Signing certificated key compromise date cannot be specified. Removed values marked as deprecated: • document types: KRS, NIP, CEIDG, DUNS, OTHERREG, EMPLOY, CONTRACT, ORDER, NAMELIST, INVOICE, OTHER • verification method EMAIL
2022-11-04	5.7	In chapter 2.9 list of product names with codes added. Revocation reasons descriptions updated, removed PRIVILEGE WITHDRAWN, added SUPERSEDED. Error code description added: 2173, 2189, 2190, 2191.
2023-05-25	5.8	In chapters 2.4 and 2.5, information about the end date of E-mail ID products added. Information about Standard CodeSigning card products removed, the missing words "in the Cloud" added to the names of Standard Codesigning, EV Code Signing and Document Signing products. In chapter 3.2.1 corrected description of the customer field content for in the Cloud products and added information about this requirement to chapters 3.5 and 3.6. In chapter 3.2.2 corrected description of the ADMIN verification method. In chapter 4.1.4 added information about blocking reissue for Code Signing products issued for a physical card. Corrected description of shortenedValidityPeriod field in quickOrder and renewCertificate methods. Error code description updated: 1176, 2122 Error codes added: 1017, 1018, 1019, 1024, 1031, 2139, 2142, 2153, 2170, 2172, 2193 Error codes removed: 1089, 1090, 1147, 1187, 1191, 2043, 2044, 2048, 2061, 2085, 2098, 2187, 1089, 1090, 1147, 1187, 1191, 2043, 2044, 2048, 2061, 2085, 2098, 2187.
2023-07-13	5.9	In chapters 2 and 3 added new S/MIME products, removed E-mail ID: • 2.1 updated the list of products, updated divisions and names • 2.2 added new S/MIME products descriptions, removed E-mail ID • 2.5 added new codes for S/MIME products, removed E-mail ID • 3.2 updated tables describing the required fields for S/MIME products, removed E-mail ID • 3.4, 3.5 and 3.5 updated the product name from E-mail ID to Certum S/MIME Sponsor Replaced "Email ID (S/MIME)" with "S/MIME" throughout the documentation. Added givenName and surname fields to methods: quickOrder and validateOrderParameters. Deprecated DNS value removed from approverMethod list. In chapter 5.34 the deprecated value PRIVILEGEWITHDRAWN has been removed. In chapters 4.1.4 and 4.1.5 removed note about Ballot SC47. Error code description updated: 1059, 2056 Error codes added: 2194, 2195, 2196, 2197, 2198, 2199, 2200, 2201

2023-12-15	5.10	Chapters 3.3, 4.2, 4.5, 5.3, 6.23 and 6.24 information about Email field in Standard Code Signing in the Cloud certificates was removed. The field is no longer present in Standard Code Signing in the cloud certificates. Chapter 4.2 optional OU field is removed from Standard CodeSigning in the Cloud, EV Code Signing in the Cloud, Document Signing in the Cloud. The field is no longer present in any of Certum's certificates. Methods QuickOrder and validateOrderParameters - organizationalUnit field is removed from API request In requestorInfo section fields firsName and lastName length is adjusted to match certificate givenName and surname. In orderParameters section email length is changed to 64 String. Error code description updated: 1065, 1072, 1106
2024-05-27	5.11	Added new document type to verifyOrder: ATTESTATION_LETTER. In chapter 6.34 description for revocationReason has changed.
2024-09-02	5.12	Throughout the document, the spelling of Subscriber has been changed to subscriber. Chapter 2.1 the address to the web interface was changed. Chapter 3.2 information about changes in E-mail ID certificates was removed. Chapter 3.3.2. description of EV Code Signing in the Cloud was updated. Chapter 4.2.2 information about limitations was removed and bullet describing limitations was added to method descriptions. Chapter 5.1.4 information about reissue limitations for physical cards was removed. Chapter 4.2.2. The validation of commonName field in the order was added
2025-02-20	5.13	The previous domain verification has been replaced by the Multi-Perspective Issuance Corroboration process. This change is implemented on the Certum side and does not affect the integration process. Error codes 2204 and 2205 have been added. Chapter 4.2.2 description of CN for Standard Code Signing has been added.