
TelHosting Client SOAP API Reference

March 2010



Disclaimer

Copyright (c) 2008-2010, Telnic Ltd.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the Telnic Ltd nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS DOCUMENTATION IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Table of Contents

About this Document	5
Revision History	5
Overview	6
Overview.....	6
Key Concepts	7
Object Model.....	10
Message Format.....	12
List of Namespaces	13
List of Schema Files.....	14
Related Links	15
Domains and Zones.....	16
Domains and Zones	16
createZone	17
updateZone.....	19
deleteZone	20
getZone	21
listZones.....	22
createDomain	23
updateDomain.....	24
deleteDomain	26
getDomain	27
listDomains.....	29
Resource Records.....	31
Resource Records	31
storeRecord	32
deleteRecord.....	38
updateRecords	39
listRecords.....	41
Data Privacy	43
Data Privacy.....	43

createGroup	44
createReader	45
deleteGroup	47
deleteReader	48
getGroup	49
getReader	50
listGroups	52
listReaders	53
updateGroup	54
updateReader	55
Profiles	57
Profiles	57
createProfile	58
deleteProfile	59
updateProfile	60
listProfiles	62
listProfilesExt	63
getProfile	64
switchToProfile	65
getActiveProfile	66
Search Data	67
Search Data	67
Allowed Keywords	68
setSearchData	70
getSearchData	72
Data Exchange	74
Data Exchange	74
exportData	75
importData	78

1. ABOUT THIS DOCUMENT

This document describes the TelHosting SOAP API, which can be used for managing and publishing contact information in .tel domain DNS records.

The reference is targeted at developers interested in creating new [.tel client applications](#) and enhancing existing ones. The document assumes that you are familiar with application programming standard SOAP (Simple Object Access Protocol) and that you understand the [key concepts](#) of the .tel innovative technology. See [Related Links](#) for references to relevant materials.

With this reference, you will learn about:

- The objects used in the interface
- The functions available to manipulate all those objects: domains, resource records, and profiles
- Specific usage requirements for each function

1.1. Revision History

Version	Description
---------	-------------

- | | |
|-----|---|
| 1.0 | Client API reference; initial version. |
| 1.1 | Added optional attribute <code>effUserName</code> for all relevant operations; fixed inaccurate descriptions in createProfile , updateProfile , deleteProfile . |
| 1.2 | Added new operation checkZone to Domain management. |
| 1.3 | Added new operations updateDomain to Domain management and restoreDomain to Data Exchange; included Example 4 to add TelAd records in the storeRecord command, and new parameters to perform key operations recursively |
| 1.4 | Updated operations updateDomain and getDomain to show the new colour and other display settings; included Example 5 to storeRecord to show adding a logo |

2. OVERVIEW

2.1. Overview

The .tel top-level domain (TLD) uses the Domain Name System (DNS) as a data store for contact information in text records, location records, and Name Address Pointer records (NAPTR). To support this innovative technology, the sponsoring organization Telnic Ltd. provides a Java-based TelHosting Software application. This application interacts with the DNS directly, and exposes a publicly available web service, using SOAP over HTTP as its protocol. Client applications implementing this SOAP interface can work with the TelHosting Software application to perform all .tel-related operations, namely:

- Configure all types of contact data: phone, mobile, e-mail, VoIP, chat, fax, locations, web links, etc.
- Publish and update contact data in real time
- Create profiles displaying different records depending on user preferences
- Control the distribution of private data to specific groups of people
- Specify keywords by which they want to be found

All APIs work by receiving request messages from the clients and sending response messages back to them unless the request message requires no response. All messages are expressed in XML, and their respective syntax is specified by an XML schema; see [List of Schema Files](#). The messages must follow the specific format, see [Message Format](#).

Groups of API Functions

- [Domains and Zones](#)
- [Resource Records](#)
- [Data Privacy](#)
- [Profiles](#)
- [Search Data](#)
- [Data Exchange](#)

In addition to the client service API, the TelHosting Software exposes an internal API for administration purposes. That API is available only to users with specific permissions (TelHosting Providers) and is not covered in this document.

2.2. Key Concepts

This reference is based on a number of basic concepts, around which the whole API is organized. Most programming concepts are represented by a matching object, see [Object Model](#).

Table 1: Key Concepts

Concept	Description
.tel client application	<p>An application that manipulates contact data published in .tel domain records. The application uses the SOAP APIs to interact with the <i>TelHosting Provider</i> and can query .tel DNS directly to perform its functions.</p> <p>See currently available applications at Telnic Developer Area.</p>
TelHosting Provider	<p>A company accredited to register .tel domain names.</p> <p>The TelHosting provider operates name servers to resolve .tel domains and provides an external interface, by which end users can maintain the contents of their domains. This Client SOAP API is a part of that interface.</p> <p>The reference implementation of the software supporting creation and maintenance of DNS zones and their publication on name servers is <i>TelHosting Software</i>.</p>
TelHosting Software	<p>The open-source software reference implementation that name server operators and registrars can deploy to support .tel domain manipulation.</p> <p>The TelHosting Software is a Java application that supports provision of NAPTR records, private data encryption, search data management, and user profile editing and switching. The software exports a number of <i>TelHosting APIs</i>.</p>
TelHosting APIs	<p>The public SOAP API that allows external applications to manipulate data associated with .tel domains. This way, .tel users can access data stored with their TelHosting Provider not only by using the web interface, but also via client applications stored on their PCs and mobile devices.</p> <p>Structurally, the TelHosting SOAP API is divided into the Client part and the Admin, or Internal part. The Client API is for .tel client applications, and the internal part is for administrative purposes, such as name server and user management. The admin API is out of the scope of this document.</p>
.tel Sponsoring Organization	<p>Telnic Ltd. is the sponsoring organization (SO) for the .tel domain. SO provides a number of services, including infrastructure for TelHosting Providers, .tel member database,</p>

and storage of keys for record encryption.

In relation to .tel client applications, the SO provides a SOAP API to support the "friending" mechanism for private data exchange. That API covers initialization in the SO system, key/credential management, and processing of friending messages. The SO API is out of the scope of this document.

Domain	<p>A logical collection of data associated with a domain name and containing resource records of allowed data types. Domains are associated with user profiles and groups of readers to manage access to individual records stored in this domain.</p> <p>See the Domain object.</p>
Zone	<p>Coordinates publication of domains. See the Zone object.</p>
Record	<p>A data item to be published as part of a .tel domain. Allowed record types are NAPTR, TXT, LOC (location), MX (mail exchange), NINFO (information), and SRV (server location).</p> <p>NAPTR (Naming Authority Pointer) records are powerful resource records which allow for the provisioning of additional information in a very flexible way. For more details on NAPTRs, please refer to the whitepaper "NAPTRs in .tel".</p>
Search Data	<p>Keyword - value pairs for inclusion in the search index of the sponsoring organization. Users can then search for .tel domains by specified keywords to enable categorized searching.</p>
Profile	<p>An assembly of records created by the user for easy publication as a single set. Multiple profiles allow the user to switch published data depending on personal circumstances, such as business hours, vacation, trips, etc. See the Profile object.</p>
User	<p>A user of the system with a certain level of access: administrator or primary user. Users are created by the TelHosting Provider and cannot be created by means of this API. Administrators can act on behalf of other users by specifying the user name in the effUserName attribute of a request, although some operations do not allow this.</p> <p>Do not mix users with Readers - external users allowed to read private data; see the Reader object.</p>
effUserName	<p>The effective user name.</p> <p>Optional attribute that can be used by administrators to act on behalf of other users. It is explicitly noted if the attribute is not allowed in an operation. If a user with no administrator rights specifies the <code>effUserName</code> attribute, its value must match the name of the calling user.</p>

Group	An assembly of readers who can read the same collection of private data. Records and readers get associated with groups. See the Group object.
Permissions	<p>User authorization defined by the TelHosting Provider's policy to regulate creation of second- and greater-level domains and zones, as well as their delegation to users. Primary users can even create separate zones for individual third- and greater-level domains, so that they appear separately in the DNS, possibly, on a different set of name servers.</p> <p>The policies may prohibit creation of new domains and users, and request the primary user to place a request by other means, such as online registration.</p>

2.3. Object Model

The TelHosting Software is built around a set of abstract objects. The majority of objects are used to represent the user's data, while the others are for administrative purposes only and, as such, are only accessible via the administrative interfaces.

Table 2: Client SOAP API Objects

Object Name	Description
Domain	<p>The logical collection of data associated with a domain name. It consists of the records, the search data, and different views, which result from the association with Profiles on the one side and with Readers and Groups on the other side. The Domain is not limited to second-level domains; deeper levels may be managed as well.</p> <p>Note: the Domain object itself does not control on which name servers the domain is published. This is done by the Zone object.</p>
Zone	<p>Coordinates publication of domains by referring to the Name Server objects of the name servers that shall publish the zone. The Zone object has a domain name associated with it. It automatically binds all Domain objects that lie within the zone and are not covered by any other subordinated zones. There is a often one-to-one relationship between Domains and Zones, i.e. for each Domain object, there is a corresponding Zone object with the same name. But this is not a requirement. For simplicity, the web interface offers the creation/deletion of the respective zone when a domain is created or deleted.</p>
Record	<p>Represents a piece of information that shall be published. It more or less correlates to the records defined in the DNS protocol. Records are associated with Profile objects and Group objects, which has influence on when and how (encrypted or publicly readable) the record is published. Records may have owner names, which are then relative to the name of the Domain object to which they are linked.</p>
Profile	<p>Allows the user to group records and publish them by selecting a profile for publishing. There is always a default profile that cannot be deleted.</p> <p>Note that throughout this API, existing profiles are addressed by their ids, not by their names.</p>
Reader	<p>Represents an external user allowed to read private data. The object is created during the so-called friending process. The Reader object itself does not grant access to the private data. Instead, the Group object links the reader to the data.</p>

Group	Defines user groups allowed to read private data, e.g., “family”, “friends”, “colleagues”, or “business partners”. Records and Readers may belong to zero, one or more Groups. Records that are associated with at least one group are no longer publicly readable.
User	Represents a user of the system. Depending on the level of access, a user can be an administrator or a primary user. Primary users are typical users of the system who can create domains and zones.

2.4. Message Format

Request

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
  xmlns:typ="_xmlns.telnic.org_namespace_here_">
  <soap:Header/>
  <soap:Body>
  ...
  </soap:Body>
</soap:Envelope>
```

Instead of `_xmlns.telnic.org_namespace_here_`, insert the corresponding namespace from the [List of Namespaces](#).

Response

```
<S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope">
  <S:Body>
  ...
  </S:Body>
</S:Envelope>
```

2.5. List of Namespaces

Name	API Function Group
http://xmlns.telnic.org/ws/nsp/client/domain/types-1.0	Domain and Zone Management
http://xmlns.telnic.org/ws/nsp/client/record/types-1.0	Resource Record Management
http://xmlns.telnic.org/ws/nsp/client/reader/types-1.0	Data Privacy Management
http://xmlns.telnic.org/ws/nsp/client/profile/types-1.0	Profile Management
http://xmlns.telnic.org/ws/nsp/client/searchdata/types-1.0	Search Data Management
http://xmlns.telnic.org/ws/nsp/client/exchange/types-1.0	Data Exchange Management

2.6. List of Schema Files

The various schema files can be found inside the distribution of TelHosting Software or with this reference. One schema matches one API function group. The [Client-Service-1.0.wsdl](#) file represents the whole client interface, and the types file [Client-1.0.wsdl](#).

Name	API Function Group
Domain-1.0.xsd	Domain and Zone Management
Record-1.0.xsd	Resource Record Management
Reader-1.0.xsd	Data Privacy Management
Profile-1.0.xsd	Profile Management
SearchData-1.0.xsd	Search Data Management
Exchange-1.0.xsd	Data Exchange Management
xml/Exchange-1.0.xsd	Export/Import Data Format

2.7. Related Links

- Telnic Developer Area, <http://dev.telnic.org/index.html>
- Telnic Developers FAQ, <http://dev.telnic.org/pages/faq.html>
- Telnic Developers Guide, <http://dev.telnic.org/pages/howtos.html>
- TelHosting Admin SOAP API, <http://dev.telnic.org/api/admin-soap/index.html>
- SO SOAP API, <http://dev.telnic.org/api/so-soap/index.html>
- SOAP, Simple Object Access Protocol, <http://www.w3.org/TR/soap/>
- Telnic Whitepaper "NAPTRs in .tel", <http://dev.telnic.org/docs/naptr.pdf>

3. DOMAINS AND ZONES

3.1. Domains and Zones

This API provides a means of maintaining zone and domain objects, see the [Domain-1.0.xsd](#) schema. Depending on permissions, [users](#) may or may not create, update, and delete these objects, see Table 3 for details.

Table 3: Summary of domain and zone management API

Operation	Admin User		Description
createZone	yes	permissions	Creates a new zone (Zone object), with automatic and/or manual name server assignment.
updateZone	yes	permissions	Deletes a zone.
deleteZone	yes	permissions	Deletes a Zone object.
listZones	yes	yes	List the names of all zones.
checkZone	yes	permissions	Check whether the zone already exists.
getZone	yes	yes	Retrieves data for a zones.
createDomain	yes	permissions	Creates a Domain object.
updateDomain	yes	permissions	Renames a sub-domain and/or adds a display string.
deleteDomain	yes	permissions	Deletes a Domain object.
getDomain	yes	yes	Retrieves domain information.
listDomains	yes	yes	Lists the names of all domains.

Please refer to Data Exchange section for the [restoreDomain](#) operation reference.

3.2. createZone

This operation creates a new zone object with a given domain name.

You can let the system assign name servers for the new zone or specify the name servers from the list of servers managed by TelHosting Software. With automatic name server assignment, the system can distribute the zones depending on the capability and load of the available servers. The number of servers assigned to a zone varies between 2 and 13; the maximum number available for you depends on your permissions. If you have defined fewer servers than the specified number, the system will select the remaining servers.

Set the optional `synchronized` parameter to `true` to ensure that the operation is valid, especially when performing a domain transfer. With this parameter, the system does the following:

- checks that the domain for this new zone belongs to the correct registrar by comparing the IANA ID in the WHOIS database and the Partition for the user creating the zone
- finds any existing zones for this domain across all partitions and deletes any zones that belong to other users and may block the creation of the new zone

createZoneRequest

Attribute Name	Description
<code>zoneName</code>	Name of the zone to be created
<code>effUserName</code>	<i>optional</i> name of the user performing the operation; used by administrators acting on behalf of other users. If a user with no administrator rights specifies the <code>effUserName</code> attribute, its value must match the name of the calling user.
<code>synchronized</code>	<i>optional</i> Flag to verify that the zone is created for the domain in the correct partition and that no zones exist in other partition to prevent creation of this zone

Element Name	Description
<code>nameservers</code>	Name servers to be associated with the new zone. Attributes: <ul style="list-style-type: none"> • <code>count</code> - <i>optional</i> Number of name servers to use for the zone Subelements: <ul style="list-style-type: none"> • <code>host</code> - Domain name of the host

createZoneResponse

No elements.

Example

March 2010

Create a zone for reggie.tel with three name servers. Let the system select the third name server.

Request

```
<typ:createZoneRequest zoneName="reg.reggie.tel" synchronized="true">
  <typ:nameservers count="3">
    <typ:host>ns1.example.com</typ:host>
    <typ:host>ns2.example.com</typ:host>
  </typ:nameservers>
</typ:createZoneRequest>
```

Response

```
<createZoneResponse
  xmlns="http://xmlns.telnic.org/ws/nsp/client/record/types-1.0"/>
```

3.3. updateZone

This operation updates the information for a given zone object.

updateZoneRequest

Attribute Name	Description
zoneName	Name of the zone to be updated.
effUserName	<i>optional</i> name of the user performing the operation; used by administrators acting on behalf of other users. If a user with no administrator rights specifies the <code>effUserName</code> attribute, its value must match the name of the calling user.

Element Name	Description
nameservers	Name servers to be associated with the zone. Attributes: <ul style="list-style-type: none"> <code>count</code> - <i>optional</i> Number of name servers to use for the zone Subelements: <ul style="list-style-type: none"> <code>host</code> - Domain name of the host

updateZoneResponse

No elements.

Example

Update the zone of reggie.tel to use only ns3 and ns4.

Request

```
<typ:updateZoneRequest zoneName="reg.reggie.tel">
  <typ:nameservers>
    <typ:host>ns3.example.com</typ:host>
    <typ:host>ns4.example.com</typ:host>
  </typ:nameservers>
</typ:updateZoneRequest>
```

Response

```
<updateZoneResponse
xmlns="http://xmlns.telnic.org/ws/nsp/client/record/types-1.0"/>
```

3.4. deleteZone

This operation deletes a zone object.

deleteZoneRequest

No elements.

Attribute Name	Description
zoneName	Name of the zone to be deleted.
effUserName	<i>optional</i> name of the user performing the operation; used by administrators acting on behalf of other users. If a user with no administrator rights specifies the <code>effUserName</code> attribute, its value must match the name of the calling user.

deleteZoneResponse

No elements.

Example

Delete the zone `reg.reggie.tel`.

Request

```
<typ:deleteZoneRequest zoneName="reg.reggie.tel"/>
```

Response

```
<deleteZoneResponse  
xmlns="http://xmlns.telnic.org/ws/nsp/client/domain/types-1.0"/>
```

3.5. getZone

This operation retrieves the data associated with the zone, i.e. the name servers.

getZoneRequest

No elements.

Attribute Name	Description
zoneName	Name of the zone to get.
effUserName	<i>optional</i> name of the user performing the operation; used by administrators acting on behalf of other users. If a user with no administrator rights specifies the <code>effUserName</code> attribute, its value must match the name of the calling user.

getZoneResponse

Element Name	Description
nameservers	Name servers to be associated with the zone. Attributes: <ul style="list-style-type: none"> <code>count</code> - <i>optional</i> Number of name servers to use for the zone Subelements: <ul style="list-style-type: none"> <code>host</code> - Domain name of the host

Example

Get the zone `reg.reggie.tel`.

Request

```
<typ:getZoneRequest zoneName="reg.reggie.tel"/>
```

Response

```
<ns4:getZoneResponse
xmlns:ns4="http://xmlns.telnic.org/ws/nsp/client/domain/types-1.0">
<ns4:nameservers>
  <ns4:host>ns4.example.com</ns4:host>
  <ns4:host>ns3.example.com</ns4:host>
</ns4:nameservers>
</ns4:getZoneResponse>
```

3.6. listZones

This operation lists all zones associated with the calling user.

listZonesRequest

No elements or attributes.

listZonesResponse

Element Name	Description
zoneName	Name of the zone to be created.
effUserName	<i>optional</i> name of the user performing the operation; used by administrators acting on behalf of other users. If a user with no administrator rights specifies the <code>effUserName</code> attribute, its value must match the name of the calling user.

Example

Request

```
<typ:listZonesRequest/>
```

Response

```
<ns4:listZonesResponse
xmlns:ns4="http://xmlns.telnic.org/ws/nsp/client/domain/types-1.0">
  <ns4:zoneName>reggie.tel</ns4:zoneName>
  <ns4:zoneName>reg.reggie.tel</ns4:zoneName>
  <ns4:zoneName>greg.reggie.tel</ns4:zoneName>
</ns4:listZonesResponse>
```

3.7. createDomain

This operation creates a domain name in the system.

Description

Profiles, resource records, and private data are associated with a domain name, although the [Domain](#) object itself stores only its delegation status.

createDomainRequest

Attribute Name	Description
domainName	Name of the domain to be created
effUserName	<i>optional</i> name of the user performing the operation; used by administrators acting on behalf of other users. If a user with no administrator rights specifies the <code>effUserName</code> attribute, its value must match the name of the calling user.

createDomainResponse

No elements or attributes.

Example

Create a domain.

Request

```
<typ:createDomainRequest domainName="reg-reggie.tel"/>
```

Response

```
<createDomainResponse  
  xmlns="http://xmlns.telnic.org/ws/nsp/client/domain/types-1.0"/>
```

3.8. updateDomain

This operation updates the name of a domain object and/or the domain display string in the system for non-apex domains (lower than 2nd level).

The **domain display string** (DDS) is used as the <title> and <h1> tags when the domain is queried via the web, and can be displayed instead of the domain name elsewhere. If no DDS is specified, then the domain name is shown. To remove the DDS, perform this operation and set the `displayString` element to an empty value. The DDS is stored as a TXT record and has all the attributes of this record type: support for UTF-8 and the length of 255 bytes. You can also manage the DDS using the Resource Records API, but this is not recommended and, therefore, no example of that operation is given.

Use either or both optional elements in the request to perform the update. If neither of the elements is present, the operation has no effect.

When the update is performed to change the domain name in the DNS, all non-terminal NAPTRs in the user's account pointing to the renamed domain are updated automatically, while domain data remains unchanged.

updateDomainRequest

Attribute Name	Description
<code>displayString</code>	<i>optional</i> value of the display string.
<code>newDomainName</code>	<i>optional</i> new name to be used for this domain.
<code>effUserName</code>	<i>optional</i> name of the user performing the operation; used by administrators acting on behalf of other users. If a user with no administrator rights specifies the <code>effUserName</code> attribute, its value must match the name of the calling user.
<code>color1</code>	<i>optional</i> hex code for the colour of the top bar on the web proxy.
<code>color2</code>	<i>optional</i> hex code for the colour of the title bar on the web proxy.
<code>color3</code>	<i>optional</i> hex code for the colour of the background on the web proxy.
<code>css</code>	<i>optional</i> ID for the CSS template to be applied on the web proxy.
<code>pss</code>	<i>optional</i> scope of the search feature; "false" to only show "This .tel" and "true" to allow searching throughout the .tel zone with "All .tels"

updateDomainResponse

No elements or attributes.

Example

Remove the delegation from the domain.

Request

```
<typ:updateDomainRequest domainName="greg.reggie.tel">
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
  xmlns:typ="http://xmlns.telnic.org/ws/nsp/client/domain/types-1.0">
  <soap:Header/>
  <soap:Body>
    <typ:updateDomainRequest domainName="greg.reggie.tel">
      <!--Optional:-->
      <typ:newDomainName>john.reggie.tel</typ:newDomainName>
      <!--Optional:-->
      <typ:displayString>The domain of Greg Reggie</typ:displayString>
      <!--Optional:-->
      <typ:color1>#000000</typ:color1>
      <!--Optional:-->
      <typ:color2>#ffffff</typ:color2>
      <!--Optional:-->
      <typ:color3>#89abcd</typ:color3>
      <!--Optional:-->
      <typ:css>1</typ:css>
      <!--Optional:-->
      <typ:pss>>false</typ:pss>
    </typ:updateDomainRequest>
  </soap:Body>
</soap:Envelope>
</typ:updateDomainRequest>
```

Response

```
<updateDomainResponse
xmlns="http://xmlns.telnic.org/ws/nsp/client/domain/types-1.0"/>
```

3.9. deleteDomain

This operation removes a domain from the system.

deleteDomainRequest

Attribute Name	Description
domainName	Name of the domain to be deleted.
recursive	<i>optional</i> Flag to delete the entire domain tree below the given domainName when set to true
effUserName	<i>optional</i> name of the user performing the operation; used by administrators acting on behalf of other users. If a user with no administrator rights specifies the effUserName attribute, its value must match the name of the calling user.

deleteDomainResponse

No elements or attributes.

Example

Delete the domain reg-reggie.tel.

Request

```
<typ:deleteDomainRequest domainName="reg-reggie.tel"
recursive="true"/>
```

Response

```
<deleteDomainResponse
xmlns="http://xmlns.telnic.org/ws/nsp/client/domain/types-1.0"/>
```

3.10. getDomain

This operation returns the information for a given domain name.

getDomainRequest

Attribute Name	Description
domainName	Name of the domain to get.
effUserName	<i>optional</i> name of the user performing the operation; used by administrators acting on behalf of other users. If a user with no administrator rights specifies the <code>effUserName</code> attribute, its value must match the name of the calling user.

getDomainResponse

Element Name	Description
delegatedTo	<i>not supported</i> User receiving delegation of the new domain; if this element is omitted, the domain is not delegated. Currently, domain delegation is not allowed, so no domains will have the delegation information. Attributes: <ul style="list-style-type: none"> <code>mayUseSubdomainNames</code> - <i>optional</i> Whether the receiving user may use sub-domain names of the given domain name Subelements: <ul style="list-style-type: none"> <code>username</code> - Name of the user receiving delegation of the new domain
displayString	<i>optional</i> value of the display string.
color1	<i>optional</i> hex code for the colour of the top bar on the web proxy.
color2	<i>optional</i> hex code for the colour of the title bar on the web proxy.
color3	<i>optional</i> hex code for the colour of the background on the web proxy.
css	<i>optional</i> ID for the CSS template to be applied on the web proxy.
pss	<i>optional</i> scope of the search feature; "false" to only show "This .tel" and "true" to allow searching throughout the .tel zone with "All .tels"

Example

Get domain greg.reggie.tel.

Request

```
<typ:getDomainRequest domainName="greg.reggie.tel"/>
```

Response

```
<ns4:getDomainResponse  
  xmlns:ns4="http://xmlns.telnic.org/ws/nsp/client/domain/types-1.0">  
  <ns4:displayString>Cartman's Domain</ns4:displayString>  
  <ns4:color1>#db1cdb</ns4:color1>  
  <ns4:color2>#f5f5f5</ns4:color2>  
  <ns4:color3>#bd42bd</ns4:color3>  
  <ns4:css>1</ns4:css>  
  <ns4:pss>true</ns4:pss>  
</ns4:getDomainResponse>
```

3.11. listDomains

This operation lists all domains associated with the calling user. Please note that because delegation is not allowed for .tel domains, any delegation-related information is not relevant.

In the request, only one of the optional parameters can be used in one request, not both.

listDomainsRequest

Attribute Name	Description
effUserName	<i>optional</i> name of the user performing the operation; used by administrators acting on behalf of other users. If a user with no administrator rights specifies the effUserName attribute, its value must match the name of the calling user.
apexonly	<i>optional</i> flag to only list apex (second-level) domains for the user when set to true ; no false value allowed

listDomainsResponse

Element Name	Description
domainName	Name of a domain in the list. Attributes: <ul style="list-style-type: none"> • mayUseSubdomainNames - Whether the receiving user may use sub-domain names of the given domain name • isDelegable - Whether the domain can be delegated • isDelegated- Whether the domain has been delegated to the requested user <p>The attributes isDelegable and isDelegated are mutually exclusive.</p>

Example

Request

```
<typ:listDomainsRequest/>
```

Response

```
<ns4:listDomainsResponse
xmlns:ns4="http://xmlns.telnic.org/ws/nsp/client/domain/types-1.0">
<ns4:domainName isDelegable="true" isDelegated="false"
mayUseSubdomainNames="false">greg.reggie.tel</ns4:domainName>
<ns4:domainName isDelegable="true" isDelegated="false"
mayUseSubdomainNames="false">reggie.tel</ns4:domainName>
<ns4:domainName isDelegable="true" isDelegated="false"
```

```
    mayUseSubdomainNames="false">regina.tel</ns4:domainName>  
</ns4:listDomainsResponse>
```

4. RESOURCE RECORDS

4.1. Resource Records

This API allows the management of [resource records](#) that are published in a domain, see the [Record-1.0.xsd](#) schema.

Note: All operations in this API group have a required attribute, `domainName`, specifying the name of the domain on which to act.

Table 4: Summary of the resource record management API.

Operation	Description
storeRecord	Adds new or replaces existing record.
deleteRecord	Deletes record.
listRecords	Lists records of a domain.
updateRecords	Performs a bulk update of a domain's content.

4.2. storeRecord

This operation allows a single record to be stored. Please refer to the sample code at the end of this section:

- [Example 1](#) - TXT record
- [Example 2](#) - NAPTR record to store a web link with labels
- [Example 3](#) - Non-terminal NAPTR record to store a link to a sub-domain
- [Example 4](#) - TXT record with an advertisement
- [Example 5](#) - NAPTR record to link to an image, shown at the top of the folder

storeRecordRequest

Exactly one of the top-level elements with its sub-elements must be present to signify the type of record. Each valid child element of the `storeRecordRequest` element (listed in the second part of the table below) specifies a DNS record type, and uses all the attributes listed directly below.

Attribute Name	Description
<code>id</code>	<p><i>optional</i> Record id; interpreted as follows:</p> <ul style="list-style-type: none"> • If no id is given, the record is added to the current set of records and assigned a new id. • If an existing id is given, the new record replaces the old one. The new record may be totally different from the old record. • If a non-existent id is given, the system returns an error.
<code>owner</code>	<p><i>optional</i> Owner name of the record relative to the domain in which it is contained. Similar to the DNS notation, an at-sign (@) denotes the domain's name itself.</p>
<code>class</code>	<p><i>optional</i> DNS class that the record belongs to.</p> <p>Format: numeric with values from 0 to 254 or symbolic, with one of the following values: IN (Internet), CH (Chaos), CS (CSNET), HS (Hesoid). See RFC 1034, 1035 for details.</p> <p>Default value: IN</p>
<code>ttl</code>	<p><i>optional</i> Time-to-live value for the record, which sets the maximum time in seconds, for which the record can be cached by a name server.</p> <p>If omitted, a system default is used.</p>
<code>profiles</code>	<p><i>optional</i> List of ids of profiles, with which the record will be associated.</p>

If omitted or empty, the record is associated with the default profile.

Special values:

- `_none_` - associated with no profile
- `_all_` - associated with all profiles, including future profiles

`groups`

optional List of ids of groups, with which the record will be associated.

Note: for non-encryptable records, this attribute must be empty or omitted.

`effUserName`

optional name of the user performing the operation; used by administrators acting on behalf of other users. If a user with no administrator rights specifies the `effUserName` attribute, its value must match the name of the calling user.

Element Name	Description
<code>txt</code>	One or multiple subelements <code><text></code> to specify the free text contact data
<code>naptr</code>	<p>Contact information, see whitepaper on NAPTRs for a description of this record type and its processing within the .tel architecture.</p> <p>Subelements:</p> <ul style="list-style-type: none"> • <code>order</code> - order of this record • <code>preference</code> - preference of this record • <code>flags</code> - any flags for this record • <code>services</code> - services of this record; may include "+x-lbl:" value with a short label that's stored inside the NAPTR; the short label is now deprecated and replaced by the long label • <code>longLabel</code> - text up to 255 characters long in UTF; this will be stored as a separate TXT .tlb record that displays as a description for this NAPTR record • <code>regexp</code> or <code>replacement</code> - A choice between the regular expression version and the replacement version; for non-terminal NAPTRs (NTNs) the replacement domain name is specified.
<code>srv</code>	The location of server(s) for a specific protocol and domain,

RFC2782.

Subelements:

- `priority` - priority of this record
- `weight` - weight of this record
- `port` - port number for this record
- `target` - target of this record

`mx`

Mail exchange record, see RFC1035.

Subelements:

- `priority` - priority of this record
- `exchange` - domain name of the mail exchange server for the record

`loc`

Location as a combination of longitude, latitude and altitude, RFC 1876.

Subelements:

- `longitude` with attribute `direction` (north or south) and subelements:
 - `degrees`
 - *optional* `minutes`
 - *optional* `seconds`
- `latitude` with attribute `direction` (east or west) and the same subelements as `longitude`
- `altitude` - altitude of the entity, in meters
- *optional* `size` - size of the described entity, in meters
- *optional* `horizPre` - horizontal precision, in meters
- *optional* `vertPre` - vertical precision, in meters

`ninfo`

A sequence of `text` nodes describing the current status of the domain owner, such as "on vacation" or "in California".

`generic`

Generic record type.

Subelements:

- `type` - record type
- `data` - binary representation of the record data

storeRecordResponse

The response repeats the `id` given in the request or returns the newly assigned `id`.

Element Name	Description
<code>id</code>	The record <code>id</code> : the one specified in the request, or a newly assigned <code>id</code> if the request contained no <code>id</code> .

Example 1

Create a new TXT record with the owner name `sample.reggie.tel` associated with profile 5.

Request

```
<typ:storeRecordRequest domainName="reggie.tel">
<typ:txt ttl="3600" owner="sample" profiles="5" groups="">
  <typ:text>Hello World!</typ:text>
</typ:txt>
</typ:storeRecordRequest>
```

Response

```
<storeRecordResponse
xmlns="http://xmlns.telnic.org/ws/nsp/client/record/types-1.0">
<id>61</id>
</storeRecordResponse>
```

Example 2

Create a NAPTR record with a link to a URL with a short label and a long label. The following records are produced:

```
telnicnsp10.tel. 60 IN NAPTR 10 57 "u" "E2U+web:http+x-lbl:MyValue"
"!^.*$!http://www.gmail.co.uk!" .
telnicnsp10.tel. 60 IN TXT ".tlb" "1" "10" "57" "Long label value for
contact"
```

Request

```
<typ:storeRecordRequest domainName="telnicnsp10.tel">
<typ:naptr owner="@" class="in" ttl="1000">
  <typ:order>10</typ:order>
  <typ:preference>57</typ:preference>
  <typ:flags>u</typ:flags>
  <typ:services>E2U+web:http+x-lbl:MyValue</typ:services>
  <typ:regexp>!^.*$!http://www.gmail.co.uk!</typ:regexp>
  <typ:longLabel>Long label value for contact</typ:longLabel>
</typ:naptr>
</typ:storeRecordRequest>
```

Response

```
<storeRecordResponse
xmlns="http://xmlns.telnic.org/ws/nsp/client/record/types-1.0">
<id>59</id>
</storeRecordResponse>
```

Example 3

Create a non-terminal NAPTR record pointing to example.tel. Note the terminating dot used in the <replacement> element to obtain an absolute domain name.

The following NAPTR is produced:

```
telnicnsp10.tel. 1000 IN NAPTR 10 17 "" "" "" example.tel.
```

Request

```
<typ:storeRecordRequest domainName="telnicnsp10.tel">
<typ:naptr>
```

March 2010

```

<typ:order>10</typ:order>
<typ:preference>17</typ:preference>
<typ:flags></typ:flags>
<typ:services></typ:services>
<typ:replacement>example.tel.</typ:replacement>
</typ:naptr>
</typ:storeRecordRequest>

```

Response

```

<storeRecordResponse
xmlns="http://xmlns.telnic.org/ws/nsp/client/record/types-1.0">
<id>17</id>
</storeRecordResponse>

```

Example 4

Create a TelAd record for the advertisement.

The following TXT is produced:

```

telnicnsp10.tel.IN TXT ".tad" "1" "1" "51" "Plumber"
"www.ReactFast.co.uk/Plumbers" "uri" "http://www.reactfast.co.uk" "desc"
"24 Hour Fast Response Plumbers" "desc" "10% Book Online Discount Today !"

```

Request

```

<typ:storeRecordRequest domainName="telnicnsp10.tel">
<typ:txt profiles="_all_" owner="_ad">
<typ:text>.tad</typ:text>
<typ:text>1</typ:text>
<typ:text>1</typ:text>
<typ:text>51</typ:text>
<typ:text>Plumber</typ:text>
<typ:text>www.ReactFast.co.uk/Plumbers</typ:text>
<typ:text>uri</typ:text>
<typ:text>http://www.reactfast.co.uk</typ:text>
<typ:text>desc</typ:text>
<typ:text>24 Hour Fast Response Plumbers</typ:text>
<typ:text>desc</typ:text>
<typ:text>10% Book Online Discount Today !</typ:text>
</typ:txt>
</typ:storeRecordRequest>

```

Response

```

<storeRecordResponse
xmlns="http://xmlns.telnic.org/ws/nsp/client/record/types-1.0">
<id>22</id>
</storeRecordResponse>

```

Example 5

Create a NAPTR record with a link to an image; the x-photo type means that the image will be rendered as a logo at the top of the page, rather than linked to as a web URL.

The following record is produced:

```

IN NAPTR 100 103 "u" "E2U+x-photo:http+x-lbl:Mugshot" "!^.*$!
http://server.url/path.jpg!" .

```

Request

```
<typ:storeRecordRequest domainName="telnicnsp10.tel">
<typ:naptr owner="@ " class="in" ttl="1000">
  <typ:order>10</typ:order>
  <typ:preference>57</typ:preference>
  <typ:flags>u</typ:flags>
  <typ:services>E2U+x-photo:http+x-lbl:Mugshot</typ:services>
  <typ:regexp>!^.*$! http://server.url/path.jpg!</typ:regexp>
</typ:naptr>
</typ:storeRecordRequest>
```

Response

```
<storeRecordResponse
xmlns="http://xmlns.telnic.org/ws/nsp/client/record/types-1.0">
<id>44</id>
</storeRecordResponse>
```

4.3. deleteRecord

This operation allows to delete a record with the given id.

deleteRecordRequest

Attribute Name	Description
domainName	The name of the domain, with which the record is associated.
effUserName	<i>optional</i> name of the user performing the operation; used by administrators acting on behalf of other users. If a user with no administrator rights specifies the <code>effUserName</code> attribute, its value must match the name of the calling user.

Element Name	Description
id	Id of the record to be deleted.

deleteRecordResponse

No elements or attributes.

Example

Delete record 58 of domain reggie.tel.

Request

```
<typ:deleteRecordRequest domainName="reggie.tel">  
  <typ:id>58</typ:id>  
</typ:deleteRecordRequest>
```

Response

```
<deleteRecordResponse  
xmlns="http://xmlns.telnic.org/ws/nsp/client/record/types-1.0"/>
```

4.4. updateRecords

This operation can update one or multiple records.

To add more flexibility and efficiency to this operation, this API allows the client application to:

- skip the [deleteRecord](#) operation by using [updateRecords](#) or [storeRecord](#) with an existing record id to overwrite an existing record
- delete all existing content by using the 'replace all' implementation
- group records that share the same attributes (`owner`, `class`, `tTL`, `profiles` or `groups`) via group elements

updateRecordsRequest

Attribute Name	Description
<code>effUserName</code>	<i>optional</i> name of the user performing the operation; used by administrators acting on behalf of other users. If a user with no administrator rights specifies the <code>effUserName</code> attribute, its value must match the name of the calling user.

Element Name	Description
<code>delete</code>	Id of a single record to be deleted; multiple elements are allowed in one operation. Elements <code>delete</code> and <code>deleteAll</code> are mutually exclusive.
<code>deleteAll</code>	All existing records will be eliminated before any records contained in this request are added. Elements <code>delete</code> and <code>deleteAll</code> are mutually exclusive.
<code>_record_</code>	Replace <code>_record_</code> with <code>txt</code> , <code>naptr</code> , <code>srv</code> , <code>mx</code> , <code>loc</code> , <code>ninfo</code> , or <code>generic</code> . See storeRecord for a description of these elements, their attributes and subelements.
<code>group</code>	A group of records of the same type and sharing some or all of the attributes; used instead of an individual record. Further sub-grouping is allowed.

updateRecordsResponse

No elements or attributes.

Example

Delete all existing records, then create a text record and a few records belonging to the profile with id 2 using a subgroup to overwrite the owner attribute of the mail exchange records.

Request

```
<typ:updateRecordsRequest domainName="reggie.tel">'
  <typ:deleteAll/>
  <typ:txt owner="@ " class="in">
    <typ:text>Hello Welt!</typ:text>
  </typ:txt>
  <typ:group owner="mystuff" profiles="2">
    <typ:naptr class="in" ttl="1000">
      <typ:order>10</typ:order>
      <typ:preference>20</typ:preference>
      <typ:flags>a</typ:flags>
      <typ:services>b</typ:services>
      <typ:regexp>c</typ:regexp>
    </typ:naptr>
    <typ:naptr class="1" groups="13" id="62" owner="@ " profiles="5">
      <typ:order>42</typ:order>
      <typ:preference>10</typ:preference>
      <typ:flags>u</typ:flags>
      <typ:services>E2U+mailto</typ:services>
      <typ:regexp>!^.*$!mailto:theemail@example.com!</typ:regexp>
    </typ:naptr>
    <typ:group owner="mail" profiles="2">
      <typ:mx class="in" ttl="3928">
        <typ:priority>20</typ:priority>
        <typ:exchange>mymail1.example.com.</typ:exchange>
      </typ:mx>
      <typ:mx class="in" ttl="3928">
        <typ:priority>20</typ:priority>
        <typ:exchange>mymail2.example.com.</typ:exchange>
      </typ:mx>
    </typ:group>
  </typ:group>
</typ:updateRecordsRequest>
```

Response

```
<updateRecordsResponse
  xmlns="http://xmlns.telnic.org/ws/nsp/client/record/types-1.0"/>
```

4.5. listRecords

This operation retrieves all records associated with a given domain name.

listRecordsRequest

No elements.

Attribute Name	Description
domainName	The name of the domain, with which the record is associated.
effUserName	<i>optional</i> name of the user performing the operation; used by administrators acting on behalf of other users. If a user with no administrator rights specifies the <code>effUserName</code> attribute, its value must match the name of the calling user.

listRecordsResponse

Element Name	Description
<code>_record_</code>	Replace <code>_record_</code> with <code>txt</code> , <code>naptr</code> , <code>srv</code> , <code>mx</code> , <code>loc</code> , <code>ninfo</code> , or generic. See storeRecord for a description of these elements, their attributes and subelements.

Example

Request

```
<typ:listRecordsRequest domainName="reggie.tel"/>
```

Response

```
<listRecordsResponse
xmlns="http://xmlns.telnic.org/ws/nsp/client/record/types-1.0">
  <txt class="1" groups="" id="61" owner="sample" profiles="5"
ttl="3600">
    <text>Hello World!</text>
  </txt>
  <typ:naptr class="1" groups="13" id="62" owner="@ " profiles="5">
    <typ:order>102</typ:order>
    <typ:preference>10</typ:preference>
    <typ:flags>u</typ:flags>
    <typ:services>E2U+mailto</typ:services>
    <typ:regexp>!^.*$!mailto:myemail@example.com!</typ:regexp>
  </typ:naptr>
  <typ:naptr class="1" groups="12 11" id="68" owner="@ " profiles="5"
ttl="7200">
    <order>99</order>
    <preference>100</preference>
    <flags/>
    <services/>

```

```
<replacement>www.example.com.</replacement>  
</typ:naptr>  
</listRecordsResponse>
```

5. DATA PRIVACY

5.1. Data Privacy

This interface provides all the functionality required by clients for private data management, i.e. provisioning of readers and the grouping of readers and associated resource records. Please read the [Security](#) page for an overview of the privacy model. For definitions of provided interfaces, see the [Reader-1.0.xsd](#) schema file.

Table 5: Summary of the public/private data interface API

Operation	Description
createGroup	Create a new group for a domain.
updateGroup	Update an existing group with new data.
deleteGroup	Delete an existing group from a domain.
getGroup	Get all data of a specific group.
listGroups	List all groups stored for a domain.
createReader	Create a new reader for a domain.
updateReader	Update an existing reader with new data.
deleteReader	Delete an existing reader from a domain.
getReader	Get all data of a specific reader.
listReaders	List all readers stored for a domain.

5.2. createGroup

This creates a new [group](#); returns the newly assigned group id in the response.

createGroupRequest

Attribute Name	Description
domainName	Name of the domain, with which the group is associated.
effUserName	<i>optional</i> name of the user performing the operation; used by administrators acting on behalf of other users. If a user with no administrator rights specifies the <code>effUserName</code> attribute, its value must match the name of the calling user.

Element Name	Description
name	Name of the group displayed to the user.
readers	<i>optional</i> Whitespace - separated list of the readers initially belonging to the group.
records	<i>optional</i> ids of the resource records initially associated with the group.

createGroupResponse

Element Name	Description
id	Id of the created group.

Example

Create a new group Friends with readers 15 and 1.

Request

```
<typ:createGroupRequest domainName="reggie.tel">
  <typ:name>Friends</typ:name>
  <typ:readers>15 1</typ:readers>
</typ:createGroupRequest>
```

Response

```
<ns2:createGroupResponse
  xmlns:ns2="http://xmlns.telnic.org/ws/nsp/client/reader/types-1.0">
  <ns2:id>17</ns2:id>
</ns2:createGroupResponse>
```

5.3. createReader

This operation creates a new [Reader](#) object and links it to the name of the domain, to which the reader will gain access. The reader is identified by this domain name, which is a pseudo name assigned by the system to support guest users. The response contains the id of the reader.

createReaderRequest

Attribute Name	Description
domainName	Name of the domain, with which the user is associated.
effUserName	<i>optional</i> name of the user performing the operation; used by administrators acting on behalf of other users. If a user with no administrator rights specifies the <code>effUserName</code> attribute, its value must match the name of the calling user.

Element Name	Description
name	Name of the reader.
readerPseudoDomainName	The reader's pseudo domain name to be used to calculate the private sub-domain label.
keyLocation	Domain where the public key of the reader is located.
reference	The reference value received in the friending request from this reader, if any.
groups	<i>optional</i> List of groups to which the reader will belong.

createReaderResponse

Element Name	Description
id	Id of the reader.
label	The label that shall be used to publish the data for the specific reader.

Example

Create a reader object and add it to the group with id 16. The example assumes that such a group exists.

Request

```
<typ:createReaderRequest domainName="reggie.tel">
  <typ:name>Jane Doe</typ:name>
  <typ:readerPseudoDomainName>1234.reader.keys.tel</typ:readerPseudoDomainName>
</typ:createReaderRequest>
```

March 2010

```
ame>
  <typ:reference>ffb7afb10fc19137ecd42f298ea6acb247c48141</typ:reference>
  <typ:keyLocation>jane.doe.keys.tel</typ:keyLocation>
  <typ:reference>ffb7afb10fc19137ecd42f298ea6acb247c48141</typ:reference>
  <typ:groups>16</typ:groups>
</typ:createReaderRequest>
```

Response

```
<ns2:createReaderResponse
  xmlns:ns2="http://xmlns.telnic.org/ws/nsp/client/reader/types-1.0">
  <ns2:id>11</ns2:id>
  <ns2:label>61e09be55eb4fd64cc20d802bdfa22722ae23efa</ns2:label>
</ns2:createReaderResponse>
```

5.4. deleteGroup

This operation deletes a group.

Note that all resource records that have exclusively been associated with this group are deleted also, in order to avoid an accidental exposure of private data. However, if the resource records are still associated with other groups, they are not deleted.

Readers are never deleted, even if they have been exclusively associated with the group.

deleteGroupRequest

Attribute Name	Description
domainName	Name of the domain with which the group is associated.
effUserName	<i>optional</i> name of the user performing the operation; used by administrators acting on behalf of other users. If a user with no administrator rights specifies the <code>effUserName</code> attribute, its value must match the name of the calling user.

Element Name	Description
id	Id of the group.

deleteGroupResponse

No elements or attributes.

Example

Delete group of id 17 associated with domain reggie.tel.

Request

```
<typ:deleteGroupRequest domainName="reggie.tel">
  <typ:id>17</typ:id>
</typ:deleteGroupRequest>
```

Response

```
<deleteGroupResponse
xmlns="http://xmlns.telnic.org/ws/nsp/client/reader/types-1.0"/>
```

5.5. deleteReader

This operation deletes a reader of the specified domain name.

deleteReaderRequest

Attribute Name	Description
domainName	Name of the domain with which the user was be associated.
effUserName	<i>optional</i> name of the user performing the operation; used by administrators acting on behalf of other users. If a user with no administrator rights specifies the <code>effUserName</code> attribute, its value must match the name of the calling user.

Element Name	Description
id	Id of the group.

deleteReaderResponse

No elements or attributes.

Example

Delete reader of id 11 associated with domain reggie.tel.

Request

```
<typ:deleteReaderRequest domainName="reggie.tel">  
  <typ:id>11</typ:id>  
</typ:deleteReaderRequest>
```

Response

```
<deleteReaderResponse  
xmlns="http://xmlns.telnic.org/ws/nsp/client/reader/types-1.0"/>
```

5.6. getGroup

Returns the details of the group: the name, readers, and records.

getGroupRequest

Attribute Name	Description
domainName	Name of the domain that the group belongs to.
effUserName	<i>optional</i> name of the user performing the operation; used by administrators acting on behalf of other users. If a user with no administrator rights specifies the <code>effUserName</code> attribute, its value must match the name of the calling user.

Element Name	Description
id	Id of the group.

getGroupResponse

Element Name	Description
name	Name of the group displayed to the user.
readers	Whitespace - separated list of the readers belonging to the group.
records	ids of the resource records associated with the group.

Example

Retrieve data of group 17 associated with domain reggie.tel.

Request

```
<typ:getGroupRequest domainName="reggie.tel">
  <typ:id>17</typ:id>
</typ:getGroupRequest>
```

Response

```
<ns2:getGroupResponse
xmlns:ns2="http://xmlns.telnic.org/ws/nsp/client/reader/types-1.0">
  <ns2:name>Close Friends</ns2:name>
  <ns2:readers>15</ns2:readers>
  <ns2:records>62 71</ns2:records>
</ns2:getGroupResponse>
```

5.7. getReader

Retrieves information about a reader.

getReaderRequest

Attribute Name	Description
domainName	The domain with which the user is associated.
effUserName	<i>optional</i> name of the user performing the operation; used by administrators acting on behalf of other users. If a user with no administrator rights specifies the <code>effUserName</code> attribute, its value must match the name of the calling user.

Element Name	Description
id	Id of the reader.

getReaderResponse

Element Name	Description
name	Name of the reader.
label	The label that used to publish the data for the reader.
keyLocation	Domain where the public key of the reader is located.
reference	The reference value received in the friending request from this reader, if any.
readerPseudoDomainName	The reader's pseudo domain name to be used to calculate the private sub-domain label.
groups	List of groups to which the reader belongs.

Example

Get reader with id 11 for domain reggie.tel.

Request

```
<typ:getReaderRequest domainName="reggie.tel">
  <typ:id>11</typ:id>
</typ:getReaderRequest>
```

Response

```
<ns2:getReaderResponse
xmlns:ns2="http://xmlns.telnic.org/ws/nsp/client/reader/types-1.0">
  <ns2:name>Jane Doe</ns2:name>
```

March 2010

```
<ns2:label>61e09be55eb4fd64cc20d802bdfa22722ae23befa</ns2:label>  
<ns2:keyLocation>jane.doe.keys.tel</ns2:keyLocation>  
<ns2:reference>ffb7afb10fc19137ecd42f298ea6acb247c48141</ns2:reference>  
<ns2:readerPseudoDomainName>1234.reader.keys.tel</ns2:readerPseudoDomainName>  
<ns2:groups>16</ns2:groups>  
</ns2:getReaderResponse>
```

5.8. listGroups

Returns the list of groups that exist for a given domain.

listGroupsRequest

Attribute Name	Description
domainName	Name of the domain, with which the groups are associated.
effUserName	<i>optional</i> name of the user performing the operation; used by administrators acting on behalf of other users. If a user with no administrator rights specifies the <code>effUserName</code> attribute, its value must match the name of the calling user.

listGroupsResponse

Element Name	Description
group	Name of the group in the list of groups associated with the given domain. Attributes: <ul style="list-style-type: none"> <code>id</code> - the id of the group

Example

Request

```
<typ:listGroupsRequest domainName="reggie.tel"/>
```

Response

```
<ns2:listGroupsResponse
  xmlns:ns2="http://xmlns.telnic.org/ws/nsp/client/reader/types-1.0">
  <ns2:group id="17">Close Friends</ns2:group>
  <ns2:group id="12">Friends</ns2:group>
  <ns2:group id="5">Family</ns2:group>
</ns2:listGroupsResponse>
```

5.9. listReaders

This operation lists all readers associated with a domain name.

listReadersRequest

Attribute Name	Description
domainName	Name of the domain, with which the readers are associated.
effUserName	<i>optional</i> name of the user performing the operation; used by administrators acting on behalf of other users. If a user with no administrator rights specifies the <code>effUserName</code> attribute, its value must match the name of the calling user.

listReadersResponse

Element Name	Description
reader	Name of the reader in the list of readers associated with the given domain. Attributes: <ul style="list-style-type: none"> <code>id</code> - the id of the reader

Example

Request

```
<typ:listReadersRequest domainName="reggie.tel"/>
```

Response

```
<ns2:listReadersResponse
  xmlns:ns2="http://xmlns.telnic.org/ws/nsp/client/reader/types-1.0">
  <ns2:reader id="11">Jane Doe</ns2:reader>
  <ns2:reader id="3">John Doe</ns2:reader>
</ns2:listReadersResponse>
```

5.10. updateGroup

This operation updates the name and associated readers and records for a group. The data not included in the update request is left unchanged. If a resource record association is removed and this record is no longer associated with any group, the record is removed from the system in order to avoid accidental exposure of private data.

updateGroupRequest

Attribute Name	Description
domainName	Name of the domain that the group belongs to.
effUserName	<i>optional</i> name of the user performing the operation; used by administrators acting on behalf of other users. If a user with no administrator rights specifies the <code>effUserName</code> attribute, its value must match the name of the calling user.

Element Name	Description
id	Id of the group to be updated.
name	<i>optional</i> Name of the group displayed to the user.
readers	<i>optional</i> Whitespace - separated new list of the readers.
records	<i>optional</i> New list of resource record ids associated with the group.

updateGroupResponse

No elements or attributes.

Example

Update group 17 to be called "Close Friends", to include reader 15 and to show records 62 and 71.

Request

```
<typ:updateGroupRequest domainName="reggie.tel">
  <typ:id>17</typ:id>
  <typ:name>Close Friends</typ:name>
  <typ:readers>15</typ:readers>
  <typ:records>62 71</typ:records>
</typ:updateGroupRequest>
```

Response

```
<updateGroupResponse
  xmlns="http://xmlns.telnic.org/ws/nsp/client/reader/types-1.0"/>
```

5.11. updateReader

This operation allows to update the reader data and/or to update the groups the reader is associated with.

updateReaderRequest

Attribute Name	Description
domainName	Name of the domain with which the user is associated.
effUserName	<i>optional</i> name of the user performing the operation; used by administrators acting on behalf of other users. If a user with no administrator rights specifies the <code>effUserName</code> attribute, its value must match the name of the calling user.

Element Name	Description
id	Id of the reader to be updated.
name	Name of the reader.
readerPseudoDomainName	The reader's pseudo domain name to be used to calculate the private sub-domain label.
keyLocation	Domain where the public key of the reader is located.
reference	The reference value received in the friending request from this reader, if any.
groups	<i>optional</i> New list of groups to which the reader will belong.

updateReaderResponse

Element Name	Description
label	The label that shall be used to publish the data for the specific reader.

Example

Update reader 11 with the name Jane Doe and all required data.

Request

```
<typ:updateReaderRequest domainName="reggie.tel">
  <typ:id>11</typ:id>
  <typ:name>Jane Doe</typ:name>
  <typ:readerPseudoDomainName>1234.reader.keys.tel</typ:readerPseudoDomainName>
</typ:updateReaderRequest>
```

March 2010

```
<typ:keyLocation>jane.doe.keys.tel</typ:keyLocation>  
<typ:reference>ffb7afb10fc19137ecd42f298ea6acb247c48141</typ:reference>  
<typ:groups>16</typ:groups>  
</typ:updateReaderRequest>
```

Response

```
<updateReaderResponse  
  xmlns="http://xmlns.telnic.org/ws/nsp/client/reader/types-1.0"/>  
  <typ:label>61e09be55eb4fd64cc20d802bdfa22722ae23efa</typ:label>
```

6. PROFILES

6.1. Profiles

This interface provides all the functionality required by clients to create, edit, delete, query and switch [profiles](#), see the [Profile-1.0.xsd](#) schema file.

Table 6: Summary of the profile editing and switching API.

Operation	Description
createProfile	Create a new profile for a domain.
deleteProfile	Delete an existing profile from a domain.
updateProfile	Update an existing profile with new data.
getProfile	Get all data of a specific profile.
listProfiles	List all profiles stored for a domain.
listProfilesExt	List all profiles stored for a domain with details.
switchToProfile	Switch to the specified profile.
getActiveProfile	Determine the currently active profile.

6.2. createProfile

This operation creates a new profile. The system assigns an id to this profile and returns it in the response.

createProfileRequest

Attribute Name	Description
domainName	Name of the domain for which the profile is created.
effUserName	<i>optional</i> name of the user performing the operation; used by administrators acting on behalf of other users. If a user with no administrator rights specifies the <code>effUserName</code> attribute, its value must match the name of the calling user.

Element Name	Description
profile	The profile to be updated, with the optional attribute <code>default</code> to set the new profile as the default one.
name	Unique profile name.
records	<i>optional</i> List of record ids making up the profile.

createProfileResponse

Element Name	Description
id	Id of created profile

Example

Create profile "Work" associated with resource records 7, 12, and 23.

Request

```
<typ:createProfileRequest domainName="reggie.tel">
  <typ:profile default="false">
    <typ:name>Work</typ:name>
    <typ:records>7 12 23</typ:records>
  </typ:profile>
</typ:createProfileRequest>
```

Response

```
<ns3:createProfileResponse
  xmlns:ns3="http://xmlns.telnic.org/ws/nsp/client/profile/types-
1.0">
  <ns3:id>24</ns3:id>
</ns3:createProfileResponse>
```

6.3. deleteProfile

This operation deletes an existing profile. The default profile cannot be deleted.

The operation receives a `DeleteProfileRequestMessage` and returns a `DeleteProfileResponseMessage`, expressed by the `deleteProfileRequest` and `deleteProfileResponse` elements from the XSD, respectively.

deleteProfileRequest

Attribute Name	Description
<code>domainName</code>	The domain with which the profile is associated
<code>effUserName</code>	<i>optional</i> name of the user performing the operation; used by administrators acting on behalf of other users. If a user with no administrator rights specifies the <code>effUserName</code> attribute, its value must match the name of the calling user.

Element Name	Description
<code>profile</code>	The profile to be deleted. Identified by the <code>id</code> attribute.

deleteProfileResponse

No elements or attributes.

Example

Delete the profile with the id 24.

Request

```
<typ:deleteProfileRequest domainName="reggie.tel">
  <typ:profile id="24"/>
</typ:deleteProfileRequest>
```

Response

```
<deleteProfileResponse
xmlns="http://xmlns.telnic.org/ws/nsp/client/profile/types-1.0"/>
```

6.4. updateProfile

This operation updates an existing profile.

updateProfileRequest

Attribute Name	Description
domainName	Name of the domain for which the profile is updated.
effUserName	<i>optional</i> name of the user performing the operation; used by administrators acting on behalf of other users. If a user with no administrator rights specifies the <code>effUserName</code> attribute, its value must match the name of the calling user.

Element Name	Description
profile	The profile to be updated. Attributes: <ul style="list-style-type: none"> <code>id</code> - ID of the profile to be updated <code>default</code> - <i>optional</i> flag to set the new profile as the default one. Note: there must be a default profile, so <code>default="false"</code> is silently ignored if applied to the default profile.
name	<i>optional</i> Unique new profile name.
records	<i>optional</i> New list of record ids making up the profile to replace the current list of resource records. If this element is omitted, the resource records list remains unchanged.

updateProfileResponse

No elements or attributes.

Example

This request updates the profile with the id 24 to be the default profile.

Request

```
<typ:updateProfileRequest domainName="reggie.tel">
  <typ:profile id="24" default="true">
    <typ:name>Holidays</typ:name>
    <typ:records>71 73</typ:records>
  </typ:profile>
</typ:updateProfileRequest>
```

Response

```
<updateProfileResponse  
xmlns="http://xmlns.telnic.org/ws/nsp/client/profile/types-1.0"/>
```

6.5. listProfiles

This operation retrieves the ids of all existing profiles of a domain. See [listProfilesExt](#) for an extended version returning more information.

listProfilesRequest

Attribute Name	Description
domainName	Name of the domain for which the list of profiles is requested.
effUserName	<i>optional</i> name of the user performing the operation; used by administrators acting on behalf of other users. If a user with no administrator rights specifies the <code>effUserName</code> attribute, its value must match the name of the calling user.

listProfilesResponse

Element Name	Description
profiles	List of profiles for the given domain. The ids of the profiles are provided as a whitespace - separated list in the value of attribute <code>id</code> .

Example

Request

```
<typ:listProfilesRequest domainName="reggie.tel"/>
```

Response

```
<ns3:listProfilesResponse  
xmlns:ns3="http://xmlns.telnic.org/ws/nsp/client/profile/types-1.0">  
<ns3:profiles ids="6 21 24"/>  
</ns3:listProfilesResponse>
```

6.6. listProfilesExt

This operation is an extended version of [listProfiles](#). The operation retrieves not only the ids of all existing profiles of a domain, but also their names and information about the active and default profiles.

listProfilesExtRequest

Attribute Name	Description
domainName	Name of the domain for which the list of profiles is requested.
effUserName	<i>optional</i> name of the user performing the operation; used by administrators acting on behalf of other users. If a user with no administrator rights specifies the <code>effUserName</code> attribute, its value must match the name of the calling user.

listProfilesExtResponse

Element Name	Description
profiles	List of profiles for the given domain. Attributes: <ul style="list-style-type: none"> <code>active</code> - currently active profile id <code>default</code> - default profile id Subelements: <ul style="list-style-type: none"> <code>profile</code> - individual profile associated with a domain Attributes of element <code>profile</code> : <ul style="list-style-type: none"> <code>id</code> - profile id <code>name</code> - unique profile name

Example

Request

```
<typ:listProfilesExtRequest domainName="reggie.tel"/>
```

Response

```
<ns3:listProfilesExtResponse
  xmlns:ns3="http://xmlns.telnic.org/ws/nsp/client/profile/types-
  1.0">
  <ns3:profiles active="21" default="6">
    <ns3:profile id="6" name="default"/>
    <ns3:profile id="21" name="Home"/>
    <ns3:profile id="24" name="Holidays"/>
  </ns3:profiles>
</ns3:listProfilesExtResponse>
```

6.7. getProfile

This operation retrieves the data of an existing profile.

getProfileRequest

Attribute Name	Description
domainName	Name of the domain for which the profile details are requested.
effUserName	<i>optional</i> name of the user performing the operation; used by administrators acting on behalf of other users. If a user with no administrator rights specifies the <code>effUserName</code> attribute, its value must match the name of the calling user.

Element Name	Description
profile	The profile for which to retrieve data. Identified by its <code>id</code> attribute.

getProfileResponse

Element Name	Description
profile	The profile with data. The attribute 'default' with the Boolean value identifies the status of the profile.
name	Unique name of this profile.
records	ids of resource records associated with this profile.

Example

This request retrieves the data of the profile with the id 24.

Request

```
<typ:getProfileRequest domainName="reggie.tel">
  <typ:profile id="24"/>
</typ:getProfileRequest>
```

Response

```
<ns3:getProfileResponse
xmlns:ns3="http://xmlns.telnic.org/ws/nsp/client/profile/types-1.0">

  <ns3:profile default="true">
    <ns3:name>Holidays</ns3:name>
    <ns3:records>73 71</ns3:records>
  </ns3:profile>
</ns3:getProfileResponse>
```

6.8. switchToProfile

This operation switches to a given profile, i.e. makes it the active one.

switchToProfileRequest

Attribute Name	Description
domainName	Name of the domain for which the update is requested.
effUserName	<i>optional</i> name of the user performing the operation; used by administrators acting on behalf of other users. If a user with no administrator rights specifies the <code>effUserName</code> attribute, its value must match the name of the calling user.

Element Name	Description
profile	The profile that the user wants to switch to. Identified by its <code>id</code> attribute.

switchToProfileResponse

No elements or attributes.

Example

Request

```
<typ:switchToProfileRequest domainName="reggie.tel">
  <typ:profile id="24"/>
</typ:switchToProfileRequest>
```

Response

```
<ns3:switchToProfileResponse
xmlns:ns3="http://xmlns.telnic.org/ws/nsp/client/profile/types-1.0"/>
```

6.9. getActiveProfile

This operation determines the currently active profile of a domain.

getActiveProfileRequest

Attribute Name	Description
domainName	Name of the domain for which the active profile is requested.
effUserName	<i>optional</i> name of the user performing the operation; used by administrators acting on behalf of other users. If a user with no administrator rights specifies the <code>effUserName</code> attribute, its value must match the name of the calling user.

getActiveProfileResponse

Element Name	Description
profile	The currently active profile. Identified by its <code>id</code> attribute.

Example

Request

```
<typ:getActiveProfileRequest domainName="reggie.tel"/>
```

Response

```
<ns3:getActiveProfileResponse
  xmlns:ns3="http://xmlns.telnic.org/ws/nsp/client/profile/types-
  1.0">
  <ns3:profile id="24"/>
</ns3:getActiveProfileResponse>
```

7. SEARCH DATA

7.1. Search Data

This interface provides all the functionality required by clients for [search data](#) management, see the [SearchData-1.0.xsd](#) schema file. The API presents an abstraction from the actual representation of the search data in the .tel zones. The generation of appropriate [resource records](#), as well as the notification of the [SO](#)'s zone crawler, is automatically done behind the scenes.

Table 7: Summary of the search data management API.

Operation	Description
setSearchData	Set/update search data of a domain, replacing all previous data (also allows the deletion of all search data by providing an empty new set of data)
getSearchData	Get the current search data of a domain.

7.2. Allowed Keywords

In addition to free-form text that can be added as keyword search data, the current API provides a number of pre-defined keyword types. See Tables 8 and 9 for currently available keywords. Note that each keyword has an abbreviation assigned to it, which is stored in the `field` attribute.

The interface also supports addition of new keyword types; however, it is recommended to consider using pre-defined types because the system is configured to group and display them correctly.

Table 8: Keywords for Individuals

	Grouping Keyword	Abbrev.	Description / Example
	domain	N/A	Label of the .tel domain. This is not really a keyword, but the hosting domain name (Unicode); it can be used on queries, but not in the TXT records.
Name	nameLabel	nl	Grouping keyword
	salutation	s	"Mr", "Mrs"
	firstName	fn	"Adam"
	lastName	ln	"Smith"
	nickName	nn	"Ade"
	commonName	cn	"Addie"
	dateOfBirth	dob	"1968-02-19"
	gender	g	"male", "female"
	maritalStatus	ms	"married", "single"
Address	postalAddress	pa	Grouping keyword
	addressLine[1..3]	a[1..3]	"8 Wilfried Street" Multiple lines/values possible
	townCity	tc	"London"
	stateProvince	sp	"Hampshire"
	postalCode	pc	"SW1E 6PL"
	country	c	"Scotland"

	latitudeLongitude	ll	Geographic latitude/longitude
Work	businessInformation	bi	Grouping keyword
	organization	o	To be left blank for entries representing corporations
	department	d	"IT"
	jobTitle	jt	"chief executive officer"
	hobbiesInterests	hi	"scuba diving"

In addition, a number of business-oriented keywords are allowed, see Table 9.

Table 9: Keywords for Businesses

	Grouping Keyword	Abbrev.	Description / Example
	businessPostalAddress	bpa	Company address
Business Details	directoryInformation	di	Grouping keyword
	businessName	bn	"Marie's Marriage Emporium"
	businessArea	bar	"Weddings"
	businessSubArea	bsa	"Dresses, Flowers, Venues"
	serviceArea	sa	Geographic area in which services are offered, e.g. "Essex, Cambridgeshire"

Finally, the `ft` keyword, free text, can be used by both individuals and businesses, but cannot be provisioned in a group with other keywords.

7.3. setSearchData

This operation sets new search data for a domain. It replaces all search data that may have been present prior to the operation; an empty string deletes all data.

setSearchDataRequest

Attribute Name	Description
domainName	Name of the domain for which the keywords are updated.
effUserName	<i>optional</i> name of the user performing the operation; used by administrators acting on behalf of other users. If a user with no administrator rights specifies the <code>effUserName</code> attribute, its value must match the name of the calling user.

Element Name	Description
searchData	Container for a sequence of <code>keyword</code> sub-elements. Subelements: <ul style="list-style-type: none"> <code>keyword</code> - primary (first-level) keyword; can include subordinate keyword elements, which in turn cannot have subordinates Attributes of <code>keyword</code>: <ul style="list-style-type: none"> <code>field</code> - non-empty string defining the keyword; see Allowed Keywords for a list of allowed values for this attribute <code>value</code> - the value associated with the particular keyword

setSearchDataResponse

No elements or attributes.

Example

Set search data for the domain `reggie.tel`: `fn` and `ln` are specified as primary keywords without any associated secondary keywords, while `bpa` and `pa` are primary keywords grouping address data (`tc`, `pc`) specified as secondary keywords.

Request

```
<typ:setSearchDataRequest domainName="reggie.tel">
  <typ:searchData>
    <typ:keyword field="fn" value="John"/>
    <typ:keyword field="ln" value="Smith"/>
    <typ:keyword field="bpa" value="My shop">
      <typ:keyword field="tc" value="Bristol"/>
      <typ:keyword field="pc" value="12345"/>
    </typ:keyword>
    <typ:keyword field="pa" value="My home">
      <typ:keyword field="tc" value="London"/>
    </typ:keyword>
  </typ:searchData>
</typ:setSearchDataRequest>
```

```
<typ:keyword field="pc" value="67890"/>  
</typ:keyword>  
</typ:searchData>  
</typ:setSearchDataRequest>
```

Response

```
<setSearchDataResponse  
xmlns="http://xmlns.telnic.org/ws/nsp/client/searchdata/types-1.0"/>
```

7.4. getSearchData

This operation retrieves the current set of search data stored for a given domain.

getSearchDataRequest

Attribute Name	Description
domainName	The name of the domain for which to retrieve keywords.
effUserName	<i>optional</i> name of the user performing the operation; used by administrators acting on behalf of other users. If a user with no administrator rights specifies the <code>effUserName</code> attribute, its value must match the name of the calling user.

getSearchDataResponse

Element Name	Description
searchData	Container for a sequence of <code>keyword</code> sub-elements. Subelements: <ul style="list-style-type: none"> <code>keyword</code> - primary (first-level) keyword; can include subordinate keyword elements, which in turn cannot have subordinates Attributes of <code>keyword</code>: <ul style="list-style-type: none"> <code>field</code> - non-empty string defining the keyword; see Allowed Keywords for a list of allowed values for this attribute <code>value</code> - the value associated with the particular keyword

Example

Retrieve the search data for the domain `reggie.tel`.

Request

```
<typ:getSearchDataRequest domainName="reggie.tel"/>
```

Response

```
<ns5:getSearchDataResponse
xmlns:ns5="http://xmlns.telnic.org/ws/nsp/client/searchdata/types-1.0">
  <ns5:searchData>
    <ns5:keyword field="fn" value="John"/>
    <ns5:keyword field="ln" value="Smith"/>
    <ns5:keyword field="bpa" value="My shop">
      <ns5:keyword field="tc" value="Bristol"/>
      <ns5:keyword field="pc" value="12345"/>
    </ns5:keyword>
    <ns5:keyword field="pa" value="My home">
```

March 2010

```
<ns5:keyword field="tc" value="London"/>  
<ns5:keyword field="pc" value="67890"/>  
</ns5:keyword>  
</ns5:searchData>  
</ns5:getSearchDataResponse>
```

8. DATA EXCHANGE

8.1. Data Exchange

This interface provides all the functionality required by clients to export and import the whole data that is associated with one or more domains: records, profiles, readers, groups, search data, and zones. The most important applications of this feature are to create backups and copies of data to support transfers.

During an exchange operation, all data is represented as valid XML content matching the defined schema format contained in an <exports> element. Multiple <exports> elements in the container can hold different representations of the same exported data in different formats. The system can support multiple formats and use the best-suited one. The currently used format is defined in [xml/Exchange-1.0.xsd](#). More formats will be introduced with time.

```
<?xml version="1.0" encoding="UTF-8"?>
<exports xmlns="http://xmlns.telnic.org/nsp/exchange-container-1.0">
<export xmlns="http://xmlns.telnic.org/nsp/exchange-1.0">
<domains>[...]</domains>
<zones>[...]</zones>
</export>
</exports>
```

For data exchange function definitions, please consult the [Exchange-1.0.xsd](#) schema file.

Table 11: Summary of the Export / Import API

Operation	AdminUser	Description
exportData	yes	permissionsExports the data of domains and zones.
importData	yes	permissionsImports the data of domains and zones.
restoreDomain	yes	permissionsImports the data for domain(s) and replaces any existing domains of the same name.

8.2. exportData

This operation allows a user to export the data associated with one or more domains and zones in one go. You can export the whole domain tree or a part of it. For the latter, use the optional `<domains>` element to list all the domains you need to export. This list needs to be complete as the export is not recursive.

Note that the `<privateSalt>` specifying a base-64-encoded string used to create the reference for friending messages, has a fixed length of 64. For brevity, the values of this element in the examples are twice as short.

exportDataRequest

Attribute Name	Description
----------------	-------------

<code>effUserName</code>	<i>optional</i> name of the user performing the operation; used by administrators acting on behalf of other users. If a user with no administrator rights specifies the <code>effUserName</code> attribute, its value must match the name of the calling user.
--------------------------	--

Element Name	Description
--------------	-------------

<code>domains</code>	<p><i>optional</i> List of domains, from which to export data. Domains are given as a sequence of <code>domainName</code> elements.</p> <p>If missing, all domains are exported.</p> <p>If present, only the specified domains are exported.</p> <p>If an empty list is present, no domain data is exported.</p> <p>If a non-existent domain is specified, it is silently ignored.</p> <p>If optional <code>recursive</code> attribute is set to <code>true</code>, all sub-domains of the given domain are exported. Otherwise, the command needs to explicitly list all sub-domains to be exported.</p>
----------------------	---

<code>zones</code>	<p><i>optional</i> List of zones, from which to export data. Zones are given as a sequence of <code>domainName</code> elements.</p> <p>If missing, all zones are exported.</p> <p>If present, only the specified zones are exported.</p> <p>If an empty list is present, no zone data is exported.</p> <p>If a non-existent zone is specified, it is silently ignored.</p>
--------------------	--

exportDataResponse

All requested data matching the requested domain and/or zone, including:

- profiles associated with the requested domain
- readers for the requested domain, including private reader information

March 2010

- groups of readers and records made available to these groups, including system information
- records published in the requested domain
- <privateSalt>, the system element used to calculate the private label, under which private data for a particular reader is stored
- zone information, including associated name servers

Example

Get data from the domain example.tel and zone example.tel.

Request

```
<typ:exportDataRequest>
  <typ:domains>
    <typ:domainName>example.tel</typ:domainName>
  </typ:domains>
  <typ:zones>
    <typ:domainName>example.tel</typ:domainName>
  </typ:zones>
</typ:exportDataRequest>
```

Response

```
<ns6:exportDataResponse
xmlns:ns6="http://xmlns.telnic.org/ws/nsp/client/exchange/types-1.0">
  <ns6:data>
    <![CDATA[<?xml version="1.0" encoding="UTF-8"?>
      <exports xmlns="http://xmlns.telnic.org/nsp/exchange-container-1.0">
        <export xmlns="http://xmlns.telnic.org/nsp/exchange-1.0">
          <domains> <domain domainName="example.tel">
            <profiles>
              <profile default="true">
                <name>work</name>
                <records>record17 record16</records> </profile>
              </profiles>
              <readers>
                <reader id="reader11">
                  <name>reggie</name>
                  <label>42bc4550afb3967205dd25bb3662eff12a72387c</label>
                  <keyLocation>d1ql.node1.keys.tel</keyLocation>
                  <reference>ae4fup89du02398ruisd9pitrjq3rj4umas3ioufp</reference>
                  <readerDomainName>1234.reader.keys.tel</readerDomainName>
                </reader> </readers>
              <groups>
                <group>
                  <name>friends</name>
                  <readers>reader11</readers>
                  <records>record16</records>
                  <allFriendsFlag>true</allFriendsFlag>
                </group>
                <group>
                  <name>family</name>
```

```
<readers></readers>
<records>record17 record16</records>
<allFriendsFlag>>false</allFriendsFlag>
</group> </groups>
<records>
<naptr class="1" id="record17" owner="@">
  <order>100</order>
  <preference>100</preference>
  <flags>u</flags>
  <services>E2U+voice:tel+x-work</services>
  <regexp>!^.*$!tel:+42.4!</regexp>
</naptr>
<naptr class="1" id="record16" owner="@">
  <order>100</order>
  <preference>90</preference>
  <flags>u</flags>
  <services>E2U+voice:tel+x-home</services>
  <regexp>!^.*$!tel:+44.1!</regexp>
</naptr>
</records>
<searchData/>
<privateSalt>
  D1zzX0l9nAX0v1MF5jgXN3zjy59OU42ftaLjyGCVyG9zbnk5hyjugqjku
</privateSalt>
</domain> </domains>
<zones>
  <zone zoneName="example.tel">
    <nameservers>
      <host>ns1.example.tel</host>
      <host>ns2.example.tel</host>
    </nameservers>
  </zone> </zones>
</export> </exports>]]>
</ns6:data>
</ns6:exportDataResponse>
```

8.3. importData

This operation allows a user to import the data associated with one or more domains and zones in one go. You can import the whole exported data set, or a part of it. To limit the imported data set, do one of the following:

- Specify exactly which domains to import; this operation is not recursive, so a full list is needed
- Clean up the data section of the data set to remove everything that you do not need
- Limit the export data set, so you don't have to clean up later

Note: if the import operation is performed by the user, the zone information is ignored because zones are managed by the TelHosting provider. It's usually easiest to remove information about zones from the imported data set.

importDataRequest

Attribute Name	Description
<code>overwrite</code>	<p>Boolean value specifying whether existing data will be overwritten.</p> <p>If set to <code>true</code>, the domain for which the data is imported, will have exactly the same data as specified in the import operation. Other data is removed from the domain.</p> <p>If set to <code>false</code>, no data is changed for existing domains.</p>
<code>effUserName</code>	<p><i>optional</i> name of the user performing the operation; used by administrators acting on behalf of other users. If a user with no administrator rights specifies the <code>effUserName</code> attribute, its value must match the name of the calling user.</p>

Element Name	Description
<code>domains</code>	<p><i>optional</i> List of domains, from which to import data. Domains are given as a sequence of <code>domainName</code> elements.</p> <p>If missing, all domains are imported.</p> <p>If present, only the specified domains are imported.</p> <p>If an empty list is present, no domain data is imported.</p> <p>If a non-existent domain is specified, it is silently ignored.</p>
<code>zones</code>	<p><i>optional</i> List of zones, from which to imported data. Zones are given as a sequence of <code>domainName</code> elements.</p> <p>If missing, all zones are imported.</p> <p>If present, only the specified zones are imported.</p>

If an empty list is present, no zone data is imported.

If a non-existent zone is specified, it is silently ignored.

data

The imported data set as valid XML in UTF-8 encoding; see [exportData](#) for a description of the data that can be exported/imported.

importDataResponse

Element Name	Description
domains	Sequence of domainName elements with actually imported domains.
zones	Sequence of domainName elements with actually imported zones.

Example

Request

```
<typ:importDataRequest>
  <typ:domains>
    <typ:domainName>example.tel</typ:domainName> </typ:domains>
  <typ:zones/>
  <ns6:data>
    <![CDATA[<?xml version="1.0" encoding="UTF-8"?>
1.0">
    <exports xmlns="http://xmlns.telnic.org/nsp/exchange-container-
    <export xmlns="http://xmlns.telnic.org/nsp/exchange-1.0">
      <domains> <domain domainName="example.tel">
        <profiles> <profile default="true">
          <name>always</name>
          <records>record42 record43</records>
        </profile> </profiles>
        <readers/> <groups/>
      <records>
        <naptr class="1" id="record42" owner="@">
          <order>100</order>
          <preference>100</preference>
          <flags>u</flags>
          <services>E2U+voice:tel+x-work</services>
          <regex>!^.*$!tel:+42.4!</regex>
        </naptr>
        <naptr class="1" id="record43" owner="@">
          <order>100</order>
          <preference>100</preference>
          <flags>u</flags>
          <services>E2U+voice:tel+x-mobile</services>
          <regex>!^.*$!tel:+42.543342!</regex>
        </naptr>
      </records>
    </export>
  </ns6:data>
  <privateSalt>
    D1zzX019nAX0v1MF5jgXN3zjy59OU42ftaLjyGCVyG9zbnk5hyjugqjku
  </privateSalt>
</domain>
```

March 2010

```
</domains>
<zones>
  <zone zoneName="example.tel">
    <nameservers>
      <host>ns1.example.tel</host>
      <host>ns2.example.tel</host>
    </nameservers> </zone> </zones>
  </export>
</exports>]]>
</ns6:data>
</typ:importDataRequest>
```

Response

```
<ns3:importDataResponse
  xmlns:ns3="http://xmlns.telnic.org/ws/nsp/client/exchange/types-
1.0">
  <ns3:domains>
    <ns3:domainName>example.tel</ns3:domainName>
  </ns3:domains>
  <ns3:zones/>
</ns3:importDataResponse>
```