



NETWORKING4ALL

API

Specification and documentation

Version 2.0.1
28 december 2017

Table of contents

1. Overview.....	5
1.1 Version notes.....	5
2. Basic principle of the API.....	6
2.1 Basic principle.....	6
2.2 Connecting to the API.....	6
2.2.1 Staging environment.....	6
2.2.2 Test and live environment.....	6
2.3 Ordering an SSL certificate.....	7
3. Web service details: relations.....	8
3.1 createRelation.....	8
3.1.1 createRelation request.....	8
3.1.2 createRelation response.....	10
3.2 updateRelation.....	10
3.2.1 updateRelation request.....	11
3.2.2 updateRelation response.....	12
3.3 deleteRelation.....	13
3.3.1 deleteRelation request.....	13
3.3.2 deleteRelation response.....	13
3.4 getRelation.....	13
3.4.1 getRelation request.....	14
3.4.2 getRelation response.....	14
3.5 getRelationContacts.....	16
3.5.1 getRelationContacts request.....	16
3.5.2 getRelationContacts response.....	16
3.6 addRelationContacts.....	18
3.6.1 addRelationContacts Request.....	18
3.6.2 addRelationContacts Response.....	19
3.7 removeRelationContacts.....	19
3.7.1 removeRelationContacts request.....	19
3.7.2 removeRelationContacts response.....	20
3.8 getRelationResellerClients.....	20
3.8.1 getRelationResellerClients request.....	20
3.8.2 getRelationResellerClients response.....	21
3.9 addRelationResellerClients.....	22
3.9.1 addRelationResellerClients request.....	23
3.9.2 addRelationResellerClients response.....	23
3.10 removeRelationResellerClients.....	23
3.10.1 removeRelationResellerClients request.....	23
3.10.2 removeRelationResellerClients response.....	24
4. Web service details: products.....	25
4.1 getProduct.....	25
4.1.1 getProduct request.....	25
4.1.2 getProduct response.....	25

4.2	getProducts.....	26
4.2.1	getProducts request.....	26
4.2.2	getProducts response.....	27
5.	Web service details: SSL certificates.....	28
5.1	checkSSLCertificateCSR.....	28
5.1.1	checkSSLCertificateCSR request.....	28
5.1.2	checkSSLCertificateCSR response.....	28
5.2	getSSLCertificateApproverList.....	29
5.2.1	getSSLCertificateApproverList request.....	29
5.2.2	getSSLCertificateApproverList response.....	29
5.3	orderSSLCertificate.....	30
5.3.1	orderSSLCertificate request.....	30
5.3.2	orderSSLCertificate response.....	31
5.4	resendEmailSSLCertificate.....	31
5.4.1	resendEmailSSLCertificate request.....	32
5.4.2	resendEmailSSLCertificate response.....	32
5.5	getSSLCertificate.....	33
5.5.1	getSSLCertificate request.....	33
5.5.2	getSSLCertificate response.....	33
5.6	getSSLCertificateList.....	35
5.6.1	getSSLCertificateList request.....	36
5.6.2	getSSLCertificateList response.....	36
5.7	replaceSSLCertificate.....	38
5.7.1	replaceSSLCertificate request.....	38
5.7.2	replaceSSLCertificate response.....	39
5.8	changeSANSSLCertificate.....	39
5.8.1	changeSANSSLCertificate request.....	40
5.8.2	changeSANSSLCertificate response.....	40
6.	Additional API information.....	41
6.1	the requestHeader.....	41
6.2	the responseHeader.....	41
6.2.1	resultCodes.....	42
6.2.2	warningCodes.....	42
6.2.3	errorCodes.....	42
6.3	Field definitions.....	44
6.4	code examples SSL certificates.....	51
6.4.1	Create company.....	51
6.4.2	Create contact.....	52
6.4.3	CSR.....	54
6.4.4	The approver email address.....	55
6.4.5	Order the SSL certificate.....	56
6.4.6	Order SSL certificate (DV) with automatic Domain Approval.....	57
6.4.7	Resend or edit the approver email address.....	58
6.4.8	Replacing an SSL certificate.....	59
6.4.9	ordering a SAN certificate.....	60

6.4.10 Retrieving the approver list for a SAN certificate.....	61
6.4.11 ordering a SAN certificate.....	62
6.4.12 Checking the status of the order and the certificate.....	63
6.4.13 Retrieving list of SSL certificates through the API.....	64
6.4.14 Change your malware scan-options.....	64
6.4.15 Request an on-demand scan.....	65
6.4.16 Add an additional SAN-name to your certificate.....	66
6.4.17 Order a GlobalSign PersonalSign certificate.....	66
6.4.18 Ordering a GlobalSign CodeSigning certificate.....	67
6.5 UTF-8 encoding and decoding.....	69
7. Product information.....	70
7.1 Status codes.....	70
7.1.1 Status codes getSSLCertificateOrder.....	70
7.1.2 Status codes getSSLCertificate and getSSLCertificateList.....	70
8. The API control panel.....	72
8.1 Overview.....	72
8.2 Detail page.....	72

1. Overview

Networking4all offers its partners a Simple Object Access Protocol (SOAP) Application Programming Interface (API) as a way to manage relations and products in real-time, and to order SSL certificates. API clients can use, among other functionalities, the following options: create new relations, manage relations, request information about relations, and order or replace SSL certificates.

This documentation contains all the necessary information required in order to integrate the Networking4all services into the partner website.

1.1 Version notes

Version 2.0.1 (28 december 2017):

– new version API documentation, updated to latest requirements and product changes.

2. Basic principle of the API

2.1 Basic principle

The partner website performs a request through the API to Networking4all, which is answered by the API with an appropriate response. This response contains the requested information, as far as such information is available and the request was performed correctly, such as a relation handle. If the command or request was incorrect, the response will contain an error code. If this is the case, the request will not have been executed.

2.2 Connecting to the API

2.2.1 Staging environment

In order to test important changes in our API, we have developed a staging environment. This gives you the option to make updates to your system, before these changes are pushed to our live environment. The connection to our API is made through a Simple Object Access Protocol (SOAP) client in WDSL mode. The URL used to connect with is:

<http://www.stagingenvironment.com/api/soap/api.wsdl>

Please note: the variable 'test=1' will be ignored on the staging environment.

2.2.2 Test and live environment

The connection to our API is made through a Simple Object Access Protocol (SOAP) client in WDSL mode. The URL used to connect with is:

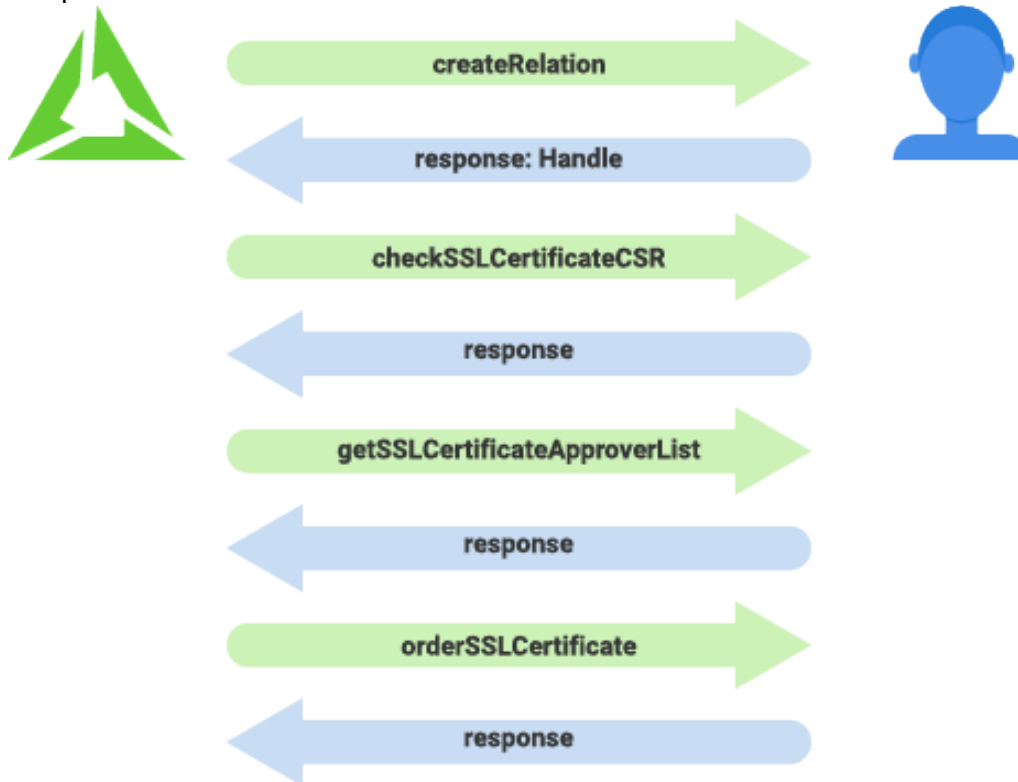
<https://www.networking4all.com/api/soap/api.wsdl>

Please note: the variable 'test=1' will transfer you to the test environment. When this variable is changed to 0, or left blank, you will be connected to the live environment.

Code examples can be found in section 6.4 of this document.

2.3 Ordering an SSL certificate

The SSL order process is as follows:



1. (optional) Create a company with the `createRelation` function. The API will respond with a handle-response.
2. (optional) Create a contact for the company with the `createRelation` function. The API will respond with a handle-response.
3. Check the CSR with the `checkSSLCertificateCSR` function. The API will either return a correct or an incorrect response.
4. Request the approver email list with the `getSSLCertificateApproverList` function. The response will contain the approver list. Please note: this step is mandatory for some SSL certificates.
5. Order the SSL certificate with the `orderSSLCertificate` function. In this step, the CSR, handle data, and if required, the approver email address, will be sent along with the request. The response will either be a success or a fail.

Sections 3 and 5 of this document will discuss the process in detail. Section 6 contains code examples of this process.

After the order process, the invoice will be sent and, if required, an approver email with the request to confirm the purchase of the SSL certificate. Once the request has been confirmed, you will receive the certificate via email.

3. Web service details: relations

The following XML notation will be used in this document:

() will be followed by either the *, ?, or + symbol, to determine the cardinality:

- * stands for 0 or more
- ? stands for 0 or 1
- + stands for 1 or more

The notation <!-- --> is used for comments within the code.

3.1 createRelation

The createRelation function is used to create company and contact relations. It is possible to combine these two purposes into one and create a company with a contact in one go. This is done by adding contact information to the creation process.

You can choose to create your own username and password during the contact creation process. If you choose not to do so, a username and password will be automatically created for you. Please note: in order to create a username and password of your own choosing, the option 'createAccount option =1' will have to be activated, or the information you enter will be ignored.

In our system, a contact is always connected to a company. This makes finding contacts in our portal much easier. When creating a contact through the API, you must enter a company handle. If this does not exist, it will have to be created first. The handle returned to you in the response can then be used in the contact creation process and will automatically connect the contact to the company. If you use a system which connects the company and the contact, you can leave out the handle. The contact will not be connected to a company handle, but to your reseller account.

Please note: It is imperative that you use the handles that were created for your company, instead of generating new handles for the same company or contact. Creating multiple handles for the same company or contact can create technical difficulties.

3.1.1 createRelation request

```
<createRelation
xmlns="https://www.networking4all.com/api/soap/api.wSDL">
  <requestHeader>
    <auth>
      <handle>      <!-- Login handle -->
      <username>
      <password>
    </auth>
    (<language>)?
    (<reference>)?
    (<test>)?
  </requestHeader>
</request>
```



```

<type> <!-- contact or company -->
(<handle>)? <!-- To add a contact -->
(<contact>
  <title>
  <initials>
  <firstName>
  (<middleName>)?
  <lastName>
  (<gender>)?
  (<birthDate>)?
  (<passportNumber>)?
</contact>)? <!-- To create a
contact -->
  (<company>
    <name>
    (<employees>)?
    (<chamberOfCommerce>)?
  </company>)? <!-- To create a
company -->
  (<address>
    <type> <!-- visitor or postal
address type -->
    <street> <!-- DEPRECATED,
gebruik street1 -->
    <streetNumber> <!-- DEPRECATED,
gebruik street1 -->
    (<streetNumberExtension>)? <!--
DEPRECATED, use street1 -->
    (<buildingName>)?
    <street1>
    (<street2>)?
    (<street3>)?
    <postalCode>
    <city>
    <province>
    <countryCode><!-- ISO 3166-1 default
country codes -->
    </address>)+
    <phone> <!-- DEPRECATED, use
phoneNumber -->
    <countryCode>
    <trunkCode>
    <subscriberNumber>
    (<extension>)?
    </phone>
    (<fax> <!-- DEPRECATED, use
faxNumber -->
    <countryCode>
    <trunkCode>
    <subscriberNumber>
    (<extension>)?
    </fax>)?
    (<mobile> <!-- DEPRECATED, use
mobileNumber -->
    <countryCode>
    <trunkCode>
    <subscriberNumber>
    (<extension>)?
    </mobile>)?
    <phoneNumber>
    (<faxNumber>)?
    (<mobileNumber>)?
    <email>
    (<website>)?

```

```

        (<account>
            <username>
            <password>
            <passwordConfirm>
        </account>)?
        (<financial>
            (<accountNumber>)?
            (<accountType>)?
            (<bicCode>)?
            (<ibanNumber>)?
            (<vatIn>)?
        </financial>)?
        (<option>
            <createAccount>
        </option>)?
    </request>
</createRelation>

```

3.1.2 createRelation response

```

<createRelationResponse
xmlns="https://www.networking4all.com/api/soap/api.wsd1">
    <responseHeader>
        <resultCode>
        (<resultMessage>)?
        (<warnings>
            (<warning>
                <code>
                <message>
                <field>
            </warning>)+
        </warnings>)?
        (<errors>
            (<error>
                <code>
                <message>
                <field>
            </error>)+
        </errors>)?
        <timestamp>
        (<language>)?
        (<reference>)?
        (<test>)?
    </responseHeader>
    (<response>
        (<handle>)+
        (<account>
            <username>
            <password>
        </account>)?
    </response>)?    <!-- response will only be given
when resultCode = 0 -->
</createRelationResponse>

```

3.2 updateRelation

The updateRelation function allows you to edit company- and contact information. This function also allows you to edit the username and password, provided the account information option is specified. This is shown in the code in the section <account>.

3.2.1 updateRelation request

```

<updateRelation
xmlns="https://www.networking4all.com/api/soap/api.wsdl">
  <requestHeader>
    <auth>
      <handle>      <!-- Login handle -->
      <username>
      <password>
    </auth>
    (<language>)?
    (<reference>)?
    (<test>)?
  </requestHeader>
  <request>
    (<type>)?      <!-- contact or company -->
    <handle>      <!-- Handle in case of
update -->
    (<contact>
      <title>
      <initials>
      <firstName>
      (<middleName>)?
      <lastName>
      (<gender>)?
      (<birthDate>)?
      (<passportNumber>)?
    </contact>)?  <!-- To update a contact -->
    (<company>
      <name>
      (<employees>)?
      (<chamberOfCommerce>)?
    </company>)?  <!-- To update a company -->
    (<address>
      <type>      <!-- visitor or postal
address type -->
      <street>    <!-- DEPRECATED, use
street1 -->
      <streetNumber>  <!-- DEPRECATED, use
street1 -->
      (<streetNumberExtension>)? <!--
DEPRECATED, use street1 -->
      (<buildingName>)?
      <street1>
      (<street2>)?
      (<street3>)?
      <postalCode>
      <city>
      <province>
      <countryCode><!-- ISO 3166-1 default
country codes -->
    </address>)+
    <phone>      <!-- DEPRECATED, use
phoneNumber -->
      <countryCode>
      <trunkCode>
      <subscriberNumber>
      (<extension>)?
    </phone>
    (<fax>      <!-- DEPRECATED, use
faxNumber -->
      <countryCode>
      <trunkCode>
      <subscriberNumber>
      (<extension>)?
    </fax>)?
  </request>
</updateRelation>

```

```

mobileNumber (<mobile>                                <!-- DEPRECATED, use
-->
    <countryCode>
    <trunkCode>
    <subscriberNumber>
    (<extension>)?
</mobile>)?
<email>
(<website>)?
(<account>
    <username>
    <password>
    <passwordConfirm>
</account>)?
(<financial>
    (<accountNumber>)?
    (<accountType>)?
    (<bicCode>)?
    (<ibanNumber>)?
    (<vatin>)?
</financial>)?
(<option>
    <createAccount>
</option>)?
</request>
</updateRelation>

```

3.2.2 updateRelation response

```

<updateRelationResponse
xmlns="https://www.networking4all.com/api/soap/api.wsdl">
    <responseHeader>
        <resultCode>
        (<resultMessage>)?
        (<warnings>
            (<warning>
                <code>
                <message>
                <field>
            </warning>)+
        </warnings>)?
        (<errors>
            (<error>
                <code>
                <message>
                <field>
            </error>)+
        </errors>)?
        <timestamp>
        (<language>)?
        (<reference>)?
        (<test>)?
    </responseHeader>
    (<response>
        <handle>
        (<account>
            <username>
            <password>
        </account>)?
    </response>)? <!-- response will only be given
when resultCode = 0 -->
</updateRelationResponse>

```

3.3 deleteRelation

The deleteRelation function allows you to delete a relation.

3.3.1 deleteRelation request

```
<deleteRelation
xmlns="https://www.networking4all.com/api/soap/api.wsdl">
  <requestHeader>
    <auth>
      <handle>      <!-- Login handle -->
      <username>
      <password>
    </auth>
    (<language>)?
    (<reference>)?
    (<test>)?
  </requestHeader>
  <request>
    <handle>      <!-- Handle for removal -->
  </request>
</deleteRelation>
```

3.3.2 deleteRelation response

```
<deleteRelationResponse
xmlns="https://www.networking4all.com/api/soap/api.wsdl">
  <responseHeader>
    <resultCode>
    (<resultMessage>)?
    (<warnings>
      (<warning>
        <code>
        <message>
        <field>
      </warning>)+
    </warnings>)?
    (<errors>
      (<error>
        <code>
        <message>
        <field>
      </error>)+
    </errors>)?
    <timestamp>
    (<language>)?
    (<reference>)?
    (<test>)?
  </responseHeader>
  (<response>
    <handle>
  </response>)?<!-- response will only be given when
resultCode = 0 -->
</deleteRelationResponse>
```

3.4 getRelation

The getRelation function retrieves the company- and contact information of a relation.

3.4.1 getRelation request

```

<getRelation
xmlns="https://www.networking4all.com/api/soap/api.wsdl">
  <requestHeader>
    <auth>
      <handle>      <!-- Login handle -->
      <username>
      <password>
    </auth>
    (<language>)?
    (<reference>)?
    (<test>)?
  </requestHeader>
  <request>
    (<handle>)?      <!-- relation handle -->
    (<name>)?
    (<type>)?
    (<firstname>)?
    (<lastname>)?
  </request>
</getRelation>

```

3.4.2 getRelation response

```

<getRelationResponse
xmlns="https://www.networking4all.com/api/soap/api.wsdl">
  <responseHeader>
    <resultCode>
    (<resultMessage>)?
    (<warnings>
      (<warning>
        <code>
        <message>
        <field>
      </warning>)+
    </warnings>)?
    (<errors>
      (<error>
        <code>
        <message>
        <field>
      </error>)+
    </errors>)?
    <timestamp>
    (<language>)?
    (<reference>)?
    (<test>)?
  </responseHeader>
  (<response>
    <type> <!-- contact or company -->
    <handle>
    (<clientNumber>) <!-- Only in case of company
-->
    (<reseller>)? <!-- Only in case of reseller
client -->
    (<resellerHandle>)? <!-- Only in case of
reseller client -->
    (<companyHandle>)+ <!-- Only in case of company
-->
    (<contact>
      <title>

```

```

        <initials>
        <firstName>
        (<middleName>)?
        <lastName>
        (<gender>)?
        (<birthDate>)?
        (<passportNumber>)?
    </contact>)?
    (<company>
        <name>
        (<employees>)?
        (<chamberOfCommerce>)?
    </company>)?
    (<address>
        <type>                                <!-- visitors or
postal address type -->
        <street>                                <!-- DEPRECATED, use
street1 -->
        <streetNumber>                          <!-- DEPRECATED, use
street1 -->
        (<streetNumberExtension>)? <!--
DEPRECATED, use street1 -->
        (<buildingName>)?
        <street1>
        (<street2>)?
        (<street3>)?
        <postalCode>
        <city>
        <province>
        <countryCode><!-- ISO 3166-1 default
country codes -->
    </address>)+
    <phone>                                <!-- DEPRECATED, use
phoneNumber -->
        <countryCode>
        <trunkCode>
        <subscriberNumber>
        (<extension>)?
    </phone>
    (<fax>                                <!-- DEPRECATED, use
faxNumber -->
        <countryCode>
        <trunkCode>
        <subscriberNumber>
        (<extension>)?
    </fax>)?
    (<mobile>                                <!-- DEPRECATED, use
mobileNumber -->
        <countryCode>
        <trunkCode>
        <subscriberNumber>
        (<extension>)?
    </mobile>)?
    <phoneNumber>
    (<faxNumber>)?
    (<mobileNumber>)?
    <email>
    (<website>)?
    (<account>
        <username>
    </account>)?
    (<financial>
        <accountNumber>
        <accountType>
        (<bicCode>)?
        (<ibanNumber>)?

```

```

        (<vatin>)?
    </financial>)?
    <audit>
        <created>
            (<createdHandle>)?
            (<updated>)?
            (<updatedHandle>?
        </audit>
        <status>?
    </response>)?      <!-- response will only be given
when resultCode = 0 -->
</getRelationResponse>

```

3.5 getRelationContacts

The getRelationContacts function retrieves a list of contacts for the company.

3.5.1 getRelationContacts request

```

<getRelationContacts
xmlns="https://www.networking4all.com/api/soap/api.wsdl">
    <requestHeader>
        <auth>
            <handle>      <!-- Login handle -->
            <username>
            <password>
        </auth>
        (<language>)?
        (<reference>)?
        (<test>)?
    </requestHeader>
    <request>
        <handle>          <!-- relation handle -->
    </request>
</getRelationContacts>

```

3.5.2 getRelationContacts response

```

<getRelationContactsResponse
xmlns="https://www.networking4all.com/api/soap/api.wsdl">
    <responseHeader>
        <resultCode>
        (<resultMessage>)?
        (<warnings>
            (<warning>
                <code>
                <message>
                <field>
            </warning>)+
        </warnings>)?
        (<errors>
            (<error>
                <code>
                <message>
                <field>
            </error>)+
        </errors>)?
        <timestamp>
        (<language>)?
        (<reference>)?

```



```

        (<test>)?
    </responseHeader>
    (<response>
        (<item>
            <type><!-- contact or company -->
            (<handle>)?
                <!-- Used to add contact -->
            (<clientNumber>)
                <!-- Only in case of company -->
            (<reseller>)?
                <!-- Only in case of reseller -->
            (<resellerHandle>)?
                <!-- Only in case of reseller -->
            (<companyHandle>)+
                <!-- Only in case of company -->
            (<contact>
                <title>
                <initials>
                <firstName>
                (<middleName>)?
                <lastName>
                (<gender>)?
                (<birthDate>)?
                (<passportNumber>)?
            </contact>)?
                <!-- Required for contact creation
-->
                (<company>
                    <name>
                    (<employees>)?
                    (<chamberOfCommerce>)?
                </company>)?
                (<address>
                    <type><!-- visitors of postal
address type -->
                    <street> <!-- DEPRECATED, use
street1 -->
                    <streetNumber> <!-- DEPRECATED,
use street1 -->
                    (<streetNumberExtension>)? <!--
DEPRECATED, use street1 -->
                    (<buildingName>)?
                    <street1>
                    (<street2>)?
                    (<street3>)?
                    <postalCode>
                    <city>
                    <province>
                    <countryCode><!-- ISO 3166-1
default country codes -->
                </address>)+
                <phone> <!-- DEPRECATED, use
phoneNumber -->
                    <countryCode>
                    <trunkCode>
                    <subscriberNumber>
                    (<extension>)?
                </phone>
                (<fax> <!-- DEPRECATED, use
faxNumber -->
                    <countryCode>
                    <trunkCode>
                    <subscriberNumber>
                    (<extension>)?
                </fax>)?

```

```

mobileNumber -->
    (<mobile>
        <!-- DEPRECATED, use
        <countryCode>
        <trunkCode>
        <subscriberNumber>
        (<extension>)?
    </mobile>)?
    <phoneNumber>
    (<faxNumber>)?
    (<mobileNumber>)?
    <email>
    (<website>)?
    (<account>
        <username>
    </account>)?
    (<financial>
        <accountNumber>
        <accountType>
        (<bicCode>)?
        (<ibanNumber>)?
        (<vatIn>)?
    </financial>)?
    <audit>
        <created>
        (<createdHandle>)?
        (<updated>)?
        (<updatedHandle>?
    </audit>
    <status>?
    </item>)+
    </response>)? <!-- response will only be given
when resultCode = 0 -->
</getRelationContactsResponse>

```

3.6 addRelationContacts

The addRelationContacts function allows you to add contacts to a company.

3.6.1 addRelationContacts Request

```

<addRelationContacts
xmlns="https://www.networking4all.com/api/soap/api.wsdl">
    <requestHeader>
        <auth>
            <handle> <!-- Login handle -->
            <username>
            <password>
        </auth>
        (<language>)?
        (<reference>)?
        (<test>)?
    </requestHeader>
    <request>
        <handle> <!-- relation handle -->
        (<contacts>)+ <!-- Contact handles to be
added -->
    </request>
</addRelationContacts>

```

3.6.2 addRelationContacts Response

```

<addRelationContactsResponse
xmlns="https://www.networking4all.com/api/soap/api.wsdl">
  <responseHeader>
    <resultCode>
      (<resultMessage>)?
    <warnings>
      (<warning>
        <code>
          <message>
            <field>
          </warning>)+
    </warnings>)?
    <errors>
      (<error>
        <code>
          <message>
            <field>
          </error>)+
    </errors>)?
    <timestamp>
      (<language>)?
      (<reference>)?
      (<test>)?
    </responseHeader>
    <response>
      <handle>
      <result>      <!-- Total number of added
contacts -->
    </response>)?      <!-- response will only be given
when resultCode = 0 -->
  </addRelationContactsResponse>

```

3.7 removeRelationContacts

The removeRelationContacts function allows you to remove contacts from a company.

3.7.1 removeRelationContacts request

```

<removeRelationContacts
xmlns="https://www.networking4all.com/api/soap/api.wsdl">
  <requestHeader>
    <auth>
      <handle>      <!-- Login handle -->
      <username>
      <password>
    </auth>
    (<language>)?
    (<reference>)?
    (<test>)?
  </requestHeader>
  <request>
    <handle>      <!-- relation handle -->
    (<contacts>)+      <!-- Contact handles to be
removed -->
  </request>
</removeRelationContacts>

```

3.7.2 removeRelationContacts response

```

<removeRelationContactsResponse
xmlns="https://www.networking4all.com/api/soap/api.wsdl">
  <responseHeader>
    <resultCode>
      (<resultMessage>)?
    <warnings>
      (<warning>
        <code>
          <message>
            <field>
          </warning>)+
    </warnings>)?
    <errors>
      (<error>
        <code>
          <message>
            <field>
          </error>)+
    </errors>)?
    <timestamp>
      (<language>)?
      (<reference>)?
      (<test>)?
    </responseHeader>
    <response>
      <handle>
      <result>      <!-- Total number of removed
contacts -->
      </response>)?      <!-- response will only be given
when resultCode = 0 -->
</removeRelationContactsResponse>

```

3.8 getRelationResellerClients

The getRelationResellerClients function allows you to retrieve a list of resellers belonging to a company.

3.8.1 getRelationResellerClients request

```

<getRelationResellerClients
xmlns="https://www.networking4all.com/api/soap/api.wsdl">
  <requestHeader>
    <auth>
      <handle>      <!-- Login handle -->
      <username>
      <password>
    </auth>
    (<language>)?
    (<reference>)?
    (<test>)?
  </requestHeader>
  <request>
    <handle>      <!-- relation handle -->
  </request>
</getRelationResellerClients>

```

3.8.2 getRelationResellerClients response

```

<getRelationResellerClients Response
xmlns="https://www.networking4all.com/api/soap/api.wsdl">
  <responseHeader>
    <resultCode>
      (<resultMessage>)?
    <warnings>
      (<warning>
        <code>
          <message>
            <field>
          </warning>)+
    </warnings>)?
    <errors>
      (<error>
        <code>
          <message>
            <field>
          </error>)+
    </errors>)?
    <timestamp>
      (<language>)?
      (<reference>)?
      (<test>)?
  </responseHeader>
  (<response>
    (<item>
      <type><!-- contact or company -->
      (<handle>)? <!-- To add a contact
-->
      (<clientNumber>) <!-- Only in case of
company -->
      (<reseller>)?
        <!-- Only in case of reseller -->
      (<resellerHandle>)?
        <!-- Only in case of reseller -->
      (<companyHandle>)+
        <!-- Only in case of company -->
      (<contact>
        <title>
          <initials>
            <firstName>
              (<middleName>)?
            <lastName>
              (<gender>)?
              (<birthDate>)?
              (<passportNumber>)?
            </contact>)? <!-- Required for contact
creation -->
      (<company>
        <name>
          (<employees>)?
          (<chamberOfCommerce>)?
        </company>)?
      (<address>
        <type><!-- visitors or postal
address type -->
        <street> <!-- DEPRECATED, use
street1 -->
        <streetNumber> <!-- DEPRECATED,
use street1 -->
          (<streetNumberExtension>)? <!--
DEPRECATED, use street1 -->

```

```

                (<buildingName>)?
                <street1>
                (<street2>)?
                (<street3>)?
                <postalCode>
                <city>
                <province>
                <countryCode><!-- ISO 3166-1
default country codes -->
            </address>)+
phoneNumber -->
            <phone>                <!-- DEPRECATED, use
                <countryCode>
                <trunkCode>
                <subscriberNumber>
                (<extension>)?
            </phone>
faxNumber -->
            <fax>                <!-- DEPRECATED, use
                <countryCode>
                <trunkCode>
                <subscriberNumber>
                (<extension>)?
            </fax>)?
mobileNumber -->
            <mobile>                <!-- DEPRECATED, use
                <countryCode>
                <trunkCode>
                <subscriberNumber>
                (<extension>)?
            </mobile>)?
            <phoneNumber>
            (<faxNumber>)?
            (<mobileNumber>)?
            <email>
            (<website>)?
            <account>
                <username>
            </account>)?
            (<financial>
                <accountNumber>
                <accountType>
                (<bicCode>)?
                (<ibanNumber>)?
                (<vatIn>)?
            </financial>)?
            <audit>
                <created>
                (<createdHandle>)?
                (<updated>)?
                (<updatedHandle>)?
            </audit>
            <status>?
        </item>)+
    </response>)?                <!-- response will only be given
when resultCode = 0 -->
</getRelationResellerClients>

```

3.9 addRelationResellerClients

The addRelationResellerClients function allows you to add resellers to a company.

3.9.1 addRelationResellerClients request

```

<addRelationContacts
xmlns="https://www.networking4all.com/api/soap/api.wsdl">
  <requestHeader>
    <auth>
      <handle>      <!-- Login handle -->
      <username>
      <password>
    </auth>
    (<language>)?
    (<reference>)?
    (<test>)?
  </requestHeader>
  <request>
    <handle>      <!-- relation handle -->
    (<clients>)+ <!-- Reseller client handles to be
added -->
  </request>
</addRelationContacts>

```

3.9.2 addRelationResellerClients response

```

<addRelationContactsResponse
xmlns="https://www.networking4all.com/api/soap/api.wsdl">
  <responseHeader>
    <resultCode>
    (<resultMessage>)?
    (<warnings>
      (<warning>
        <code>
        <message>
        <field>
      </warning>)+
    </warnings>)?
    (<errors>
      (<error>
        <code>
        <message>
        <field>
      </error>)+
    </errors>)?
    <timestamp>
    (<language>)?
    (<reference>)?
    (<test>)?
  </responseHeader>
  (<response>
    <handle>
    <result>      <!-- Total number of added
reseller clients -->
  </response>)?      <!-- response will only be given
when resultCode = 0 -->
</addRelationContactsResponse>

```

3.10 removeRelationResellerClients

3.10.1 removeRelationResellerClients request

```

<removeRelationResellerClients
xmlns="https://www.networking4all.com/api/soap/api.wsdl">
  <requestHeader>
    <auth>
      <handle>      <!-- Login handle -->
      <username>
      <password>
    </auth>
    (<language>)?
    (<reference>)?
    (<test>)?
  </requestHeader>
  <request>
    <handle>      <!-- relation handle -->
    (<clients>)+ <!-- Reseller client handles to be
added -->
  </request>
</removeRelationResellerClients>

```

3.10.2 removeRelationResellerClients response

```

<removeRelationResellerClientsResponse
xmlns="https://www.networking4all.com/api/soap/api.wsdl">
  <responseHeader>
    <resultCode>
    (<resultMessage>)?
    (<warnings>
      (<warning>
        <code>
        <message>
        <field>
      </warning>)+
    </warnings>)?
    (<errors>
      (<error>
        <code>
        <message>
        <field>
      </error>)+
    </errors>)?
    <timestamp>
    (<language>)?
    (<reference>)?
    (<test>)?
  </responseHeader>
  (<response>
    <handle>
    <result>      <!-- Total number of removed
reseller clients -->
  </response>)?      <!-- response will only be given
when resultCode = 0 -->
</removeRelationResellerClientsResponse>

```


4. Web service details: products

4.1 getProduct

The getProduct function allows you to request information about a product by means of the product code.

4.1.1 getProduct request

```
<getProduct
xmlns="https://www.networking4all.com/api/soap/api.wsdl">
  <requestHeader>
    <auth>
      <handle>      <!-- Login handle -->
      <username>
      <password>
    </auth>
    (<language>)?
    (<reference>)?
    (<test>)?
  </requestHeader>
  <request>
    <productCode>  <!-- ProductCode -->
  </request>
</getProduct>
```

4.1.2 getProduct response

```
<getProductResponse
xmlns="https://www.networking4all.com/api/soap/api.wsdl">
  <responseHeader>
    <resultCode>
    (<resultMessage>)?
    (<warnings>
      (<warning>
        <code>
        <message>
        <field>
      </warning>)+
    </warnings>)?
    (<errors>
      (<error>
        <code>
        <message>
        <field>
      </error>)+
    </errors>)?
    <timestamp>
    (<language>)?
    (<reference>)?
    (<test>)?
  </responseHeader>
  (<response>
    <productCode>
    <name>
```

```

        <period>
        <description>
            <short>
            (<long>)?
        </description>
        <fees>
            <setup>          <!-- DEPRECATED, see fee
structure -->
                <retail>
                <actual>
            </setup>
            (<fee>
                <period>
                <retail>
                <actual>
                <type>
                <currency>
            </fee>)+
        </fees>
        (<vat>)?
        (<details>
            <mailValidation>
            <dv>
            <ov>
            <ev>
            <webserver>
            <company>
            <codesigning>
            <san>
            <wildcard>
            <intranet>
            <mobile>
            <competitiveUpgrade>
        (<details>)?
        </response>)?<!-- response will only be given when
resultCode = 0 -->
</getProductResponse>

```

4.2 getProducts

The `getProducts` function allows you to retrieve a list of product codes and product names. This list does not contain additional information such as prices. For this, you need to use the product specific function `getProduct`. See section 4.1 for more information.

4.2.1 getProducts request

```

<getProducts
xmlns="https://www.networking4all.com/api/soap/api.wsdl">
  <requestHeader>
    <auth>
      <handle>          <!-- Login handle -->
      <username>
      <password>
    </auth>
    (<language>)?
    (<reference>)?
    (<test>)?
  </requestHeader>
  <request>

```

```

        <type>          <!-- type : ssl, domains, hosting
→
        (<brand>)?    <!-- For SSL only -->
        (<sslType>)? <!-- For SSL only, DV, OV, EV,
CLIENT -->
        </request>
</getProducts>

```

4.2.2 getProducts response

```

<getProductsResponse
xmlns="https://www.networking4all.com/api/soap/api.wsdl">
  <responseHeader>
    <resultCode>
      (<resultMessage>)?
      (<warnings>
        (<warning>
          <code>
            <message>
              <field>
            </warning>)+
      </warnings>)?
      (<errors>
        (<error>
          <code>
            <message>
              <field>
            </error>)+
      </errors>)?
    <timestamp>
    (<language>)?
    (<reference>)?
    (<test>)?
  </responseHeader>
  (<response>
    <products>
      (<product>
        <code>
          <name>
            (<brand>)? <!-- For SSL only -->
          </product>)+
    </products>
  </response>)?<!-- response will only be given when
resultCode = 0 -->
</getProductsResponse>

```

5. Web service details: SSL certificates

5.1 checkSSLCertificateCSR

The checkSSLCertificateCSR function checks the CSR for correctness and validity.

5.1.1 checkSSLCertificateCSR request

```
<checkSSLCertificateCSR
xmlns="https://www.networking4all.com/api/soap/api.wsdl">
  <requestHeader>
    <auth>
      <handle>      <!-- Login handle -->
      <username>
      <password>
    </auth>
    (<language>)?
    (<reference>)?
    (<test>)?
  </requestHeader>
  <request>
    <productCode>  <!-- ProductCode -->
    <csr>          <!-- CSR -->
    (<san>)*
  </request>
</checkSSLCertificateCSR>
```

5.1.2 checkSSLCertificateCSR response

```
<checkSSLCertificateCSRResponse
xmlns="https://www.networking4all.com/api/soap/api.wsdl">
  <responseHeader>
    <resultCode>
    (<resultMessage>)?
    (<warnings>
      (<warning>
        <code>
        <message>
        <field>
      </warning>)+
    </warnings>)?
    (<errors>
      (<error>
        <code>
        <message>
        <field>
      </error>)+
    </errors>)?
    <timestamp>
    (<language>)?
    (<reference>)?
    (<test>)?
  </responseHeader>
  (<response>
    <organisation>
    <city>
```

```

        <province>
        <country>
        <email>
        <cn>
        (<csr>)?
        (<san>)*
    </response>)?<!-- response will only be given when
    resultCode = 0 -->
</checkSSLCertificateCSRResponse>

```

5.2 getSSLCertificateApproverList

The getSSLCertificateApproverList function retrieves a list of approver email addresses required to order Domain Validated (DV) SSL certificates.

5.2.1 getSSLCertificateApproverList request

```

<getSSLCertificateApproverList
xmlns="https://www.networking4all.com/api/soap/api.wsdl">
  <requestHeader>
    <auth>
      <handle>      <!-- Login handle -->
      <username>
      <password>
    </auth>
    (<language>)?
    (<reference>)?
    (<test>)?
  </requestHeader>
  <request>
    <productCode>  <!-- ProductCode -->
    <domain>      <!-- Common Name or domain
name -->
  </request>
</getSSLCertificateApproverList>

```

5.2.2 getSSLCertificateApproverList response

```

<getSSLCertificateApproverListResponse
xmlns="https://www.networking4all.com/api/soap/api.wsdl">
  <responseHeader>
    <resultCode>
    (<resultMessage>)?
    (<warnings>
      (<warning>
        <code>
        <message>
        <field>
      </warning>)+
    </warnings>)?
    (<errors>
      (<error>
        <code>
        <message>
        <field>
      </error>)+
    </errors>)?
    <timestamp>
    (<language>)?

```

```

        (<reference>)?
        (<test>)?
    </responseHeader>
    (<response>
        (<approvers>)+
        (<orderNumber>)?      <!-- Only for GlobalSign DV
products -->
    </response>)?<!-- response will only be given when
resultCode = 0 -->
</getSSLCertificateApproverListResponse>

```

5.3 orderSSLCertificate

The orderSSLCertificate function allows you to order an SSL certificate.

5.3.1 orderSSLCertificate request

```

<orderSSLCertificate
xmlns="https://www.networking4all.com/api/soap/api.wsd1">
    <requestHeader>
        <auth>
            <handle>      <!-- Login handle -->
            <username>
            <password>
        </auth>
        (<language>)?
        (<reference>)?
        (<test>)?
    </requestHeader>
    <request>
        <companyHandle>
        <adminHandle>
        <techHandle>
        <productCode>
        (<orderKind>)? <!-- new or renewal, default is
new -->
        (<traceNumber>)? <!-- Only for renewal -->
        <period>      <!-- Period in years -->
        <csr>
        (<licences>)? <!-- Optional for some products
-->
        (<approverEmail>)?
            <!-- Not required for all SSL
certificates -->
        (<san>)*
        (<reference>)?
        (<orderNumber>)?<!-- Only required for
GlobalSign DV products -->
        (<certificateType>)? <!-- Only required for
CodeSigning Certicates →
        (<commonName>)? <!-- GlobalSign
PersonalSign/CodeSigning -->
        (<title>)? <!-- GlobalSign
PersonalSign/CodeSigning -->
        (<organization>)? <!-- GlobalSign
PersonalSign/CodeSigning →
        (<organizationUnit>)? <!-- GlobalSign
CodeSigning →
        (<locality>)? <!-- GlobalSign
PersonalSign/CodeSigning →
        (<stateOrProvince>)? <!-- GlobalSign
PersonalSign/CodeSigning

```

```

        (<country>)? <!-- GlobalSign
PersonalSign/CodeSigning →
        (<email>)? <!-- GlobalSign
PersonalSign/CodeSigning →
        (<extraPreferences>)? <!-- Optional for some
products -->
        (<option>
            (<approverMethod>)? <!-- Optional for
some Symantec/Networking4all DV products , FILE, EMAIL,
defaults to EMAIL -->
            </option>)?
        </request>
</orderSSLCertificate>

```

5.3.2 orderSSLCertificate response

```

<orderSSLCertificateResponse
xmlns="https://www.networking4all.com/api/soap/api.wsd1">
    <responseHeader>
        <resultCode>
            (<resultMessage>)?
            (<warnings>
                (<warning>
                    <code>
                    <message>
                    <field>
                </warning>)+
            </warnings>)?
            (<errors>
                (<error>
                    <code>
                    <message>
                    <field>
                </error>)+
            </errors>)?
            <timestamp>
            (<language>)?
            (<reference>)?
            (<test>)?
        </responseHeader>
        (<response>
            <traceNumber>          <!-- TraceNumber required
for resendEmailSSLCertificate and replaceSSLCertificate -->
            (<orderNumber>)?      <!-- Only for direct order
-->
            (<pickupPassword>)? <!-- GlobalSign
PersonalSign/CodeSigning →
            (<fileName>)? <!-- Filename for FILE DV
approver -->
            (<fileContent>)? <!-- File content for FILE DV
approver -->
        </response>)? <!-- response will only be given when
resultCode = 0 -->
</orderSSLCertificateResponse>

```

5.4 resendEmailSSLCertificate

The resendEmailSSLCertificate function allows you to resend the approver email, or to edit it. Editing the approver email address is only available for products that use email validation. When this functionality is not available for a product, the entered approver email address will be ignored.

5.4.1 resendEmailSSLCertificate request

```

<resendEmailSSLCertificate
xmlns="https://www.networking4all.com/api/soap/api.wsdl">
  <requestHeader>
    <auth>
      <handle>      <!-- Login handle -->
      <username>
      <password>
    </auth>
    (<language>)?
    (<reference>)?
    (<test>)?
  </requestHeader>
  <request>
    <traceNumber>      <!-- TraceNumber as given in
orderSSLCertificate response -->
    (<approverEmail>)? <!-- Optional, only use when
you want to edit the approver email address -->
  </request>
</resendEmailSSLCertificate>

```

5.4.2 resendEmailSSLCertificate response

```

<resendEmailSSLCertificateResponse
xmlns="https://www.networking4all.com/api/soap/api.wsdl">
  <responseHeader>
    <resultCode>
    (<resultMessage>)?
    (<warnings>
      (<warning>
        <code>
        <message>
        <field>
      </warning>)+
    </warnings>)?
    (<errors>
      (<error>
        <code>
        <message>
        <field>
      </error>)+
    </errors>)?
    <timestamp>
    (<language>)?
    (<reference>)?
    (<test>)?
  </responseHeader>
  (<response>
    <traceNumber>
    <orderNumber>
    (<approverEmail>)?
  </response>)?<!-- response will only be given when
resultCode = 0 -->
</resendEmailSSLCertificateResponse>

```


5.5 getSSLCertificate

The getSSLCertificate function allows you to check the status of your SSL order, and, if the order has been completed (the order has status code 216), to retrieve the SSL certificate through our API.

5.5.1 getSSLCertificate request

```
<getSSLCertificate
xmlns="https://www.networking4all.com/api/soap/api.wsdl">
  <requestHeader>
    <auth>
      <handle>      <!-- Your location handle
-->
      <username>
      <password>
    </auth>
    (<language>)?
    (<reference>)?
    (<test>)?
  </requestHeader>
  <request>
    <traceNumber>
      <!-- TraceNumber as given in
orderSSLCertificate response -->
    (<option>
      (<returnAllSSLSubstatus>)?
    </option>)?
  </request>
</getSSLCertificate>
```

5.5.2 getSSLCertificate response

```
<getSSLCertificateResponse
xmlns="https://www.networking4all.com/api/soap/api.wsdl">
  <responseHeader>
    <resultCode>
    (<resultMessage>)?
    (<warnings>
      (<warning>
        <code>
        <message>
        <field>
      </warning>)+
    </warnings>)?
    (<errors>
      (<error>
        <code>
        <message>
        <field>
      </error>)+
    </errors>)?
    <timestamp>
    (<language>)?
    (<reference>)?
    (<test>)?
  </responseHeader>
  (<response>
    <traceNumber>
    <Number>
```

```

    <status>
      <code>
        (<message>)?
        (<substatuses>
          (<substatus>
            <code>
              <name>
              <startDate>
              (<endDate>)? <!-- if status
is completed -->
              (<comment>)? <!-- comments
from vetting team -->
            </substatus>)+
          </substatuses>)?
    </status>
    <domain>
    (<sanNames>
      (<san>)+
    </sanNames>)?
    (<certificateType>)? <!-- CodeSigning -->
    (<commonName>)? <!-- GlobalSign
PersonalSign/CodeSigning -->
    (<title>)? <!-- GlobalSign
PersonalSign/CodeSigning -->
    (<organization>)? <!-- GlobalSign
PersonalSign/CodeSigning →
    (<locality>)? <!-- GlobalSign
PersonalSign/CodeSigning →
    (<stateOrProvince>)? <!-- GlobalSign
PersonalSign/CodeSigning →
    (<country>)? <!-- GlobalSign
PersonalSign/CodeSigning →
    (<email>)? <!-- GlobalSign
PersonalSign/CodeSigning →
    (<pickupPassword>)? <!-- GlobalSign
PersonalSign/CodeSigning →
    <products>
      (<product>
        <productCode>
        <name>
        <period>
        <description>
          <short>
          (<long>)?
        </description>
        <fees>
          <setup> <!-- DEPRECATED, see
fee structure -->
            <retail>
            <actual>
          </setup>
          (<fee>
            <period>
            <retail>
            <actual>
            <type>
            <currency>
          </fee>)+
        </fees>
        (<vat>)?
        (<details>
          <mailValidation>
          <dv>
          <ov>
          <ev>
          <webserver>

```

```

        <company>
        <codesigning>
        <san>
        <wildcard>
        <intranet>
        <mobile>
        <competitiveUpgrade>
        (<details>)?
        (<total>)?
    </product>)+
    (<total>)?
</products>
<period>
<startDate>
<expDate>
(<reference>)?
(<paymentMethod>)?
<company>
<contact>
<adminContact>
<techContact>
<fulfillment> <!-- fulfillment only when status
= 216 -->
    <certificate>
    (<caCertificates>
        (<caCertificate>
            (<type>)?
            (<name>)?
            <certificate>
        </caCertificate>)+
    </caCertificates>)?
</fulfillment>
(<extraServices>
    (<extraPreferences>)?
    (<extraServicesResults>
        (<lastScanDate>)?
        (<sealStatus>)?
        (<scanStatus>)?
        (<resultsFound>)?
        (<scanResults>
            (<infection>
                <type>
                <date>
                <url>
                (<description>)?
            </infection>)+
        </scanResults>)?
    </extraServicesResults>)?
</extraServices>)?
</response>)?<!-- response will only be given when
resultCode = 0 -->
</getSSLCertificateResponse>

```

5.6 getSSLCertificateList

The `getSSLCertificateList` function allows you to retrieve a list with one or multiple SSL orders, and to check the status of this order or these orders. You can add several filters to define the list, such as start- and end date, status, and handle. If the order has been completed (the order has status code 216), you can retrieve the certificate through our API.

5.6.1 getSSLCertificateList request

```

<getSSLCertificateList
xmlns="https://www.networking4all.com/api/soap/api.wsdl">
  <requestHeader>
    <auth>
      <handle>      <!-- Login handle -->
      <username>
      <password>
    </auth>
    (<language>)?
    (<reference>)?
    (<test>)?
  </requestHeader>
  (<request>
    (<fromDate>)?
    (<toDate>)?
    (<fromExpDate>)?
    (<toExpDate>)?
    (<status>)?
    (<handle>)?
  </request>)?
</getSSLCertificateList>

```

5.6.2 getSSLCertificateList response

```

<getSSLCertificateListResponse
xmlns="https://www.networking4all.com/api/soap/api.wsdl">
  <responseHeader>
    <resultCode>
    (<resultMessage>)?
    (<warnings>
      (<warning>
        <code>
        <message>
        <field>
      </warning>)+
    </warnings>)?
    (<errors>
      (<error>
        <code>
        <message>
        <field>
      </error>)+
    </errors>)?
    <timestamp>
    (<language>)?
    (<reference>)?
    (<test>)?
  </responseHeader>
  (<response>
    (<fromDate>)?
    (<toDate>)?
    (<fromExpDate>)?
    (<toExpDate>)?
    (<status>)?
    (<handle>)?
    (<result>
      <traceNumber>
      <Number>
      <status>
        <code>
        (<message>)?
    )
  )

```

```

        (<substatuses>
            (<substatus>
                <code>
                <name>
                <startDate>
                (<endDate>)? <!-- if
status is completed -->
                (<comment>)? <!--
comments from vetting team -->
            </substatus>)+
        </substatuses>)?
    </status>
    <domain>
    (<sanNames>
        (<san>)+
    </sanNames>)?
    (<certificateType>)? <!-- CodeSigning -->
    (<commonName>)? <!-- GlobalSign
PersonalSign/CodeSigning -->
    (<title>)? <!-- GlobalSign
PersonalSign/CodeSigning -->
    (<organization>)? <!-- GlobalSign
PersonalSign/CodeSigning →
    (<locality>)? <!-- GlobalSign
PersonalSign/CodeSigning →
    (<stateOrProvince>)? <!-- GlobalSign
PersonalSign/CodeSigning
    (<country>)? <!-- GlobalSign
PersonalSign/CodeSigning →
    (<email>)? <!-- GlobalSign
PersonalSign/CodeSigning →
    (<pickupPassword>)? <!-- GlobalSign
PersonalSign/CodeSigning →
    <products>
        (<product>
            <productCode>
            <name>
            <period>
            <description>
                <short>
                (<long>)?
            </description>
            <fees>
                <setup> <!--
DEPRECATED, see fee structure -->
                <retail>
                <actual>
            </setup>
            (<fee>
                <period>
                <retail>
                <actual>
                <type>
                <currency>
            </fee>)+
        </fees>
    (<vat>)?
    (<details>
        <mailValidation>
        <dv>
        <ov>
        <ev>
        <webserver>
        <company>
        <codesigning>
        <san>

```

```

        <wildcard>
        <intranet>
        <mobile>
        <competitiveUpgrade>
        (<details>)?
        (<total>)?
    </product>)+
    (<total>)?
</products>
<period>
<startDate>
<expDate>
(<reference>)?
(<paymentMethod>)?
<company>
<contact>
<adminContact>
<techContact>
(<fulfillment> <!-- Only returned when
status = 216 -->
    <certificate>
    (<caCertificates>
        (<caCertificate>
            (<type>)?
            (<name>)?
            <certificate>
        </caCertificate>)+
    </caCertificates>)?
</fulfillment>)?
(<extraServices>
    (<extraPreferences>)?
    (<extraServicesResults>
        (<lastScanDate>)?
        (<sealStatus>)?
        (<scanStatus>)?
        (<resultsFound>)?
        (<scanResults>
            (<infection>
                <type>
                <date>
                <url>
            </infection>)+
        </scanResults>)?
    </extraServicesResults>)?
    </extraServices>)?
</result>)+
</response>)?<!-- response will only be given when
resultCode = 0 -->
</getSSLCertificateListResponse>

```

5.7 replaceSSLCertificate

The replaceSSLCertificate function allows you to replace an SSL certificate.

5.7.1 replaceSSLCertificate request

```

<replaceSSLCertificate
xmlns="https://www.networking4all.com/api/soap/api.wsd1">
  <requestHeader>

```

```

    <auth>
        <handle>      <!-- Login handle -->
        <username>
        <password>
    </auth>
    (<language>)?
    (<reference>)?
    (<test>)?
</requestHeader>
<request>
    <traceNumber>      <!-- TraceNumber as given in
orderSSLCertificate response -->
    <csr>
    <approverEmail>
    (<option>
        (<approverMethod>)? <!-- Optional for
some Symantec/Networking4all DV products , FILE, EMAIL,
defaults to EMAIL -->
    </option>)?
    </request>
</replaceSSLCertificate>

```

5.7.2 replaceSSLCertificate response

```

<replaceSSLCertificateResponse
xmlns="https://www.networking4all.com/api/soap/api.wsdl">
    <responseHeader>
        <resultCode>
        (<resultMessage>)?
        (<warnings>
            (<warning>
                <code>
                <message>
                <field>
            </warning>)+
        </warnings>)?
        (<errors>
            (<error>
                <code>
                <message>
                <field>
            </error>)+
        </errors>)?
        <timestamp>
        (<language>)?
        (<reference>)?
        (<test>)?
    </responseHeader>
    (<response>
        <traceNumber>
        <orderNumber>
        (<fileName>)? <!-- Filename for FILE DV
approver -->
        (<fileContent>)? <!-- File content for FILE DV
approver
    </response>)? <!-- response will only be given when
resultCode = 0 -->
</replaceSSLCertificateResponse>

```

5.8 changeSANSSLCertificate

The changeSANSSLCertificate function allows you to add SAN-names to your SSL certificate. Please note: when using this function, you must include all SAN-names you wish to keep, and add the new SAN-names.

5.8.1 changeSANSSLCertificate request

```
<changeSANSSLCertificate
xmlns="https://www.networking4all.com/api/soap/api.wsdl">
  <requestHeader>
    <auth>
      <handle>      <!-- Login handle -->
      <username>
      <password>
    </auth>
    (<language>)?
    (<reference>)?
    (<test>)?
  </requestHeader>
  <request>
    <traceNumber>      <!-- TraceNumber as given in
orderSSLCertificate response -->
    (<san>)*           <!-- Make sure that
ALL the SAN names you want to keep are included -->
  </request>
</changeSANSSLCertificate>
```

5.8.2 changeSANSSLCertificate response

```
<changeSANSSLCertificateResponse
xmlns="https://www.networking4all.com/api/soap/api.wsdl">
  <responseHeader>
    <resultCode>
    (<resultMessage>)?
    (<warnings>
      (<warning>
        <code>
        <message>
        <field>
      </warning>)+
    </warnings>)?
    (<errors>
      (<error>
        <code>
        <message>
        <field>
      </error>)+
    </errors>)?
    <timestamp>
    (<language>)?
    (<reference>)?
    (<test>)?
  </responseHeader>
  (<response>
    <traceNumber>
    <orderNumber>
    </response>)?<!-- response will only be given when
resultCode = 0 -->
</changeSSLCertificateResponse>
```


6. Additional API information

The following XML notation will be used in this document:

() will be followed by either the *, ?, or + symbol, to determine the cardinality:

- * stands for 0 or more
- ? stands for 0 or 1
- + stands for 1 or more

The notation <!-- --> is used for comments within the code.

6.1 the requestHeader

Any API request needs to contain the following requestHeader in the format as seen below:

```

        <requestHeader>
    <auth>
        <handle>      <!-- Login handle -->
        <username>
        <password>
    </auth>
    (<language>)?
    (<reference>)?
    (<test>)?
</requestHeader>

```

The <auth>-structure must contain the handle, username, and password. Without this information, the API cannot be accessed. The other functions are optional. The <language>-structure can be used to specify the language of the error messages in the API response. This setting is set to English by default. The <reference>-structure can contain a string that you can use as a reference in your own system.

Please note: If you want to use the test environment, the option <test> must be added to the requestHeader with the value '1'. When this option is left blank, or left out of the requestHeader, you will automatically connect with the live environment.

6.2 the responseHeader

The Networking4all API includes a responseHeader in every response, in the following format:

```

<responseHeader>
    <resultCode>
    (<resultMessage>)?
    (<warnings>
        (<warning>
            <code>
            <message>
            <field>
        </warning>)+
    </warnings>)?
    (<errors>
        (<error>

```

```

        <code>
        <message>
        <field>
    </error>)+
</errors>)?
<timestamp>
(<language>)?
(<reference>)?
(<test>)?
</responseHeader>

```

6.2.1 resultCode

The <resultCode> and <resultMessage> show the status of the process. They can contain the following values:

resultCode	resultMessage
0	The task has been performed successfully.
1	The task has been performed, no results have been found.
2	The task could not be performed, errors have occurred, please see error structure.
3	The API is currently in maintenance mode.

6.2.2 warningCodes

The <warning>-codes show warnings. They can contain the following values:

Code	Message
1000	Function is deprecated.
1001	Field is deprecated.

6.2.3 errorCodes

The <error>-structure can contain the following information:

- <code>: a numeric error code indicating the type of error.
- <message>: a more detailed error message, in the chosen language.
- <field>: the field in which the error occurred.

Code	Message
101	Value is empty.
102	Value is not alphanumerical.
103	Not an alphabetic character.
104	Not within the limit.
105	Not strictly within the limit.

106	Value is not numerical.
107	Value is not a float.
108	Not larger than
109	Value is not hexadecimal.
110	Token is missing.
111	Values are not equal.
112	Not in array.
113	Value is not an integer.
114	Not an IP address.
115	Not smaller than
116	Value is an empty string.
1001	Parameter is missing.
1002	Value is not valid.
1003	Value is too short.
1004	Value is too long.
1005	Value does not exist.
1006	Value already exists.
2001	Email validation error.
2002	Host name validation error.
2003	Username validation error.
2004	Password strength error.
2005	Creditcard validation error.
3001	SSL certificate error.
3002	Nameserver error.
3003	Domein name error.
4001	Login error.
5001	Unknown error.

6.3 Field definitions

The following table contains all API request and response fields in alphabetical order.

Veld	Beschrijving	Type	Min	Max
<pre><account> <username> <password> <passwordConfirm> </account></pre>	Account structure			
<accountNumber>	Bank account number			
<accountType>	Bank type: 'GIRO', 'BANK' or 'IBAN'.	string		
<actual>	Current price, including any discounts	string		
<pre><address> <type> <street> <streetNumber> (<streetNumberExtension>? <street1> (<street2>)? (<street3>)? (<buildingName>)? <postalCode> <city> <province> <countryCode> </address></pre>	Address structure. Use <street1> instead of <street>, <streetNumber> and <streetNumberExtension>.			
<adminHandle>	Admin contact handle	string		10
<approverEmail>	Approver email for SSL certificate	string		
<approvers>	Approver email response	string		
<pre><audit> <created> (<createdHandle>)? (<updated>)? (<updatedHandle>)? </audit></pre>	Audit information structure response for getRelation			
<pre><audit> <created> (<createdHandle>)? (<updated>)? (<updatedHandle>)? </audit></pre>	Authentication structure, part of requestHeader			
<bicCode>	BIC code	string		
<birthDate>	Birth date in JJJJ-MM-DD	string		

	format			
<buildingName>	Building name	string	3	255
<caCertificates> (<caCertificate> (<type>)? (<name>)? <certificate> </caCertificate>)+ </caCertificates>	Contains the root and/or intermediate certificates.			
<chamberOfCommerce>	Chamber of Commerce number	string		
<certificate>	Contains SSL certificate	string		
<certificateType>	Only for Code Signing certificates. Allowed types: ADOBE_AIR, APPLE (only Thawte), JAVASOFT, MS_AUTHENTICODE, VBA	string		
<city>	City name	String	3	255
<clients>	Reseller client handles	String		10
<clientNumber>	Client number	string		
<cn>	Common Name	string		
<company> <name> (<employees>)? (<chamberOfCommerce>)? </company>	Structure for company data			
<companyHandle>	Company handle	String		10
<contact> <title> <initials> <firstName> (<middleName>)? <lastName> (<gender>)? (<birthDate>)? </contact>	Structure for contact data			
<contacts>	Contact handle	String		10
<country>	Country for GlobalSign PersonalSign/CodeSigning	String		2
<countryCode>	Country code, ISO 3166-2 default	String		2
<countryCode>	Country code as part of phone/fax/mobile number	Integer		3
<createAccount>	Option for the creation of account information, value 0	Integer		

	or 1.			
<created>	Date, in JJJ-MM-DD format	String		
<createdHandle>	Contact handle	String		10
<csr>	Valid CSR, including opening and closing lines	String		
<description> <short> (<long>)? </description>	Structure for product description			
<details> <mailValidation> <dv> <ov> <ev> <webserver> <company> <codesigning> <san> <wildcard> <intranet> <mobile> <competitiveUpgrade> </details>	Structure for SSL certificate product details			
<domain>	Valid FQDN	string		
<email>	Email address	string		
<employees>	Number of employees	Integer		
<extension>	Extension of a phone/fax/mobile number	integer		
<fee> <period> <retail> <actual> <type> <currency> </fee>	Structure for fee, contains period in months, price and price including VAT, and any discounts.			
<fees> <setup> <retail> <actual> </setup> (<fee> <period> <retail> <actual> <type> <currency> </fee>)+ </fees>	Structure for fees.			
<faxNumber>	Fax number	string		
<firstName>	First name of contact	String	2	20
<financial> <accountNumber>	Structure for financial			

<code><accountType> (<bicCode>)? (<ibanNumber>)? (<vatin>)? </financial></code>	information			
<code><fulfillment> <certificate> (<caCertificates> (<caCertificate> (<type>)? (<name>)? <certificate> </caCertificate> + </caCertificates>)? </fulfillment></code>	Structure for fulfillment, part of getSSLCertificate function			
<code><gender></code>	Gender, can contain "M" of "W"	String		1
<code><handle></code>	Relation handle	String		10
<code><ibanNumber></code>	IBAN	String		
<code><infection> <type> <date> <url> (<description>)? </infection></code>	Structure for infection			
<code><initials></code>	Initials	String		
<code><ipv4></code>	IPv4-address	String		
<code><ipv6></code>	IPv6-address	string		
<code><lastName></code>	Last name of contact	String	2	20
<code><language></code>	Language code	String		2
<code><licences></code>	Server licences, optional for some certificates	Integer		
<code><locality></code>	Location for GlobalSign PersonalSign/CodeSigning	String		
<code><long></code>	Long description	string		
<code><middleName></code>	Middle name of contact	String	2	20
<code><mobileNumber></code>	Mobile number	String		
<code><name></code>	Company name	String		
<code><option> <createAccount> </option></code>	Optional structure			
<code><orderNumber></code>	Order number of CA	String		
<code><organisation></code>	Organisation name	String		

<organisationUnit>	Organisation unit/afdeling	String		
<passportNumber>	Document number passport	String		
<password>	Password	String	8	8
<passwordConfirm>	Confirm password	String	8	8
<paymentMethod>	Payment method	String		
<period>	Period in years	Integer		
<period>	Period in months (as returned in structure for fee)			
<phoneNumber>	Phone number	String		
<postalCode>	Postal code	String		
<pre> <products> (<product> <productCode> <name> <period> <description> <short> (<long>)? </description> <fees> <setup> <retail> <actual> </setup> (<fee> <period> <retail> <actual> </fee>)+ </fees> (<vat>)? (<total>)? </product>)+ (<total>)? </products> </pre>	Product structure, part of getSSLCertificate function			
<pre> <products> (<product> <code> <name> (<brand>)? <!-- For SSL only --> </product>)+ </products> </pre>	Structure of product for getProducts function			
<productCode>	Product code	string		
<province>	Province	String		
<reference>	Reference	String		

<code><reseller></code>	Contains value 'Y' in response when contact is reseller	String		
<code><resellerHandle></code>	Reseller handle	String		10
<code><result></code>	Number of results	Integer		
<code><retail></code>	Price without discounts	String		
<code><san></code>	FQDN SAN	String		
<code>(<sanNames> (<san>)+ </sanNames>)?</code>	Structure for SAN names			
<code><sealStatus></code>	Seal status. Can contain following values: SECURE_SEAL, TRUST_SEAL, OFF	String		
<code><setup> <retail> <actual> </setup></code>	Structure for setup			
<code><short></code>	Short description	String		
<code><status></code>	Status of contact	integer		
<code><status> <code> (<message>)? (<substatusses> (<substatus> <code> <name> <startDate> (<endDate>)? <!-- when status is completed --> (<comment>)? <!-- comment from vetting team --> </substatus>)+ </substatusses>)? </status></code>	Structure for status of getSSLCertificate and SSLCertificateList			
<code><stateOrProvince></code>	State or province for GlobalSign PersonalSign/CodeSigning	string		
<code><street1></code>	Full address (street name, postal code, city)	string		
<code><street2></code>	Optional	String		
<code><street3></code>	Optional	string		
<code><subscriberNumber></code>	Subscriber number for	Integer		10

	phone/fax/mobile number			
<techHandle>	Handle technical contact	String		10
<test>	Toggle function for test or live environment. For value 1, the connection will be made with the test environment.	Integer		
<title>	Title for contact	String		
<traceNumber>	Trace number used to replace SSL certificate or edit email approver	String		
<trunkCode>	Net number of a telephone/fax/mobile number	Integer		10
<type>	Relation type, "contact" or "company"	string		
<type>	Structure of address, "visitor" or "postal"	String		
<type>	Price type in getProduct function. Possible values: setup, new, registration, transfer, trade, restore, renew	String		
<updated>	Date in JJJJ-MM-DD format	String		
<updatedHandle>	Contact handle	String		10
<username>	Username	String	4	8
<vat>	VAT percentage	integer		
<vatIn>	VAT number	string		
<website>	Website URL	string		

6.4 code examples SSL certificates

The following code examples are written in PHP and show the steps taken to order an SSL certificate. The product used in this example is the Networking4all Basic SSL certificate, product code 2005, for a period of one year.

6.4.1 Create company

For the purpose of this example, we will first create a new company.

```
<?php

error_reporting(E_ALL);

ini_set("soap.wsdl_cache_enabled", "0");
use_soap_error_handler(true);

$url = 'https://www.networking4all.com/api/soap/api.wsdl';

$client = new SoapClient($url, array('trace' => 1));

$requestHeader = array();
$requestHeader['auth']['handle'] = utf8_encode('TST-000000');
$requestHeader['auth']['username'] = utf8_encode('username');
$requestHeader['auth']['password'] = utf8_encode('password');

$requestHeader['language'] = utf8_encode('NL');
$requestHeader['reference'] = utf8_encode('My Reference');
$requestHeader['test'] = 1;

$params = array();
$params['requestHeader'] = $requestHeader;

$params['request']['type'] = utf8_encode('company');

/*
 * Company information
 */
$params['request']['company']['name'] = utf8_encode('Test company');
$params['request']['company']['employees'] = 2;
$params['request']['company']['chamberOfCommerce'] = '';

/*
 * Address information
 */
$address = array();
$address['type'] = utf8_encode('visit');
$address['street1'] = utf8_encode('Testlaan 562');
$address['postalCode'] = utf8_encode('0000 EN');
$address['city'] = utf8_encode('Den Haag');
$address['province'] = utf8_encode('Zuid Holland');
$address['countryCode'] = utf8_encode('NL');

$params['request']['address'][] = $address;

/*
 * Address information (optional)
 */
```

```

 */
$address = array();
$address['type'] = utf8_encode('post');
$address['street1'] = utf8_encode('Postbus 00000');
$address['postalCode'] = utf8_encode('0000 EN');
$address['city'] = utf8_encode('Amsterdam');
$address['province'] = utf8_encode('Noord Holland');
$address['countryCode'] = utf8_encode('NL');

$params['request']['address'][] = $address;

/*
 * Phone number
 */
$params['request']['phoneNumber'] = '31207000000';

/*
 * Fax number
 */
$params['request']['faxNumber'] = '31207000000';

/*
 * Mobile number
 */
$params['request']['mobileNumber'] = '316001000000';

/*
 * Digital contact information
 */
$params['request']['email'] =
utf8_encode('info@testcompany.com');
$params['request']['website'] = '';

/*
 * Back financial information
 */
$params['request']['financial']['accountNumber'] =
'0000000';
$params['request']['financial']['accountType'] =
utf8_encode('GIRO');
$params['request']['financial']['bicCode'] = '';
$params['request']['financial']['ibanNumber'] = '';
$params['request']['financial']['vatin'] = '';

$response = $client->createRelation($params);

?>

```

The response given to this request is a handle: TST-000001 (please note: this handle is an example).

6.4.2 Create contact

The next step is to create a contact for our newly created company.

```

<?php
error_reporting(E_ALL);

ini_set("soap.wsdl_cache_enabled", "0");
use_soap_error_handler(true);

$url = 'https://www.networking4all.com/api/soap/api.wsdl';

```

```

$client = new SoapClient($url, array('trace' => 1));

$requestHeader = array();
$requestHeader['auth']['handle'] = utf8_encode('TST-000000');
$requestHeader['auth']['username'] = utf8_encode('username');
$requestHeader['auth']['password'] = utf8_encode('password');

$requestHeader['language'] = utf8_encode('NL');
$requestHeader['reference'] = utf8_encode('My Reference');
$requestHeader['test'] = 1;

$params = array();
$params['requestHeader'] = $requestHeader;

$params['request']['type'] = utf8_encode('contact');

/*
 * Handle to link this contact to company
 */
$params['request']['handle'] = utf8_encode('TST-000001');

/*
 * Contact
 */
$params['request']['contact']['title'] = utf8_encode('Dhr');
$params['request']['contact']['initials'] = utf8_encode('D');
$params['request']['contact']['firstName'] = utf8_encode('David');
$params['request']['contact']['middleName'] = '';
$params['request']['contact']['lastName'] = utf8_encode('Test');
$params['request']['contact']['gender'] = utf8_encode('M');
$params['request']['contact']['birthDate'] = utf8_encode('1981-01-01');

/*
 * Address information
 */
$address = array();
$address['type'] = utf8_encode('visit');
$address['street1'] = utf8_encode('Testlaan'0 562);
$address['postalCode'] = utf8_encode('0000 EN');
$address['city'] = utf8_encode('Den Haag');
$address['province'] = utf8_encode('Zuid Holland');
$address['countryCode'] = utf8_encode('NL');

$params['request']['address'][] = $address;

/*
 * Phone number
 */
$params['request']['phoneNumber'] = '31207000000';

/*
 * Fax number
 */
$params['request']['faxNumber'] = '31207000000';

/*
 * Mobile number
 */

```

```

$params['request']['mobileNumber'] = '316001000000';

/*
 * Digital contact information
 */
$params['request']['email'] =
utf8_encode('david@yourdomain.com');
$params['request']['website'] = '';

/*
 * Login account
 */
$params['request']['account']['username'] =
utf8_encode('daid81');
$params['request']['account']['password'] =
utf8_encode('paS$wd81');
$params['request']['account']['passwordConfirm'] =
utf8_encode('paS$wd81');

/*
 * Options
 */
$params['request']['option']['createAccount'] = 1;

$response = $client->createRelation($params);

?>

```

The response given to this request is a handle: TST-000002 (please note: this handle is an example).

6.4.3 CSR

The CSR must be checked for validity and correctness before the order can be placed.

```

<?php

error_reporting(E_ALL);

ini_set("soap.wsdl_cache_enabled", "0");
use_soap_error_handler(true);

$url = 'https://www.networking4all.com/api/soap/api.wsdl';

$client = new SoapClient($url, array('trace' => 1));

$requestHeader = array();
$requestHeader['auth']['handle'] = utf8_encode('TST-
000000');
$requestHeader['auth']['username'] =
utf8_encode('username');
$requestHeader['auth']['password'] =
utf8_encode('password');

$requestHeader['language'] = utf8_encode('NL');
$requestHeader['reference'] = utf8_encode('My Reference');
$requestHeader['test'] = 1;

$csr = "-----BEGIN CERTIFICATE REQUEST-----
MIIB5jCCAQU8CAQAwgaUxCzAJBgNVBAYTAk5MMRYwFAYDVQQIEw1Ob29yZCBC
cmFi
YW50MRMwEQYDVQQHEWpSb29zZW5kYW5kYWFsMRwwGgYDVQQKEwNOZXR3b3JraW5n
NGFs

```

```

bCBCLlYuMR8wHQYDVQQDEExZ3d3cubmV0d29ya2luZzRhGwuY29tMSowKAYJ
KoZI
hvcNAQkBFhtzZWN1cm10eUBuZXR3b3JraW5nNGFsbC5jb20wgZ8wDQYJKoZI
hvcN
AQEBBQADgY0AMIGJAoGBA0si6VgMka2KLata8fn0zeXbwbmjcTKd6V973Jrm
4mCk
PAjFklWyGdXirtHV+Etbuu7/WAEgwCVJK46tjMK+ZPpmt1f/kXxIUgjtZTa
TS+r
Cfzoyl85MZSKXNhIMOICvkn7m4YXyQDMXLearR+XVa2pnJ0MwpEnAeh8YEsp
s/Pp
AgMBAAGgADANBgkqhkiG9w0BAQQFAAOBgQC3lC2dW6smhPD3SFXy0ypw9f7x
nieL
cosYjN00/yos9N1f1GsV0LuZhfgR7yZhUvLyWqM6lhZpL6f7jTUD9IstvAhT
UpLq
LY8ZNI0p00uQDljmqoQnTk1tro2dblzDTWmhlOm2BobW+aU+LnLS9oDJ8gBP
KE1V
J4beS3a2eNwORg==
-----END CERTIFICATE REQUEST-----";

$params = array();
$params['requestHeader'] = $requestHeader;
$params['request']['productCode'] = 2005;
$params['request']['csr'] = utf8_encode($csr);

$response = $client->checkSSLCertificateCSR($params);
?>

```

6.4.4 The approver email address

Once the CSR is checked, the list of approval email addresses can be retrieved. One of these email addresses must be used in the order process of the SSL certificate. For this, we need the domain name for which the SSL certificate is being ordered. This is also called the Common Name. For this example, we will be using **www.networking4all.com** as our Common Name.

```

<?php

error_reporting(E_ALL);

ini_set("soap.wsdl_cache_enabled", "0");
use_soap_error_handler(true);

$url = 'https://www.networking4all.com/api/soap/api.wsdl';

$client = new SoapClient($url, array('trace' => 1));

$requestHeader = array();
$requestHeader['auth']['handle'] = utf8_encode('TST-
000000');
$requestHeader['auth']['username'] =
utf8_encode('username');
$requestHeader['auth']['password'] =
utf8_encode('password');

$requestHeader['language'] = utf8_encode('NL');
$requestHeader['reference'] = utf8_encode('My Reference');
$requestHeader['test'] = 1;

$params = array();
$params['requestHeader'] = $requestHeader;
$params['request']['productCode'] = 2005;
$params['request']['domain'] =
utf8_encode('www.networking4all.com');

```



```
$response = $client->getSSLCertificateApproverList($params);
?>
```

6.4.5 Order the SSL certificate

All the data for the SSL certificate has now been collected, and the order can be placed through the API. This data is as follows:

```
Company:          TST-000001
Admin Contact:    TST-000002
Tech Contact:     TST-000002
ApproverEmail:    hostmaster@networking4all.com
```

We will be using the same contact for our Admin and Tech contact in this example. Of course, you can create separate contacts for these options.

The order can now be placed:

```
<?php

error_reporting(E_ALL);

ini_set("soap.wsdl_cache_enabled", "0");
use_soap_error_handler(true);

$url = 'https://www.networking4all.com/api/soap/api.wsdl';

$client = new SoapClient($url, array('trace' => 1));

$requestHeader = array();
$requestHeader['auth']['handle'] = utf8_encode('TST-000000');
$requestHeader['auth']['username'] = utf8_encode('username');
$requestHeader['auth']['password'] = utf8_encode('password');

$requestHeader['language'] = utf8_encode('NL');
$requestHeader['reference'] = utf8_encode('My Reference');
$requestHeader['test'] = 1;

$csr = "-----BEGIN CERTIFICATE REQUEST-----
MIIB5jCCAU8CAQAwwgaUxCzAJBgNVBAYTAK5MMRYwFAYDVQQIEw1Ob29yZCBC
cmFi
YW50MRMwEQYDVQQHEwpSb29zZW5kYWVFsMRwwGgYDVQQKEwNOZXR3b3JraW5n
NGFs
bCBCLlYuMR8wHQYDVQQDEwZ3d3cubmV0d29ya2luZzRhGwuY29tMSowKAYJ
KoZI
hvcNAQkBFhtzZWN1cm10eUBuZXR3b3JraW5nNGFsbc5jb20wgZ8wDQYJKoZI
hvcN
AQEBBQADgY0AMIGJAoGBAOsi6VgMka2KLata8fn0zeXbwbmjcTKd6V973Jrm
4mCk
PAjFklWyGdXirtHV+Etbu7/WAEgwCVJK46tjMK+ZPpmt1f/kXxIUgjtZTa
TS+r
Cfzoyl85MZSKXNhIMOICvkn7m4YXyQDMXLearR+XVa2pnJ0MwpEnAeh8YEsp
s/Pp
AgMBAAAGgADANBgkqhkiG9w0BAQQFAAOBgQC3lC2dW6smhPD3SFXy0ypw9f7x
nieL
cosYjN00/yos9N1flGsV0LuZhfgR7yZhUvLyWqM6lhzpL6f7jTUD9IStvAhT
UpLq
LY8ZNI0p00uQDljqmoQnTk1tro2dblzDTWmhlOm2BobW+aU+LnLS9oDJ8gBP
KE1V
J4beS3a2eNwORg==
```

```

-----END CERTIFICATE REQUEST-----";

$params = array();
$params['requestHeader'] = $requestHeader;

$params['request']['companyHandle'] = utf8_encode('TST-000001');
$params['request']['adminHandle'] = utf8_encode('TST-000002');
$params['request']['techHandle'] = utf8_encode('TST-000002');
$params['request']['productCode'] = 2005;
$params['request']['csr'] = utf8_encode($csr);
$params['request']['period'] = 1;
$params['request']['approverEmail'] =
utf8_encode('hostmaster@networking4all.com');

$response = $client->orderSSLCertificate($params);

?>

```

After the function has been processed, the API will return a response containing a traceNumber: **1000000001**. This traceNumber is used in the functions resendEmailSSLCertificate, replaceSSLCertificate, and getSSLCertificateOrder.

6.4.6 Order SSL certificate (DV) with automatic Domain Approval

All the data for the SSL certificate has now been collected, and the order can be placed through the API. This data is as follows:

Company:	TST-000001
Admin Contact:	TST-000002
Tech Contact:	TST-000002

We will be using the same contact for our Admin and Tech contact in this example. Of course, you can create separate contacts for these options.

The order can now be placed:

```

<?php

error_reporting(E_ALL);

ini_set("soap.wsdl_cache_enabled", "0");
use_soap_error_handler(true);

$url = 'https://www.networking4all.com/api/soap/api.wsdl';

$client = new SoapClient($url, array('trace' => 1));

$requestHeader = array();
$requestHeader['auth']['handle'] = utf8_encode('TST-000000');
$requestHeader['auth']['username'] =
utf8_encode('username');
$requestHeader['auth']['password'] =
utf8_encode('password');

$requestHeader['language'] = utf8_encode('NL');
$requestHeader['reference'] = utf8_encode('My Reference');
$requestHeader['test'] = 1;

```

```

$csr = "-----BEGIN CERTIFICATE REQUEST-----
MIIB5jCCAU8CAQAwwgaUxCzAJBgNVBAYTAK5MMRYwFAYDVQQIEw1Ob29yZCBC
cmFi
YW50MRMwEQYDVQQHEwpsb29zZW5kYW5kYWFsMRwwGgYDVQQKEwNOZXR3b3JraW5n
NGFs
bCBCLlYuMR8wHQYDVQQDEwZ3d3cubmV0d29ya2luZzRhbgwY29tMSowKAYJ
KoZI
hvcNAQkBFhtzZWN1cm10eUBuZXR3b3JraW5nNGFsbc5jb20wgZ8wDQYJKoZI
hvcN
AQEBBQADgY0AMIGJAoGBAOsi6VgMka2KLata8fn0zeXbwbmjcTKd6V973Jrm
4mCk
PAjFklWygDxirtHV+Etbuu7/WAEgwCVJK46tjMK+ZPpmt1f/kXxIUgjtZTa
TS+r
Cfzoyl85MZSKXNhIMOICvkn7m4YXYQDMXLearR+XVa2pnJ0MwpEnAeh8YEsp
s/Pp
AgMBAAAGgADANBgkqhkiG9w0BAQQFAAQBQC3lC2dW6smhPD3SFXy0ypw9f7x
nieL
cosYjN00/yos9NlflGsV0LuZhfgR7yZhUvLyWqM6lhZpL6f7jTUD9IStvAhT
UpLq
LY8ZNI0p00uQDljmqoQnTk1tro2dblzDTWmh1Om2BobW+aU+LnLS9oDJ8gBP
KE1V
J4beS3a2eNwORg==
-----END CERTIFICATE REQUEST-----";

$params = array();
$params['requestHeader'] = $requestHeader;

$params['request']['companyHandle'] = utf8_encode('TST-
000001');
$params['request']['adminHandle'] = utf8_encode('TST-
000002');
$params['request']['techHandle'] = utf8_encode('TST-
000002');
$params['request']['productCode'] = 2005;
$params['request']['csr'] = utf8_encode($csr);
$params['request']['period'] = 1;
$params['request']['option']['approverMethod'] =
utf8_encode("FILE");

$response = $client->orderSSLCertificate($params);

?>

```

After the function has been processed, the API will return a response containing a traceNumber: **1000000001**. This traceNumber is used in the functions resendEmailSSLCertificate, replaceSSLCertificate, and getSSLCertificateOrder.

The response also contains a fileName and fileContent. These must be created and copied to the public root folder of your website.

6.4.7 Resend or edit the approver email address

In order to edit the approver email address, you will require the traceNumber that was given to you in the response after you placed your order. Editing your approver email address is an optional function. In this example, we will be changing it from hostmaster@networking4all.com to webmaster@networking4all.com.

```

<?php
error_reporting(E_ALL);

```

```

ini_set("soap.wsdl_cache_enabled", "0");
use_soap_error_handler(true);

$url = 'https://www.networking4all.com/api/soap/api.wsdl';

$client = new SoapClient($url, array('trace' => 1));

$requestHeader = array();
$requestHeader['auth']['handle'] = utf8_encode('TST-000000');
$requestHeader['auth']['username'] = utf8_encode('username');
$requestHeader['auth']['password'] = utf8_encode('password');

$requestHeader['language'] = utf8_encode('NL');
$requestHeader['reference'] = utf8_encode('My Reference');
$requestHeader['test'] = 1;

$params = array();

$params['requestHeader'] = $requestHeader;
$params['request']['traceNumber'] = 10000000001;
$params['request']['approverEmail'] = utf8_encode('webmaster@networking4all.com');

$response = $client->resendEmailSSLCertificate($params);

?>

```

6.4.8 Replacing an SSL certificate

In this example, we will show you how an SSL certificate can be replaced. This function requires the traceNumber that was given to you in the response after you placed your order.

```

<?php
error_reporting(E_ALL);

ini_set("soap.wsdl_cache_enabled", "0");
use_soap_error_handler(true);

$url = 'https://www.networking4all.com/api/soap/api.wsdl';

$client = new SoapClient($url, array('trace' => 1));

$requestHeader = array();
$requestHeader['auth']['handle'] = utf8_encode('TST-000000');
$requestHeader['auth']['username'] = utf8_encode('username');
$requestHeader['auth']['password'] = utf8_encode('password');

$requestHeader['language'] = utf8_encode('NL');
$requestHeader['reference'] = utf8_encode('My Reference');
$requestHeader['test'] = 1;

$csr = "-----BEGIN CERTIFICATE REQUEST-----
MIIB5jCCAU8CAQAwwgaUxCzAJBgNVBAYTAK5MMRYwFAYDVQQIEw1Ob29yZCBC
cmFi
YW50MRMwEQYDVQQHEwpSb29zZW5kYW5kYWFsMRwwGgYDVQQKEwNOZXRR3b3JraW5n
NGFs

```

```

bCBCLlYuMR8wHQYDVQQDExZ3d3cubmV0d29ya2luZzRhGwuY29tMSowKAYJ
KoZI
hvcNAQkBFhtzZWN1cm10eUBuZXR3b3JraW5nNGFsbC5jb20wgZ8wDQYJKoZI
hvcN
AQEBBQADgY0AMIGJAoGBAOsi6VgMka2KLata8fn0zeXbwbmjcTKd6V973Jrm
4mCk
PAjFklWyGdXirtHV+Etbuu7/WAEgwCVJK46tjMK+ZPpmt1f/kXxIUgjtZTa
TS+r
Cfzoyl85MZSKXNhIMOICvkn7m4YXyQDMXLearR+XVa2pnJ0MwpEnAeh8YEsp
s/Pp
AgMBAAGgADANBgkqhkiG9w0BAQQFAAOBgQC3lC2dW6smhPD3SFXy0ypw9f7x
nieL
cosYjN00/yos9N1f1GsV0LuZhfgR7yZhUvLyWqM6lhZpL6f7jTUD9IstvAhT
UpLq
LY8ZNI0p00uQDljmqoQnTkltro2dblzDTWmhlOm2BobW+aU+LnLS9oDJ8gBP
KE1V
J4beS3a2eNwORg==
-----END CERTIFICATE REQUEST-----";

$params = array();
$params['requestHeader'] = $requestHeader;

$params['request']['traceNumber'] = 10000000001;
$params['request']['csr'] = utf8_encode($csr);
$params['request']['approverEmail'] =
utf8_encode('webmaster@networking4all.com');

$response = $client->replaceSSLCertificate($params);

?>

```

6.4.9 ordering a SAN certificate

In this example, we will be ordering a SAN certificate. The product used in this example is the Networking4all Business SSL OV Multidomain certificate, product code 2050.

```

<?php

error_reporting(E_ALL);

ini_set("soap.wsdl_cache_enabled", "0");
use_soap_error_handler(true);

$url = 'https://www.networking4all.com/api/soap/api.wsdl';

$client = new SoapClient($url, array('trace' => 1));

$requestHeader = array();
$requestHeader['auth']['handle'] = utf8_encode('TST-
000000');
$requestHeader['auth']['username'] =
utf8_encode('username');
$requestHeader['auth']['password'] =
utf8_encode('password');

$requestHeader['language'] = utf8_encode('NL');
$requestHeader['reference'] = utf8_encode('My Reference');
$requestHeader['test'] = 1;

$csr = "-----BEGIN CERTIFICATE REQUEST-----
MIIB5jCCAU8CAQAwgaUxCzAJBgNVBAYTAK5MMRYwFAYDVQQIEw10b29yZCBC
cmFi
YW50MRMwEQYDVQQHEwpSb29zZW5kYW5kYWFsMRwwGgYDVQQKEwNOZXR3b3JraW5n
NGFs

```

```

bCBCLlYuMR8wHQYDVQDExZ3d3cubmV0d29ya2luZzRhGwuY29tMSowKAYJ
KoZI
hvcNAQkBFhtzZWN1cm10eUBuZXR3b3JraW5nNGFsbC5jb20wgZ8wDQYJKoZI
hvcN
AQEBBQADgY0AMIGJAoGBAOsi6VgMka2KLata8fn0zeXbwbmjcTKd6V973Jrm
4mCk
PAjFklWyGdXirtHV+Etbuu7/WAEgwCVJK46tjMK+ZPpmt1f/kXxIUgjtZTa
TS+r
Cfzoyl85MZSKXNhIMOICvkn7m4YXyQDMXLearR+XVa2pnJ0MwpEnAeh8YEsp
s/Pp
AgMBAAGgADANBgkqhkiG9w0BAQQFAAOBgQC3lC2dW6smhPD3SFXy0ypw9f7x
nieL
cosYjn00/yos9N1f1GsV0LuZhfgr7yZhUvLyWqM6lhZpL6f7jTUD9IstvAhT
UpLq
LY8ZNI0p00uQDljmqoQnTk1tro2dblzDTWmhlOm2BobW+aU+LnLS9oDJ8gBP
KE1V
J4beS3a2eNwORg==
-----END CERTIFICATE REQUEST-----";

$params = array();
$params['requestHeader'] = $requestHeader;
$params['request']['productCode'] = 2050;
$params['request']['csr'] = utf8_encode($csr);
$params['request']['san'] =
array(utf8_encode('test1.networking4all.com'),
utf8_encode('test2.networking4all.com'),
utf8_encode('test3.networking4all.com'));

$response = $client->checkSSLCertificateCSR($params);

?>

```

6.4.10 Retrieving the approver list for a SAN certificate

The following function retrieves the approver list for a SAN certificate.

```

<?php
error_reporting(E_ALL);

ini_set("soap.wsdl_cache_enabled", "0");
use_soap_error_handler(true);

$url = 'https://www.networking4all.com/api/soap/api.wsdl';

$client = new SoapClient($url, array('trace' => 1));

$requestHeader = array();
$requestHeader['auth']['handle'] = utf8_encode('TST-
00000');
$requestHeader['auth']['username'] =
utf8_encode('username');
$requestHeader['auth']['password'] =
utf8_encode('password');

$requestHeader['language'] = utf8_encode('NL');
$requestHeader['reference'] = utf8_encode('My Reference');
$requestHeader['test'] = 1;

$params = array();
$params['requestHeader'] = $requestHeader;
$params['request']['productCode'] = 2050;

```

```

$params['request']['domain'] =
utf8_encode('www.networking4all.com');

$response = $client->getSSLCertificateApproverList($params);

?>

```

6.4.11 ordering a SAN certificate

For the SAN certificate order, we will be using the following data:

Company:	TST-000001
Admin Contact:	TST-000002
Tech Contact:	TST-000002
ApproverEmail:	hostmaster@networking4all.com

```

<?php

error_reporting(E_ALL);

ini_set("soap.wsdl_cache_enabled", "0");
use_soap_error_handler(true);

$url = 'https://www.networking4all.com/api/soap/api.wsdl';

$client = new SoapClient($url, array('trace' => 1));

$requestHeader = array();
$requestHeader['auth']['handle'] = utf8_encode('TST-000000');
$requestHeader['auth']['username'] =
utf8_encode('username');
$requestHeader['auth']['password'] =
utf8_encode('password');

$requestHeader['language'] = utf8_encode('NL');
$requestHeader['reference'] = utf8_encode('My Reference');
$requestHeader['test'] = 1;

$csr = "-----BEGIN CERTIFICATE REQUEST-----
MIIB5jCCAU8CAQAwwgaUxCzAJBgNVBAYTAk5MMRYwFAYDVQQIEw1Ob29yZCBC
cmFi
YW50MRMwEQYDVQQHEWpSb29zZW5kYW5kYWFsMRwwGgYDVQQKEwNOZXR3b3JraW5n
NGFs
bCBCLlYuMR8wHQYDVQQDEwZ3d3cubmV0d29ya2luZzRhbgwY29tMSowKAYJ
KoZI
hvcNAQkBFhtzZWN1cm10eUBuZXR3b3JraW5nNGFsbc5jb20wgZ8wDQYJKoZI
hvcN
AQEBBQADgY0AMIGJAoGBAOsi6VgMka2KLata8fn0zeXbwbmjcTKd6V973Jrm
4mCk
PAjFklWyGdXirtHV+Etbui7/WAEgwCVJK46tjMK+ZPpmt1f/kXxIUgjtZTa
TS+r
Cfzoyl85MZSKXNhIMOICvkn7m4YXyQDMXLearR+XVa2pnJ0MwpEnAeh8YEsp
s/Pp
AgMBAAGgADANBgkqhkiG9w0BAQQFAAOBgQC3lC2dW6smhPD3SFXy0ypw9f7x
nieL
cosYjN00/yos9N1f1GsV0LuZhfgR7yZhUvLyWqM6lhzpL6f7jTUD9ISTvAhT
UpLq
LY8ZNI0p00uQDljqmoQnTk1tro2dblzDTWmhlOm2BobW+aU+LnLS9oDJ8gBP
KE1V
J4beS3a2eNwORg==
-----END CERTIFICATE REQUEST-----";

```

```

$params = array();
$params['requestHeader'] = $requestHeader;

$params['request']['companyHandle'] = utf8_encode('TST-
000001');
$params['request']['adminHandle'] = utf8_encode('TST-
000002');
$params['request']['techHandle'] = utf8_encode('TST-
000002');
$params['request']['productCode'] = 2050;
$params['request']['csr'] =utf8_encode($csr);
$params['request']['san'] =
array(utf8_encode('test1.networking4all.com'),

utf8_encode('test2.networking4all.com'),

utf8_encode('test3.networking4all.com'));

$params['request']['period'] = 1;
$params['request']['approverEmail'] =
utf8_encode('hostmaster@networking4all.com');

$response = $client->orderSSLCertificate($params);

?>

```

After the function has been processed, the API will return a response containing a traceNumber: **1000000001**. This traceNumber is used in the functions resendEmailSSLCertificate, replaceSSLCertificate, and getSSLCertificateOrder.

6.4.12 Checking the status of the order and the certificate

With this function, you can check the status of your order through the API. If the status code is 216 ('Certificate has been issued'), you can request the certificate from the API. This function requires the traceNumber that was given to you in the response after you placed your order.

```

<?php
error_reporting(E_ALL);

ini_set("soap.wsdl_cache_enabled", "0");
use_soap_error_handler(true);

$url = 'https://www.networking4all.com/api/soap/api.wsdl';

$client = new SoapClient($url, array('trace' => 1));

$requestHeader = array();
$requestHeader['auth']['handle'] = utf8_encode('TST-
000000');
$requestHeader['auth']['username'] =
utf8_encode('username');
$requestHeader['auth']['password'] =
utf8_encode('password');

$requestHeader['language'] = utf8_encode('NL');
$requestHeader['reference'] = utf8_encode('My Reference');
$requestHeader['test'] = 1;

$params = array();
$params['requestHeader'] = $requestHeader;

```



```

$params['request']['traceNumber'] = 10000000001;

$response = $client->getSSLCertificate($params);

?>

```

6.4.13 Retrieving list of SSL certificates through the API

With this function, the API can generate a list of SSL certificate orders, with their appropriate order status and, if available, the certificate. For this function, you will need to include a search parameter in order to limit the results. These parameters can be a date range, the status, or a handle. In this example, we use the date:

```

<?php

error_reporting(E_ALL);

ini_set("soap.wsdl_cache_enabled", "0");
use_soap_error_handler(true);

$url = 'https://www.networking4all.com/api/soap/api.wsdl';

$client = new SoapClient($url, array('trace' => 1));

$requestHeader = array();
$requestHeader['auth']['handle'] = utf8_encode('TST-000000');
$requestHeader['auth']['username'] = utf8_encode('username');
$requestHeader['auth']['password'] = utf8_encode('password');

$requestHeader['language'] = utf8_encode('NL');
$requestHeader['reference'] = utf8_encode('My Reference');
$requestHeader['test'] = 1;

$params = array();
$params['requestHeader'] = $requestHeader;
$params['request']['fromDate'] = '2010-01-01';
$params['request']['toDate'] = '2010-01-13';

$response = $client->getSSLCertificateList($params);

?>

```

6.4.14 Change your malware scan-options

With this function, you can change the options of your malware scan.

```

<?php

error_reporting(E_ALL);

ini_set("soap.wsdl_cache_enabled", "0");
use_soap_error_handler(true);

$url = 'https://www.networking4all.com/api/soap/api.wsdl';

$client = new SoapClient($url, array('trace' => 1));

$requestHeader = array();

```

```

$requestHeader['auth']['handle'] = utf8_encode('TST-
000000');
$requestHeader['auth']['username'] =
utf8_encode('username');
$requestHeader['auth']['password'] =
utf8_encode('password');

$requestHeader['language'] = utf8_encode('NL');
$requestHeader['reference'] = utf8_encode('My Reference');
$requestHeader['test'] = 1;

$params = array();
$params['requestHeader'] = $requestHeader;
$params['request']['traceNumber'] = 10000000001;
$params['request']['action'] =
utf8_encode('UPDATE_EXTRA_PREFERENCES');
$params['request']['extraPreferences'] =
utf8_encode('MS_DISABLED');

$response = $client->updateSSLCertificate($params);

?>

```

6.4.15 Request an on-demand scan

This function allows you to request an on-demand scan.

```

<?php

error_reporting(E_ALL);

ini_set("soap.wsdl_cache_enabled", "0");
use_soap_error_handler(true);

$url = 'https://www.networking4all.com/api/soap/api.wsdl';

$client = new SoapClient($url, array('trace' => 1));

$requestHeader = array();
$requestHeader['auth']['handle'] = utf8_encode('TST-
000000');
$requestHeader['auth']['username'] =
utf8_encode('username');
$requestHeader['auth']['password'] =
utf8_encode('password');

$requestHeader['language'] = utf8_encode('NL');
$requestHeader['reference'] = utf8_encode('My Reference');
$requestHeader['test'] = 1;

$params = array();
$params['requestHeader'] = $requestHeader;
$params['request']['traceNumber'] = 10000000001;
$params['request']['action'] =
utf8_encode('REQUEST_ON_DEMAND_SCAN');

$response = $client->updateSSLCertificate($params);

?>

```

6.4.16 Add an additional SAN-name to your certificate

This function allows you to add additional SAN-names, or remove SAN-names from your SAN certificate. Please note: when using this function, it is imperative that you include all the SAN-names you wish to keep in the certificate, as well as the names you want to add.

```
<?php
error_reporting(E_ALL);

ini_set("soap.wsdl_cache_enabled", "0");
use_soap_error_handler(true);

$url = 'https://www.networking4all.com/api/soap/api.wsdl';

$client = new SoapClient($url, array('trace' => 1));

$requestHeader = array();
$requestHeader['auth']['handle'] = utf8_encode('TST-000000');
$requestHeader['auth']['username'] = utf8_encode('username');
$requestHeader['auth']['password'] = utf8_encode('password');

$requestHeader['language'] = utf8_encode('NL');
$requestHeader['reference'] = utf8_encode('My Reference');
$requestHeader['test'] = 1;

$params = array();
$params['requestHeader'] = $requestHeader;

$params['request']['traceNumber'] = utf8_encode(10000000001);

// Add test4.networking4all.com
$params['request']['san'] = array(utf8_encode('test1.networking4all.com'),
utf8_encode('test2.networking4all.com'),
utf8_encode('test3.networking4all.com'),
utf8_encode('test4.networking4all.com'));

$response = $client->changeSANSSLCertificate($params);

?>
```

6.4.17 Order a GlobalSign PersonalSign certificate

For the certificate order, we will be using the following data:

Company:	TST-000001
Admin Contact:	TST-000002
Tech Contact:	TST-000002

```
<?php
error_reporting(E_ALL);

ini_set("soap.wsdl_cache_enabled", "0");
```

```

use_soap_error_handler(true);

$url = 'https://www.networking4all.com/api/soap/api.wsdl';

$client = new SoapClient($url, array('trace' => 1));

$requestHeader = array();
$requestHeader['auth']['handle'] = utf8_encode('TST-000000');
$requestHeader['auth']['username'] = utf8_encode('username');
$requestHeader['auth']['password'] = utf8_encode('password');

$requestHeader['language'] = utf8_encode('NL');
$requestHeader['reference'] = utf8_encode('My Reference');
$requestHeader['test'] = 1;

$params['request']['companyHandle'] = utf8_encode('TST-000001');
$params['request']['adminHandle'] = utf8_encode('TST-000002');
$params['request']['techHandle'] = utf8_encode('TST-000002');
$params['request']['productCode'] = 1844;
$params['request']['period'] = 1;

/*
 * Fields for GlobalSign PersonalSign (PS2Pro example)
 */
$params['request']['commonName'] = utf8_encode('Test Person');
$params['request']['title'] = utf8_encode('Dhr');
$params['request']['locality'] = utf8_encode('Amsterdam');
$params['request']['stateOrProvince'] = utf8_encode('Noord-Holland');
$params['request']['country'] = utf8_encode('NL');
$params['request']['email'] = utf8_encode('test@testcompany.com');

$response = $client->orderSSLCertificate($params);

?>

```

6.4.18 Ordering a GlobalSign CodeSigning certificate

For the certificate order, we will be using the following data:

Company:	TST-000001
Admin Contact:	TST-000002
Tech Contact:	TST-000002

```

<?php
error_reporting(E_ALL);

ini_set("soap.wsdl_cache_enabled", "0");
use_soap_error_handler(true);

$url = 'https://www.networking4all.com/api/soap/api.wsdl';

$client = new SoapClient($url, array('trace' => 1));

```

```

$requestHeader = array();
$requestHeader['auth']['handle'] = utf8_encode('TST-
000000');
$requestHeader['auth']['username'] =
utf8_encode('username');
$requestHeader['auth']['password'] =
utf8_encode('password');

$requestHeader['language'] = utf8_encode('NL');
$requestHeader['reference'] = utf8_encode('My Reference');
$requestHeader['test'] = 1;

$params['request']['companyHandle'] = utf8_encode('TST-
000001');
$params['request']['adminHandle'] = utf8_encode('TST-
000002');
$params['request']['techHandle'] = utf8_encode('TST-
000002');
$params['request']['productCode'] = 1850;
$params['request']['period'] = 1;

/*
 * Fields for GlobalSign CodeSigning (CodeSigning
Organisation example)
 */
$params['request']['certificateType'] =
utf8_encode('OSJAVA');
$params['request']['commonName'] = utf8_encode('Test company
B.V. ');
$params['request']['organizationUnit'] =
utf8_encode('Test'); // Optional
$params['request']['locality'] = utf8_encode('Amsterdam');
$params['request']['stateOrProvince'] = utf8_encode('Noord-
Holland');
$params['request']['country'] = utf8_encode('NL');
$params['request']['email'] =
utf8_encode('test@testcompany.com');

$response = $client->orderSSLCertificate($params);

?>

```

6.5 UTF-8 encoding and decoding

The API is based on SOAP and requires valid UTF-8 encoded character ranges in the request. Only character ranges containing special characters require encoding, but for the sake of completeness it is considered best practice to encode all the information that is included in the request with UTF-8. This can be done with the `utf8_encode()`-function in PHP, or a similar function in other programming languages.

All responses from the API are encoded by default. It is considered best practice to first decode the responses prior to using them. This can be done with the `utf8_decode()`-function in PHP, or a similar function in other programming languages.

7. Product information

7.1 Status codes

7.1.1 Status codes `getSSLCertificateOrder`

The following status codes can be found in the response for the function `getSSLCertificateOrder`:

Code	Description
214	SSL request not approved
215	SSL certificate requested
216	SSL certificate is issued
218	SSL certificate replacement requested
221	SSL certificate expired
222	SSL certificate almost expired
225	SSL certificate repayed

7.1.2 Status codes `getSSLCertificate` and `getSSLCertificateList`

The following status codes can be found in the response for the functions `getSSLCertificate` and `getSSLCertificateList`:

Code	Description
99	message
100	Approver email
101	Approver email not yet approved
102	Approver email approved
109	CSR details do not match company data in Chamber of Commerce registry
110	Check Chamber of Commerce data
111	Company data not approved, require more data
112	Company data approved
210	Check domain name owner in DNS
211	Owner data not approved, require more data
212	Owner data approved

310	Check admin contact
311	Admin contact not approved, require more data
312	Admin contact approved
410	Contract CA sent to vetting@networking4all.com
411	Contract not valid, requires modification
412	Contract approved
510	Validation by phone
511	Check phone number in public source
512	Validation by phone with HR successful
513	Validation by phone with HR not successful
514	Validation by phone with management successful
515	Validation by phone with management not successful
516	Validation by phone with admin contact successful
517	Validation by phone with admin contact not successful
1000	Review order
1001	Review order in process
1002	Certificate will be issued
1010	Order cancelled by client
1015	Order cancelled by Certificate Authority

8. The API control panel

Our customers can access the API control panel on our website, www.networking4all.com. This panel shows an overview of all the requests you have sent to the API, with their appropriate responses. The API control panel can be found in the API tab in your account.

8.1 Overview

The overview offers the following functionalities:

- Your requests are sorted by date by default, with the newest requests at the top of the list. You can also choose to sort your requests by other criteria, such as function, handle, username, host, result, and time. You can choose these options by clicking on the column name in the table.
- Requests containing an error code will be shown in red (resultCode 2). Requests that have executed properly, but did not return any results are shown in blue (resultCode 1).
- You can navigate to the relation connected to a request by clicking on the handle or the username.
- The time shown in the overview is the time in seconds that was required to perform the request. Most requests take less than a second to perform. However, there are functions that require more time, such as `checkSSLCertificateCSR`, `getSSLCertificateApproverList`, `orderSSLCertificate`, `resendEmailSSLCertificate`, and `replaceSSLCertificate`.

By clicking on the date or the magnifying glass on the right side of the table, you will be transferred to the detail page.

8.2 Detail page

The detail page shows all the details of your request, with the appropriate response from the API. The request and response are both shown in an XML structure.

For security reasons, we do not show passwords in the request and the response.

For more information, please visit our support page:
<https://www.networking4all.com/en/support/>