

---

# **TelHosting Admin SOAP API Reference**

July 2009

---



## DISCLAIMER

Copyright (c) 2008-2009, 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

Disclaimer .....	2
1. About this Document .....	5
1.1. Revision History .....	5
2. Overview .....	6
2.1. Groups of API Functions.....	6
3. Key Concepts .....	7
4. Object Model .....	9
5. Message Format .....	10
5.1. Request.....	10
5.2. Response.....	10
6. List of Namespaces .....	10
7. List of Schema Files .....	11
8. Supported Languages .....	11
9. Related Links.....	12
10. User Management .....	13
10.1. User Management.....	13
10.2. createUser.....	14
10.3. createUserAndPassword.....	15
10.4. updateUser .....	17
10.5. deleteUser .....	19
10.6. listUsers .....	20
10.7. getUser .....	22
10.8. initSOCredentials .....	24
11. Name Server Management.....	25
11.1. Name Server Management .....	25
11.2. createNS .....	27
11.3. updateNS .....	29

---

11.4.	deleteNS.....	30
11.5.	getNS.....	31
11.6.	listNSs .....	33
12.	Permissions Management.....	34
12.1.	Permissions Management .....	34
12.2.	createPermissions .....	36
12.3.	updatePermissions.....	37
12.4.	deletePermissions.....	38
12.5.	getPermissions .....	39
12.6.	listPermissions .....	41
12.7.	getPermissionDescriptors .....	42
13.	Partition Management.....	45
13.1.	Partition Management .....	45
13.2.	createPartition .....	46
13.3.	updatePartition.....	49
13.4.	deletePartition.....	52
13.5.	getPartition.....	53
13.6.	listPartitions .....	56
14.	Virtual File Management .....	57
14.1.	Virtual File Management.....	57
14.2.	createFile .....	58
14.3.	updateFile.....	59
14.4.	deleteFile.....	60
14.5.	listFiles .....	61
14.6.	getFile .....	62

## 1. ABOUT THIS DOCUMENT

This document describes the TelHosting Admin SOAP API, which can be used for configuring the TelHosting Software and its interaction with other TelHosting Provider's systems.

The reference is targeted at TelHosting Provider administrators and 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: users, name servers, permissions, partitions and virtual files
- Specific usage requirements for each function

### 1.1. Revision History

Version	Description
1.0	Admin SOAP API reference; initial version.

## 2. 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 Naming Authority 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](#).

### 2.1. Groups of API Functions

The SOAP API is divided into two logical parts: client and admin functions. The client APIs manage domains, zones, resource records, keywords, users, and privacy settings. That part is covered in the Client SOAP API Reference document.

The admin APIs are mostly used by the TelHosting Software system administrator (to set up, configure, and monitor the system) and by its support personnel (to manage customers and, if required, act on their behalf). The most important parts deal with the control of who may use the system (“[User Management](#)”) and to which extent (“[Permissions Interface](#)”). The “[Name Server Management](#)”, on the other hand, allows the TelHosting Provider to alter the use of name servers and the way the distribution of zone data is performed. Registrar companies using the Community TelHosting service do not need to manage name servers.

### 3. 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 <a href="#">Telnic Developer Area</a>.</p>
TelHosting Provider	The organization running the <i>TelHosting Software</i> and operating name servers for the .tel zone. Cf., <i>registrar company</i> .
Registrar company	A company registering domains in the .tel zone; may or may not be the TelHosting Provider, which is irrelevant for the purposes of operating .tel domains.
.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, and storage of keys for record encryption.</p> <p>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.</p>
TelHosting Software	<p>The open-source software reference implementation that supports .tel domain manipulation.</p> <p>The TelHosting Software is a Java application that supports provisioning 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</p>

---

	purposes, such as name server and user management. The client API is out of the scope of this document.
Zone	Coordinates publication of domains. See the <a href="#">Zone</a> object.
User	<p>A user of the system with a certain level of access: administrator or primary user. Administrators and primary users are created by the TelHosting Provider and cannot be created by means of this API.</p> <p>Do not mix users with Readers - external users allowed to read private data.</p>
Partition	Representation of a tenant in a multi-tenant TelHosting Software system; see the <a href="#">Partition</a> object definition and the <a href="#">API overview</a> .
Permissions	<p>User authorization defined by the TelHosting Provider's policy to regulate creation of second- and greater-level domains and zones.</p> <p>These are independent objects that can be assigned to one or multiple users. Permissions are grouped into sets governing domains, zones, users, etc.</p> <p>See <a href="#">Permissions Management</a> for more details on permissions types.</p>

## 4. 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: Admin SOAP API Objects**

Object Name	Description
Zone	<p>Coordinates publication of domains by referring to the <a href="#">Name Server</a> 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.</p> <p>There usually is a one-to-one relationship between Domains and Zones, that is, 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>
User	<p>Represents a user of the system. Depending on the level of access, a user can be an administrator or a primary user. Administrators are technical staff supporting TelHosting Software; primary users are typical users of the system who can create domains and zones.</p>
Name Server	<p>Represents a name server operated by the TelHosting Provider. Since a name server is a concrete resource managed outside of the system, only administrators with specific rights can edit these objects. Other users only have read access to these objects.</p>
Partition	<p>Represents a tenant in a multi-tenant operation mode of TelHosting Software. For instance, multiple registrar companies using the software in Community TelHosting mode would each have a partition. Permission objects are tied to partitions.</p>
Virtual File	<p>Represents files stored in the system's database to enable tenants customize their appearance of the partition's web interface.</p>
Permissions	<p>Is a collection of individual permissions that control the authorizations of a user.</p>

## 5. MESSAGE FORMAT

### 5.1. 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](#).

### 5.2. Response

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

## 6. LIST OF NAMESPACES

Name	API Function Group
<a href="http://xmlns.telnic.org/ws/nsp/admin/user/types-1.0">http://xmlns.telnic.org/ws/nsp/admin/user/types-1.0</a>	<a href="#">User Management</a>
<a href="http://xmlns.telnic.org/ws/nsp/admin/nameserver/types-1.0">http://xmlns.telnic.org/ws/nsp/admin/nameserver/types-1.0</a>	<a href="#">Name Server Management</a>
<a href="http://xmlns.telnic.org/ws/nsp/admin/partition/types-1.0">http://xmlns.telnic.org/ws/nsp/admin/partition/types-1.0</a>	<a href="#">Partition Management</a>
<a href="http://xmlns.telnic.org/ws/nsp/admin/permissions/types-1.0">http://xmlns.telnic.org/ws/nsp/admin/permissions/types-1.0</a>	<a href="#">Permissions Management</a>
<a href="http://xmlns.telnic.org/ws/nsp/admin/virtualfile/types-1.0">http://xmlns.telnic.org/ws/nsp/admin/virtualfile/types-1.0</a>	<a href="#">Virtual File Management</a>

## 7. 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 WSDL file representing the whole API is [Admin-Service-1.0.wsdl](#), and the types file [Admin-1.0.wsdl](#).

Name	API Function Group
<a href="#">User-1.0.xsd</a>	<a href="#">User Management</a>
<a href="#">NameServer-1.0.xsd</a>	<a href="#">Name Server Management</a>
<a href="#">Permissions-1.0.xsd</a>	<a href="#">Permissions Management</a>
<a href="#">Partition-1.0.xsd</a>	<a href="#">Partitions Management</a>
<a href="#">VirtualFile-1.0.xsd</a>	<a href="#">Virtual File Management</a>

## 8. SUPPORTED LANGUAGES

The TelHosting Software provides multilingual support in the following aspects:

- UTF-8 encoding for the text header and keywords
- Web interface and email templates in a number of languages

Currently, the following languages are supported, each with its own 2-letter code according to ISO-639-1 standard:

Language	Code	Language	Code
Arabic	ar	Italian	it
Chinese	zh	Japanese	ja
English	en	Korean	ko
French	fr	Portuguese	pt
German	de	Russian	ru
Spanish	es		

## 9. 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/docs/devguide.pdf>
- TelHosting Client SOAP API, <http://dev.telnic.org/api/client-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/>

## 10. USER MANAGEMENT

### 10.1. User Management

This interface is responsible for managing users of the TelHosting Software system. A summary of the API operations is shown in Table 3. The corresponding schema definition can be found in the file [User-1.0.xsd](#)

**Table 3: Summary of user management API**

Operation	Description
<a href="#">createUser</a>	Creates a new user in the system.
<a href="#">updateUser</a>	Enables or disables a user, (re-) assigns a <a href="#">Permissions</a> object to the user, thereby changing the rights of the user. A lowering of rights is handled in the same way as in <a href="#">updatePermissions</a> .
<a href="#">deleteUser</a>	Deletes a user in the system.
<a href="#">listUsers</a>	Returns a (partial) list of users known to the system.
<a href="#">getUser</a>	Gets the specific properties of the user.
<a href="#">initSOCredentials</a>	Allows automatic initialization of a client embedded friending client.

## 10.2. createUser

This operation allows an administrator to create a user. The usual restrictions on user name and password apply. The administrator executing the operation is responsible for communicating the credentials to the user. Unlike [createUserAndPassword](#), this operation does not send an automatic email to the registrant.

### **createUserRequest**

Element Name	Description
username	Designated user name.
partition	<i>optional</i> Partition to which the user will belong. If not given, the administrator's partition is used.
type	Type of the user. Allowed values: admin, user-admin or primary-user.
source	<i>optional</i> Source of the user identity, for external authentication. <b>Sub-elements:</b> <ul style="list-style-type: none"> <li>name - string identifying the source</li> <li>ref - <i>optional</i> reference that unequivocally identifies the user at the source</li> </ul>
permissions	Name of the permissions assigned to the user.
password	<i>optional</i> Initial password of the user. If the external authentication is used, the set password will be ignored.
emailAddress	User's email address.

### **createUserResponse**

No elements.

### **Example**

Create the primary user 'regina' with password 'notlob'.

### **Request**

```
<typ:createUserRequest>
  <typ:userName>regina</typ:userName>
  <typ:type>primary-user</typ:type>
  <typ:permissions>default-primary</typ:permissions>
  <typ:password>notlob</typ:password>
</typ:createUserRequest>
```

---

## Response

```
<createUserResponse
xmlns="http://xmlns.telnic.org/ws/nsp/admin/user/types-1.0"/>
```

### 10.3. createUserAndPassword

This operation allows an administrator to create a user and get the password generated automatically by the system. The usual restrictions on user name and password apply. The usual restrictions on user name and password apply. The system will send an automated email to the registrant with their username and generated password from the email address specified as the sender address in the Partition settings.

#### **createUserAndPasswordRequest**

Element Name	Description
username	Designated user name.
partition	<i>optional</i> Partition to which the user will belong. If not given, the administrator's partition is used.
type	Type of the user. Allowed values: admin, user-admin or primary-user.
permissions	Name of the permissions assigned to the user.
emailAddress	User's email address.
languageCode	<i>optional</i> Two-letter code indicating the language to be used in the sent e-mail; if none is given the language will be English (see full list of supported languages)

#### **createUserResponse**

Element Name	Description
password	The generated initial password of the user.

#### **Example**

Create the primary user 'regina' with password 'notlob348'.

#### **Request**

```
<typ:createUserRequest>
  <typ:userName>regina</typ:userName>
  <typ:type>primary-user</typ:type>
  <typ:permissions>default-primary</typ:permissions>
  <typ:emailAddress>regina@gmail.com</typ:emailAddress>
</typ:createUserRequest>
```

#### **Response**

```
<ns6:createUserAndPasswordResponse
  xmlns:ns6="http://xmlns.telnic.org/ws/nsp/admin/user/types-1.0">
  <ns6:password>notlob348</ns6:password>
</ns6:createUserAndPasswordResponse>
```

## 10.4. updateUser

This operation allows an administrator to modify a user.

### **updateUserRequest**

Element Name	Description
username	Designated user name.
partition	<i>optional</i> Partition to which the user belongs. This element clarifies the partition on which the user being updated resides. Thus allowing an administrator in a parent partition to update users in child partitions. It does not allow a user's partition to be changed.
newUserName	<i>optional</i> New name of the user, if given. When the new user name is set, the password is cleared unless a new password is given.
source	<i>optional</i> Source of the user identity, for external authentication. <b>Sub-elements:</b> <ul style="list-style-type: none"><li>• name - string identifying the source</li><li>• ref - <i>optional</i> reference that unequivocally identifies the user at the source</li></ul>
permissions	<i>optional</i> Name of the permissions assigned to the user.
password	<i>optional</i> New password of the user. If the external authentication is used, the set password will be ignored.
emailAddress	<i>optional</i> User's e-mail address

### **updateUserResponse**

No elements.

### **Example**

Change user name 'regina' to 'reggie' with new permissions, password and e-mail address.

### **Request**

```
<typ:updateUserRequest>  
  <typ:userName>regina</typ:userName>  
  <typ:newUserName>reggie</typ:newUserName>  
  <typ:permissions>premium-customer</typ:permissions>  
  <typ:password>notlob</typ:password>  
  <typ:emailAddress>reggie@example.com</typ:emailAddress>
```

</typ:updateUserRequest>

---

## Response

---

```
<updateUserResponse  
xmlns="http://xmlns.telnic.org/ws/nsp/admin/user/types-1.0"/>
```

## 10.5. deleteUser

This operation allows an administrator to delete a user.

### ***deleteUserRequest***

Element Name	Description
username	Name of the user to delete.
partition	<i>optional</i> Partition to which the user belongs. If not given, the administrator's partition is used.

### ***deleteUserResponse***

No elements.

### ***Example***

Delete user 'regina'.

### **Request**

```
<typ:deleteUserRequest>  
  <typ:userName>regina</typ:userName>  
</typ:deleteUserRequest>
```

### **Response**

```
<deleteUserResponse  
xmlns="http://xmlns.telnic.org/ws/nsp/admin/user/types-1.0"/>
```

## 10.6. listUsers

This operation allows an administrator to list users. The results can be restricted in various ways, e.g., by partition or by user name.

### *listUsersRequest*

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

limit

*optional* Parameter limiting the returned users according to specified sub-elements.

#### **Sub-elements:**

- `userNameRange` - *optional* Configures the list to include users ordered by name and limited to the range specified via the attributes of this sub-element:
  - `minInclusive/minExclusive` - the lower bound of the search
  - `maxInclusive/maxExclusive` - the upper bound of the search

The difference between the two variants is that the `inclusive` versions return a user of the exact name given in the value, while the `exclusive` versions do not. If more than one attribute is present, the most restrictive one applies.

#### Notes:

- A previous search can be continued if the value of the last returned user is specified in the attribute `minExclusive`.
- An empty string is lexicographically sorted into the first place of the user name list. Thus, if an empty string is present in a `max` attribute, an empty list of users will be returned.
- `partition` - *optional* the partition, for which to list users; if not specified, the admin's partition is used.
- `maxUsers` - *optional* the upper limit on the number of returned entries
- `type` - *optional* the type of users to list (administrator or

primary)

### ***listUsersResponse***

A list of users matching the request query.

#### ***Example 1***

Get five users from the list. Such queries may be especially convenient for mobile clients that need to query and process limited amounts of data.

#### **Request**

```
<typ:listUsersRequest>
  <typ:limit>
    <typ:maxUsers>4</typ:maxUsers>
  </typ:limit>
</typ:listUsersRequest>
```

#### **Response**

```
<ns6:listUsersResponse
  xmlns:ns6="http://xmlns.telnic.org/ws/nsp/admin/user/types-1.0">
  <ns6:user type="admin">administrator</ns6:user>
  <ns6:user type="primary-user">regina</ns6:user>
  <ns6:user type="primary-user">marvin</ns6:user>
  <ns6:user type="primary-user">dirkg</ns6:user>
</ns6:listUsersResponse>
```

#### ***Example 2***

Get primary users with names including "reg" to "reh".

#### **Request**

```
<typ:listUsersRequest>
  <typ:limit>
    <typ:userNameRange minInclusive="reg" maxExclusive="reh"/>
    <typ:type>primary-user</typ:type>
  </typ:limit>
</typ:listUsersRequest>
```

#### **Response**

```
<ns6:listUsersResponse
  xmlns:ns6="http://xmlns.telnic.org/ws/nsp/admin/user/types-1.0">
  <ns6:user type="primary-user">reggie</ns6:user>
  <ns6:user type="primary-user">regina</ns6:user>
  <ns6:user type="primary-user">reginald</ns6:user>
</ns6:listUsersResponse>
```

## 10.7. getUser

This operation allows an administrator to retrieve information about a particular user.

### ***getUserRequest***

Element Name	Description
username	User name.
partition	<i>optional</i> Partition to which the user belongs. If not given, the administrator's partition is used.

### ***getUserResponse***

The structure of the response can be seen from the example below, for details please consult the file User-1.0.xsd.

Element Name	Description
partition	Partition to which the user belongs.
type	Type of the user. Allowed values: admin, user-admin or primary-user.
source	Source of the user identity, for external authentication. If no information is given, an empty element is returned.
permissions	Name of the permissions assigned to the user.
password	Boolean value: <ul style="list-style-type: none"> <li>true - a local password is assigned to the user</li> <li>false - the user can log in only via external authentication</li> </ul>
emailAddress	<i>optional</i> User's e-mail address.

### ***Example***

Get the user 'regina' from partition 'verySpecialPeople'.

### **Request**

```
<typ:getUserRequest>
  <typ:userName>regina</typ:userName>
  <typ:partition>verySpecialPeople</typ:partition>
</typ:getUserRequest>
```

### **Response**

```
<ns6:getUserResponse
  xmlns:ns6="http://xmlns.telnic.org/ws/nsp/admin/user/types-1.0">
```

July 2009

---

```
<ns6:partition>verySpecialPeople</ns6:partition>  
<ns6:type>primary-user</ns6:type>  
<ns6:source/>  
<ns6:permissions>special</ns6:permissions>  
<ns6:password>true</ns6:password>  
<ns6:emailAddress>reggie@example.com</ns6:emailAddress>  
</ns6:getUserResponse>
```

## 10.8. initSOCredentials

This operation initializes a user's Embedded Friending Client automatically instead of making the user do it manually. To perform this operation, the user's SO credentials (SO user name, web password and the answer to the SO challenge question) need to be specified. In other words, a client application can use this function to initialize an SO account for the given TelHosting user via the TelHosting Software interface, instead of using the SO interface.

### *initSOCredentialsRequest*

Element Name	Description
username	Name of the user for who the SO credentials will be initialized.
partition	<i>optional</i> Partition to which the user belongs. If not given, the administrator's partition is used.
soUserName	User name of the SO account associated with this TelHosting user.
soPassword	Web password of the SO account associated with this TelHosting user.
soChallengeAnswer	Answer to the SO challenge question of the account associated with this TelHosting user.

### *initSOCredentialsResponse*

No elements.

### **Example**

Get the user 'regina' from partition 'verySpecialPeople'.

### **Request**

```
<typ:initSOCredentialsRequest>
  <typ:userName>johnsmith</typ:userName>
  <typ:partition>root</typ:partition>
  <typ:soUserName>mysoname</typ:soUserName>
  <typ:soPassword>topsecret</typ:soPassword>
  <typ:soChallengeAnswer>My challenge answer</typ:soChallengeAnswer>
</typ:initSOCredentialsRequest>
```

### **Response**

```
<initSOCredentialsResponse
  xmlns="http://xmlns.telnic.org/ws/nsp/admin/user/types-1.0"/>
```

## 11. NAME SERVER MANAGEMENT

### 11.1. Name Server Management

This API allows the TelHosting Provider to manage the name servers that the TelHosting Software is using. For each name server, the system stores the configuration data shown in Table 4. A summary of the API operations is shown in Table 5. The corresponding schema definition can be found in the file [NameServer-1.0.xsd](#).

Note that you cannot change the domain name of a server. However, you can delete a name server and create a new one.

**Table 4: Configuration data stored for each name server**

Configuration data	Description
nsLocationGroup	Groups name servers sharing the same location within the Internet network topology. If a zone is created or updated, the system verifies that no two name servers share the same location group to ensure network topological diversity. This applies to both manually and automatically assigned name servers.
availability	The availability of the name server.
relativePerformance	Compares DNS performance of a name server compared to the other servers to enable equal distribution of domains and zones among the servers.
maxLoad	Sets the maximum load factor. When the loading exceeds this parameter, the server is no longer used for automatic assignment.
loaders	Lists the loaders used for the distribution of the zones assigned to the name server.

Load factors are used to determine how much data a name server is assigned to. To calculate a load factor, the server takes the following values into account:

- the number of zones hosted
- the number of domains hosted
- the number of records hosted

**Table 5: Summary of name server management API**

Operation	Description
<a href="#">createNS</a>	Creates a name server (in the system only).
<a href="#">updateNS</a>	Updates a name server.

<a href="#">deleteNS</a>	Deletes a name server.
<a href="#">getNS</a>	Retrieves information about a name server.
<a href="#">listNSs</a>	Lists name servers.

## 11.2. createNS

This operation creates a new name server object.

### ***createNSRequest***

Element Name	Description
nsName	string Name of the name server to create.
locationGroup	string The name of the group of name servers that share the same network topological location; this is used to prevent the automatic assignment of multiple name servers at the same location.
availability	enabled   disallowAutomaticAssignments   disallowNewAssignments   disabled Status of the server.
relativePerformanceint	Relative performance of this name server compared to other name servers. It is used in conjunction with the name server's load to decide which name server to use for automatic assignments. A value of "100" is treated as the neutral value.
maxLoad	int Maximum load of the server to accept automatic assignments. A value of "0" means "don't care".

### ***createNSResponse***

No elements.

### ***Example***

Create name server ns1.example.com.

### **Request**

```
<typ:createNSRequest>
  <typ:nsName>ns1.example.com</typ:nsName>
  <typ:locationGroup>basement, main building,
  Egtown</typ:locationGroup>
  <typ:availability>enabled</typ:availability>
  <typ:relativePerformance>100</typ:relativePerformance>
  <typ:maxLoad>0</typ:maxLoad>
</typ:createNSRequest>
```

### **Response**

```
<createNSResponse
```

---

xmlns="http://xmlns.telnic.org/ws/nsp/admin/nameserver/types-1.0"/>

### 11.3. updateNS

This operation updates a name server object.

#### *updateNSRequest*

Element Name	Description
nsName	string Name of the name server to update.
locationGroup	string <i>optional</i> The name of the group of name servers that share the same network topological location.
availability	enabled   disallowAutomaticAssignments   disallowNewAssignments   disabled <i>optional</i> Status of the server.
relativePerformanceInt	<i>optional</i> Relative performance of this name server compared to other name servers.
maxLoad	int <i>optional</i> Maximum load of the server to accept automatic assignments.

#### *updateNSResponse*

No elements.

#### **Example**

Update name server ns1.example.com to prohibit automatic assignments to this server and set new relative performance and maximum load parameters.

#### **Request**

```
<typ:updateNSRequest>
  <typ:nsName>ns1.example.com</typ:nsName>
  <typ:availability>disallowAutomaticAssignments</typ:availability>
  <typ:relativePerformance>150</typ:relativePerformance>
  <typ:maxLoad>42</typ:maxLoad>
</typ:updateNSRequest>
```

#### **Response**

```
<updateNSResponse
xmlns="http://xmlns.telnic.org/ws/nsp/admin/nameserver/types-1.0"/>
```

## 11.4. deleteNS

This operation deletes a name server object.

### *deleteNSRequest*

Element Name	Description
nsName	string Name of the name server to delete.

### *deleteNSResponse*

No elements.

### *Example*

Delete name server ns1.example.com.

### **Request**

```
<typ:deleteNSRequest>  
  <typ:nsName>ns1.example.com</typ:nsName>  
</typ:deleteNSRequest>
```

### **Response**

```
<deleteNSResponse  
xmlns="http://xmlns.telnic.org/ws/nsp/admin/nameserver/types-1.0"/>
```

## 11.5. getNS

This operation retrieves the configuration data for a name server.

### *getNSRequest*

Element Name	Description
nsName	string Name of the name server to get.

### *getNSResponse*

Element Name	Description
locationGroup	string The name of the group of name servers that share the same network topological location.
availability	enabled   disallowAutomaticAssignments   disallowNewAssignments   disabled Status of the server.
relativePerformanceint	Relative performance of this name server compared to other name servers.
maxLoad	int Maximum load of the server to accept automatic assignments.
currentLoad	int Current load of the server. Use this value to check for unbalanced loads and temporarily prohibit automatic assignment of new zones to this name server.

### **Example**

Get name server ns1.example.com.

### **Request**

```
<typ:getNSRequest>  
  <typ:nsName>ns1.example.com</typ:nsName>  
</typ:getNSRequest>
```

### **Response**

```
<getNSResponse
```

July 2009

---

```
xmlns="http://xmlns.telnic.org/ws/nsp/admin/nameserver/types-1.0">
  <locationGroup>basement, main building, Egtown</locationGroup>
  <availability>disallowAutomaticAssignments</availability>
  <relativePerformance>150</relativePerformance>
  <maxLoad>42</maxLoad>
  <currentLoad>17</currentLoad>
</getNSResponse>
```

## 11.6. listNSs

This operation lists the name servers.

### *listNSsRequest*

No elements.

### *listNSsResponse*

Element Name	Description
ns	Name of the name server in the list.

### *Example*

#### **Request**

```
<typ:listNSsRequest/>
```

#### **Response**

```
<listNSsResponse  
  xmlns="http://xmlns.telnic.org/ws/nsp/admin/nameserver/types-1.0">  
  <ns nsName="ns1.example.com"/>  
  <ns nsName="ns2.example.com"/>  
  <ns nsName="ns3.example.com"/>  
  <ns nsName="ns4.example.com"/>  
  <ns nsName="ns5.example.com"/>  
  <ns nsName="ns6.example.com"/>  
</listNSsResponse>
```

## 12. PERMISSIONS MANAGEMENT

### 12.1. Permissions Management

The Permissions interface allows the creation, modification, and deletion of Permissions in order to allow a remote management of the system. A summary of the API operations is shown in Table 6. The corresponding schema definition can be found in the file [Permissions-1.0.xsd](#).

With the help of permissions, the administrator of the system is able to limit the capabilities of individual users and define special settings. In general, the system groups individual permissions into [Permissions](#) objects, that may be referenced by names previously assigned to them. Each user, be it an administrator or a primary user, refers to exactly one Permissions object.

As the supported permissions may change over the lifetime of the software, the individual permissions are not hard-coded into the SOAP interface, which would require the adaption of the SOAP interface for each change. Instead, a generalized access method to the permissions has been created. To allow the user of this interface to discover the available permissions and the accepted values, a special operation, [getPermissionDescriptors](#), is provided.

#### **Maximum and Minimum Permissions**

All Permissions objects are associated with [Partitions](#) to enable multi-tenant capabilities of TelHosting Software.

To allow greater flexibility with permissions, a minimum and a maximum set of permissions can be created for each type of user. Those sets need to belong to the parent of the given partition so that administrators cannot edit permissions for their own partitions. The interpretation of “minimum” and “maximum” depends on the individual permission. An undefined permission does not alter the original permission. Unlimited values are treated according to the respective logic.

#### **Permissions for Users**

Permissions generally regulate creation of second- and greater-level domains and zones and their delegation to other 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. However, the Provider's 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.

**Table 6: Summary of permissions management API**

Operation	Description
<a href="#">createPermissions</a>	Create a new permissions object.
<a href="#">updatePermissions</a>	Update a permissions object, i.e. alter various rights. Changes have effect on all users that are assigned to this Permissions object. If the rights are reduced (e.g. number of sub-domains is lowered), this does not have any effect on the status quo. The system has to gracefully handle the excess

and should not reject operations that maintain or lower the level of excess (e.g. creating resource records for such sub-domains). If the TelHosting Provider wants to enforce the lowering of the rights, he has to take care of it by using the API to change the user's data until it meets the limits.

[deletePermissions](#)

Delete a permissions object, which is only possible if no users are linked to the object.

[getPermissions](#)

Retrieve the content of a permissions object.

[listPermissions](#)

Request a list of all permissions objects.

[getPermissionDescriptors](#) Request information about the individual permissions.

## 12.2. createPermissions

This operation creates a new permissions, or permission set, object.

### ***createPermissionsRequest***

Element Name	Description
name	string Name of the new permissions objects.
partition	string <i>optional</i> Partition to which the permissions belong; if omitted, the partition to which the calling user belongs is used.
permission	See <a href="#">getPermissionDescriptors</a> for allowed values and available permissions. <i>optional</i> A single permission of the permission set to create. Any number of <permission> elements is allowed. The permission is identified by the name attribute and has a varying number of elements, depending on what kind of permission is being defined.

### ***createPermissionsResponse***

No elements.

### ***Example***

#### **Request**

```
<typ:createPermissionsRequest>
  <typ:name>Very Special Permissions No. 1</typ:name>
  <typ:partition>verySpecialPeople</typ:partition>
  <typ:permission name="perm.usertypes">
    <typ:string>primary</typ:string>
    <typ:string>admin</typ:string>
  </typ:permission>
  <typ:permission name="user.domain.max">
    <typ:unrestricted/>
  </typ:permission>
  <typ:permission name="partition.subpartitions">
    <typ:boolean>true</typ:boolean>
  </typ:permission>
</typ:createPermissionsRequest>
```

#### **Response**

```
<createPermissionsResponse
xmlns="http://xmlns.telnic.org/ws/nsp/admin/permissions/types-1.0"/>
```

### 12.3. updatePermissions

This operation allows an administrator to modify a Permissions object.

#### ***updatePermissionsRequest***

Element Name	Description
name	string Name of the permissions objects.
partition	string <i>optional</i> Partition to which the permissions belong; if omitted, the partition to which the calling user belongs is used.
newName	string New public name of the permissions set.
permission	<i>optional</i> A single permission of the set to update. Any number of <permission> elements is allowed. The permission is identified by the name attribute and has a varying number of elements, depending on what kind of permission is being defined. See <a href="#">getPermissionDescriptors</a> for allowed values and available permissions.

#### ***updatePermissionsResponse***

No elements.

#### ***Example***

Request:

#### **Request**

```
<typ:updatePermissionsRequest>
  <typ:name>Very Special Permissions No. 1</typ:name>
  <typ:partition>verySpecialPeople</typ:partition>
  <typ:newName>Special Permissions No. 1</typ:newName>
  <typ:permission name="perm.usertypes">
    <typ:string>primary</typ:string>
  </typ:permission>
  <typ:permission name="user.domain.max">
    <typ:number>42</typ:number>
  </typ:permission>
  <typ:permission name="partition.subpartitions">
    <typ:boolean>>false</typ:boolean>
  </typ:permission>
</typ:updatePermissionsRequest>
```

#### **Response**

```
<updatePermissionsResponse
xmlns="http://xmlns.telnic.org/ws/nsp/admin/permissions/types-1.0"/>
```

## 12.4. deletePermissions

This operation deletes a [Permissions](#) object.

You cannot delete a Permissions object that is in use, that is, has users associated with it. To check whether a Permissions object is in use, use the [getPermissionsRequest](#) operation, the element `inUse`.

### ***deletePermissionsRequest***

Element Name	Description
<code>name</code>	string Name of the Permissions object to be deleted.
<code>partition</code>	string <i>optional</i> Partition to which the permissions belong; if omitted, the partition to which the calling user belongs is used.

### ***deletePermissionsResponse***

No elements.

#### ***Example***

Delete the set of permissions for partition 'verySpecialPeople'.

#### **Request**

```
<typ:deletePermissionsRequest>  
  <typ:name>Special Permissions No. 1</typ:name>  
  <typ:partition>verySpecialPeople</typ:partition>  
</typ:deletePermissionsRequest>
```

#### **Response**

```
<deletePermissionsResponse  
xmlns="http://xmlns.telnic.org/ws/nsp/admin/permissions/types-1.0"/>
```

## 12.5. getPermissions

This operation retrieves the data of the single permissions within a permission set.

### ***getPermissionsRequest***

Element Name	Description
name	string Name of the Permissions object to probe.
partition	string <i>optional</i> Partition to which the permissions belong; if omitted, the partition to which the calling user belongs is used.

### ***getPermissionsResponse***

Element Name	Description
name	string Name of the permissions objects.
inUse	Boolean Indicates whether any users are associated with the given Permissions object.
permission	See <a href="#">getPermissionDescriptors</a> for allowed values and available permissions. A single permission of the permission set. Any number of <permission> elements is allowed. The permission is identified by the name attribute and has a varying number of elements, depending on what kind of permission is being defined.

### ***Example***

Get information on the permissions set 'verySpecialPeople'.

### **Request**

```
<typ:getPermissionsRequest>  
  <typ:name>Very Special Permissions No. 1</typ:name>  
  <typ:partition>verySpecialPeople</typ:partition>  
</typ:getPermissionsRequest>
```

### **Response**

```
<ns4:getPermissionsResponse
```

```
xmlns:ns4="http://xmlns.telnic.org/ws/nsp/admin/permissions/types-1.0">
<ns4:name>Very Special Permissions No. 1</ns4:name>
<ns4:inUse>true</ns4:inUse>
<ns4:permission name="perm.usertypes">
  <ns4:string>primary</ns4:string>
</ns4:permission>
<ns4:permission name="user.domain.max">
  <ns4:number>42</ns4:number>
</ns4:permission>
<ns4:permission name="partition.subpartitions">
  <ns4:boolean>>false</ns4:boolean>
</ns4:permission>
</ns4:getPermissionsResponse>
```

## 12.6. listPermissions

This operation lists the partition objects.

### *listPermissionsRequest*

Element Name	Description
partition	string <i>optional</i> Partition to which the permission set belongs; if omitted, the partition to which the calling user belongs is used.

### *listPermissionsResponse*

List of permissions matching the request.

#### **Example**

Get permissions in the partition 'verySpecialPeople'.

#### **Request**

```
<typ:listPermissionsRequest>  
  <typ:partition>verySpecialPeople</typ:partition>  
</typ:listPermissionsRequest>
```

#### **Response**

```
<ns4:listPermissionsResponse  
  xmlns:ns4="http://xmlns.telnic.org/ws/nsp/admin/permissions/types-  
  1.0">  
  <ns4:permissions name="Default Permissions"/>  
  <ns4:permissions name="Special Permissions"/>  
  <ns4:permissions name="Very Special Permissions No. 1"/>  
</ns4:listPermissionsResponse>
```

## 12.7. getPermissionDescriptors

The operation returns a list of permissions, along with metadata describing the type and accepted values.

### ***getPermissionDescriptorsRequest***

No elements.

### ***getPermissionDescriptorsResponse***

For each permission, the following data is returned:

Attribute Name	Description
name	Public name of the permissions object.
baseType	Underlying data type. Allowed values: <code>boolean</code>   <code>number</code>   <code>string</code>
compositeType	Cardinality of values. Possible values are <code>single</code> , indicating that the permission expects exactly one single value, or <code>set</code> , where an arbitrary number of distinct values may be given.
unrestrictedAllowed	Specifies whether the special value "unrestricted" is supported for the permission. Allowed values: <code>true</code> or <code>false</code> .  This attribute can be handy for such permissions as <code>user.domain.max: unrestrictedAllowed="true"</code> would mean that no maximum limit is set to the value, so the user can have any number of domains.

Element Name	Description
minimum	<i>for numeric permissions only</i> Minimum allowed permission's value, inclusive.
maximum	<i>for numeric permissions only</i> Maximum allowed permission's value, inclusive.
values	Possible values for the permission, numeric or textual. The <code>exclusive</code> attribute has the following values: <ul style="list-style-type: none"> <li><code>true</code> - the given values are exclusive, i.e. no other values may be used besides the given ones</li> <li><code>false</code> - the values are a suggestion, and different values can also be used</li> </ul>

### ***Example***

An excerpt of the permissions given in the response.

---

**Request**

---

```
<typ:getPermissionDescriptorsRequest/>
```

---

**Response**

---

```
<getPermissionDescriptorsResponse
  xmlns="http://xmlns.telnic.org/ws/nsp/admin/permissions/types-1.0">
  <!--... -->
  <permission baseType="number" compositeType="single"
name="user.zone.max"
  unrestrictedAllowed="true">
    <minimum>
      <number>0</number>
    </minimum>
  </permission>
  <permission baseType="number" compositeType="single"
name="user.domain.max"
  unrestrictedAllowed="true">
    <minimum>
      <number>0</number>
    </minimum>
  </permission>
  <permission baseType="number" compositeType="single"
name="zone.label.max"
  unrestrictedAllowed="true">
    <minimum>
      <number>2</number>
    </minimum>
  </permission>
  <permission baseType="number" compositeType="single"
name="zone.ns.min"
  unrestrictedAllowed="false">
    <minimum>
      <number>0</number>
    </minimum>
    <maximum>
      <number>13</number>
    </maximum>
  </permission>
  <permission baseType="string" compositeType="set"
  name="domain.search.kw.name" unrestrictedAllowed="false"/>
  <permission baseType="string" compositeType="set"
  name="domain.record.type" unrestrictedAllowed="false">
    <values exclusive="true">
      <string>MX</string>
      <string>Generic</string>
      <string>NAPTR</string>
      <string>ZS</string>
      <string>SRV</string>
      <string>TXT</string>
      <string>LOC</string>
    </values>
  </permission>
  <permission baseType="boolean" compositeType="single"
name="zone.ns.manualassign" unrestrictedAllowed="false"/>
  <!--... -->
```

---

</getPermissionDescriptorsResponse>

## 13. PARTITION MANAGEMENT

### 13.1. Partition Management

The partitions interface allows the creation, modification, and deletion of partitions in order to allow a remote management of the system.

Partitions are the key to the *multi-tenant mode* of operating TelHosting Software, where multiple TelHosting Providers can co-exist. Each Provider can have a customized web interface and a separate environment inside the system. This is done by means of a [Partition](#) object, and [users](#) and [permission sets](#) are part of the respective partition. They also form a separate namespace, which means that in different partitions permissions with the same name may exist. The user and his permission must be located within the same partition, it is not possible to cross-reference permissions between partitions. The root partition has no parent and thus cannot have any permissions.

A summary of the API operations of this group is shown in Table 7. The corresponding schema definition can be found in the file [Partition-1.0.xsd](#).

Note that the names of partitions are case-insensitive and may only contain letters and digits.

**Table 7: Summary of partition management API**

Operation	Description
<a href="#">createPartition</a>	Creates a new partition object.
<a href="#">updatePartition</a>	Updates a partition object.
<a href="#">deletePartition</a>	Deletes a partition object.
<a href="#">getPartition</a>	Retrieves the content of a partition object.
<a href="#">listPartitions</a>	Requests a list of all partition objects.

## 13.2. createPartition

This operation creates a new Partition object.

### ***createPartitionRequest***

Element Name	Description
name	Name of the partition to be created.
parent	<i>optional</i> Parent partition. If not given, the administrator's partition is used.
minPermissions	<i>optional</i> Permissions that will be applied as the minimum permissions of all users of the partition and subordinate partitions. The referenced permissions must be defined in the parent partition.
maxPermissions	<i>optional</i> Permissions that will be applied as the maximum permissions of all users of the partition and subordinate partitions. The referenced permissions must be defined in the parent partition.
addressSelector	Address selectors identifying the given partition from the connection that the client uses. They are defined as a sequence of <code>selector</code> elements with the following <b>subelements</b> : <ul style="list-style-type: none"> <li>• <code>service</code> - <i>optional</i> Service type (<code>soap</code> or <code>web</code>); if omitted, the selector is used for all service types.</li> <li>• <code>transport</code> - <i>optional</i> Transport type (<code>HTTP</code>, <code>HTTPS</code>, or <code>both</code>); if omitted, the selector is used for all transport types.</li> <li>• <code>virtualHostName</code> - <i>optional</i> Virtual name that the selector will match, i.e. the domain name given in the "host" HTTP header field. If omitted, all names are matched.</li> <li>• <code>localHostName</code> - <i>optional</i> Local interface name the selector will match. If omitted, all names are matched.</li> </ul>
configuration	<i>optional</i> Partition-specific configuration. If omitted, the configuration is initialized with empty data, which might cause some functionality to fail. <p><b>Subelements:</b></p> <ul style="list-style-type: none"> <li>• <code>webBaseUrlHttp</code> - Public URL base (<code>HTTP</code>) of the web interface; if no HTTP access is provided, leave empty</li> <li>• <code>webBaseUrlHttps</code> - Public URL base (<code>HTTPS</code>) of the</li> </ul>

web interface; if no HTTPS access is provided, leave empty

- `soapBaseUrlHttp` - Public URL base (HTTP) of the SOAP interface; if no HTTP access is provided, leave empty
- `soapBaseUrlHttps` - Public URL base (HTTPS) of the SOAP interface; if no HTTPS access is provided, leave empty
- `senderAddress` - E-mail address used by the system for outgoing e-mails
- `bccAddresses` - Comma-separated list of e-mail addresses that receive a blind copy of outgoing e-mails
- `operator` - Details about the operator in the subelements
  - `companyName` - the name of the company
  - `supportEMailAddress` - e-mail address of the operator's support
  - `supportPhone` - the phone number of the operator's support
- `googleMaps` - *optional* Details for the embedded Google Maps display. `httpKey` and `httpsKey` may be left blank. In this case, no Google Maps display is provided for the respective protocol.
  - `httpKey` - API key for the use of the Google Maps if the web interface is accessed via the HTTP protocol
  - `httpsKey` - API key if the HTTPS protocol is used
  - `initialLatitude` - latitude of the point that is initially displayed
  - `initialLongitude` - longitude of the point that is initially displayed.

### ***createPartitionResponse***

No elements.

### ***Example***

#### **Request**

```
<typ:createPartitionRequest>  
<typ:name>testpartition</typ:name>
```

July 2009

---

```
<typ:minPermissions>default-secondary</typ:minPermissions>
<typ:maxPermissions>default-primary</typ:maxPermissions>
<typ:addressSelectors>
  <typ:selector>
    <typ:virtualHostName>http://www.example.com</typ:virtualHostName>
  </typ:selector>
</typ:addressSelectors>
<typ:configuration>
  <typ:webBaseUrlHttp>http://www.example.com</typ:webBaseUrlHttp>
  <typ:webBaseUrlHttps>https://www.example.com</typ:webBaseUrlHttps>
  <typ:soapBaseUrlHttp>http://soap.example.com</typ:soapBaseUrlHttp>
  <typ:soapBaseUrlHttps>https://soap.example.com</typ:soapBaseUrlHttps>
  <typ:senderAddress>ceo@example.com</typ:senderAddress>
  <typ:bccAddresses> worker@example.com, janitor@example.com
</typ:bccAddresses>
  <typ:operator>
    <typ:companyName>Example Ltd.</typ:companyName>
    <typ:supportEmailAddress>support@example.com</typ:supportEmailAddress>
    <typ:supportPhone>+42.123456789</typ:supportPhone>
  </typ:operator>
</typ:configuration>
</typ:createPartitionRequest>
```

---

## Response

---

```
<createPartitionResponse
  xmlns="http://xmlns.telnic.org/ws/nsp/admin/partition/types-1.0"/>
```

### 13.3. updatePartition

This operation allows an administrator to modify a partition object.

#### *updatePartitionRequest*

Element Name	Description
name	Name of the partition to be updated.
newName	<i>optional</i> New name of the partition.
minPermissions	<i>optional</i> Permissions that will be applied as the minimum permissions of all users of the partition and subordinate partitions. The referenced permissions must be defined in the parent partition.
maxPermissions	<i>optional</i> Permissions that will be applied as the maximum permissions of all users of the partition and subordinate partitions. The referenced permissions must be defined in the parent partition.
addressSelector	<p><i>optional</i> Address selectors identifying the given partition from the connection that the client uses. They are defined as a sequence of <code>selector</code> elements with the following <b>sub-elements</b>:</p> <ul style="list-style-type: none"> <li><code>service</code> - <i>optional</i> Service type; if omitted, the selector is used for all service types.</li> <li><code>transport</code> - <i>optional</i> Transport type; if omitted, the selector is used for all transport types.</li> <li><code>virtualHostName</code> - <i>optional</i> Virtual name that the selector will match, i.e. the domain name given in the "host" HTTP header field. If omitted, all names are matched.</li> <li><code>localHostName</code> - <i>optional</i> Local interface name the selector will match. If omitted, all names are matched.</li> </ul>
configuration	<p><i>optional</i> Partition-specific configuration. If omitted, the configuration is initialized with empty data, which might cause some functionality to fail. Note that all sub-elements except <code>googleMaps</code> are mandatory, so for changing one, you need to specify all the others as well.</p> <p><b>Sub-elements:</b></p> <ul style="list-style-type: none"> <li><code>webBaseUrlHttp</code> - Public URL base (HTTP) of the web interface</li> </ul>

- `webBaseUrlHttps` - Public URL base (HTTPS) of the web interface
- `soapBaseUrlHttp` - Public URL base (HTTP) of the SOAP interface
- `soapBaseUrlHttps` - Public URL base (HTTPS) of the SOAP interface
- `senderAddress` - E-mail address used by the system for outgoing e-mails
- `bccAddresses` - Comma-separated list of e-mail addresses that receive a blind copy of outgoing e-mails
- `operator` - Details about the operator in the sub-elements
  - `companyName` - the name of the company
  - `supportEMailAddress` - e-mail address of the operator's support
  - `supportPhone` - the phone number of the operator's support
- `googleMaps` - *optional* Details for the embedded Google Maps display. `httpKey` and `httpsKey` may be left blank. In this case, no Google Maps display is provided for the respective protocol.
  - `httpKey` - API key for the use of the Google Maps if the web interface is accessed via the HTTP protocol
  - `httpsKey` - API key if the HTTPS protocol is used
  - `initialLatitude` - latitude of the point that is initially displayed
  - `initialLongitude` - longitude of the point that is initially displayed

### ***updatePartitionResponse***

No elements.

### ***Example***

Update 'testpartition' to rename it to 'Party', set new maximum permissions for users of this partition and further configure this partition.

---

**Request**

---

```
<typ:updatePartitionRequest>
  <typ:name>testpartition</typ:name>
  <typ:newName>PartY</typ:newName>
  <typ:minPermissions></typ:minPermissions>
  <typ:maxPermissions>default-admin</typ:maxPermissions>
  <typ:configuration>
    <typ:webBaseUrlHttp>http://www.example.com</typ:webBaseUrlHttp>
    <typ:webBaseUrlHttps>https://www.example.com</typ:webBaseUrlHttps>
    <typ:soapBaseUrlHttp>http://soap.example.com</typ:soapBaseUrlHttp>
    <typ:soapBaseUrlHttps>https://soap.example.com</typ:soapBaseUrlHttps>
    <typ:senderAddress>cto@example.com</typ:senderAddress>
    <typ:bccAddresses>worker@example.com, janitor@example.com
      </typ:bccAddresses>
    <typ:operator>
      <typ:companyName>Example Ltd.</typ:companyName>
      <typ:supportEmailAddress>support@example.com</typ:supportEmailAddress>
      <typ:supportPhone>+42.1234567890</typ:supportPhone>
    </typ:operator>
  </typ:configuration>
</typ:updatePartitionRequest>
```

---

**Response**

---

```
<updatePartitionResponse
xmlns="http://xmlns.telnic.org/ws/nsp/admin/partition/types-1.0"/>
```

### 13.4. deletePartition

This operation deletes a Partition object. You cannot delete a partition that is still in use, that is, a partition with user or sub-partitions associated with it. Any permissions defined for the partition will be deleted along with the partition.

#### *deletePartitionRequest*

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

name	Name of the partition to be deleted.
------	--------------------------------------

#### *deletePartitionResponse*

No elements.

#### *Example*

Delete the partition 'party'.

#### **Request**

```
<typ:deletePartitionRequest>  
  <typ:name>party</typ:name>  
</typ:deletePartitionRequest>
```

#### **Response**

```
<deletePartitionResponse  
xmlns="http://xmlns.telnic.org/ws/nsp/admin/partition/types-1.0"/>
```

### 13.5. getPartition

This operation retrieves the configuration data for a partition.

#### ***getPartitionRequest***

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

name	Name of the partition to probe.
------	---------------------------------

#### ***getPartitionResponse***

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

name	Name of the partition.
------	------------------------

parent	<i>optional</i> Parent partition. If not given, the administrator's partition is used.
--------	--

minPermissions	<i>optional</i> Permissions that will be applied as the minimum permissions of all users of the partition and subordinate partitions. The referenced permissions must be defined in the parent partition.
----------------	---

maxPermissions	<i>optional</i> Permissions that will be applied as the maximum permissions of all users of the partition and subordinate partitions. The referenced permissions must be defined in the parent partition.
----------------	---

addressSelector	Address selectors identifying the given partition from the connection that the client uses. They are defined as a sequence of <code>selector</code> elements with the following <b>subelements</b> :
-----------------	--

- `service` - *optional* Service type (`soap` or `web`); if omitted, the selector is used for all service types.
- `transport` - *optional* Transport type (`HTTP`, `HTTPS`, or `both`); if omitted, the selector is used for all transport types.
- `virtualHostName` - *optional* Virtual name that the selector will match, i.e. the domain name given in the "host" HTTP header field. If omitted, all names are matched.
- `localHostName` - *optional* Local interface name the selector will match. If omitted, all names are matched.

configuration	Partition-specific configuration. If omitted, the configuration is initialized with empty data, which might cause some functionality to fail.
---------------	---

### Subelements:

- `webBaseURL` - Public URL base of the web interface
- `senderAddress` - E-mail address used by the system for outgoing e-mails
- `bccAddresses` - Comma-separated list of e-mail addresses that receive a blind copy of outgoing e-mails
- `operator` - Details about the operator in the subelements
  - `companyName` - the name of the company
  - `supportEMailAddress` - e-mail address of the operator's support
  - `supportPhone` - the phone number of the operator's support
- `googleMaps` - *optional* Details for the embedded Google Maps display. `httpKey` and `httpsKey` may be left blank. In this case, no Google Maps display is provided for the respective protocol.
  - `httpKey` - API key for the use of the Google Maps if the web interface is accessed via the HTTP protocol
  - `httpsKey` - API key if the HTTPS protocol is used
  - `initialLatitude` - latitude of the point that is initially displayed
  - `initialLongitude` - longitude of the point that is initially displayed.

### Example

Retrieve configuration data for partition 'PartY'.

---

#### Request

```
<typ:getPartitionRequest>  
  <typ:name>PartY</typ:name>  
</typ:getPartitionRequest>
```

---

#### Response

```
<ns5:getPartitionResponse  
  xmlns:ns5="http://xmlns.telnic.org/ws/nsp/admin/partition/types-1.0">
```

July 2009

---

```
<ns5:name>party</ns5:name>
<ns5:parent>root</ns5:parent>
<ns5:maxPermissions>default-admin</ns5:maxPermissions>
<ns5:addressSelectors>
  <ns5:selector>
    <ns5:virtualHostName>http://www.example.com</ns5:virtualHostName>
  </ns5:selector>
</ns5:addressSelectors>
<ns5:configuration>
  <ns5:webBaseURL>http://www.example.com</ns5:webBaseURL>
  <ns5:senderAddress>cto@example.com</ns5:senderAddress>
  <ns5:bccAddresses> worker@example.com, janitor@example.com
  </ns5:bccAddresses>
  <ns5:operator>
    <ns5:companyName>Example Ltd.</ns5:companyName>
    <ns5:supportEmailAddress>support@example.com</ns5:supportEmailAddress>
    <ns5:supportPhone>+42.1234567890</ns5:supportPhone>
  </ns5:operator>
</ns5:configuration>
</ns5:getPartitionResponse>
```

## 13.6. listPartitions

This operation lists the partition objects.

### ***listPartitionsRequest***

Element Name	Description
parent	<i>optional</i> Parent partition. Only sub-partitions of this partition will be given.  If not given, the administrator's partition is used.

### ***listPartitionsResponse***

List of partitions matching the request.

#### ***Example***

List sub-partitions of the root partition.

#### **Request**

```
<typ:listPartitionsRequest>  
  <typ:parent>root</typ:parent>  
</typ:listPartitionsRequest>
```

#### **Response**

```
<ns5:listPartitionsResponse  
  xmlns:ns5="http://xmlns.telnic.org/ws/nsp/admin/partition/types-  
1.0">  
  <ns5:partition name="ordinarycustomers"/>  
  <ns5:partition name="veryspecialpeople"/>  
  <ns5:partition name="partx"/>  
  <ns5:partition name="party"/>  
</ns5:listPartitionsResponse>
```

## 14. VIRTUAL FILE MANAGEMENT

### 14.1. Virtual File Management

In a multi-tenant context, the owners of different partitions of the same TelHosting Software installation may want to customize the appearance of their partition's web interface. This will usually involve setting up custom CSS or image files to be delivered to the web browser. Because all partition-related data is stored in the system's database, these customized web resources must be placed in the database as well. For this purpose, the TelHosting Software features a virtual file system that allows the management of files and their virtual paths on a per-partition basis. This section describes the corresponding SOAP API.

A summary of the virtual file system's SOAP API operations is shown in Table 8. The corresponding schema definition can be found in the file [VirtualFile-1.0.xsd](#).

**Table 8: Summary of virtual file API**

Operation	Description
<a href="#">createFile</a>	Creates a new virtual file in a partition.
<a href="#">updateFile</a>	Updates an existing virtual file.
<a href="#">deleteFile</a>	Deletes a virtual file.
<a href="#">listFiles</a>	Returns a list of all virtual files within a partition.
<a href="#">getFile</a>	Retrieves a virtual file stored at a given path.

## 14.2. createFile

This operation allows an administrator to create a new virtual file in a specified partition.

### ***createFileRequest***

Element Name	Description
partition	The partition to which the file will belong.
path	Path and name of the file. A SOAP fault will be returned if a file already exists at the specified path.
contentType	<i>optional</i> MIME type of the file; if omitted, the type is derived from the file name extension.
content	Base-64 encoded content of the file.

### ***createFileResponse***

No elements.

### ***Example***

Create file `logo.gif` in the root partition, located in folder `/css/images/`.

### **Request**

```
<typ:createFileRequest>
  <typ:partition>root</typ:partition>
  <typ:path>/css/images/logo.gif</typ:path>
  <typ:contentType>image/gif</typ:contentType>
  <typ:content>R0lGODlh0wAqAMQAAP [...]</typ:content>
</typ:createFileRequest>
```

### **Response**

```
<createFileResponse
xmlns="http://xmlns.telnic.org/ws/nsp/admin/virtualfile/types-1.0"/>
```

### 14.3. updateFile

This operation allows an administrator to update an existing file.

#### ***updateFileRequest***

Element Name	Description
partition	The partition to which the updated file belongs.
path	Path and name of the file to be updated. A SOAP fault will be returned if no file exists at the specified path.
contentType	<i>optional</i> New MIME type of the file; if omitted, the type is derived from the file name extension.
content	Base-64 encoded content of the file.

#### ***updateFileResponse***

No elements.

#### ***Example***

Update the `logo.gif` image.

#### **Request**

```
<typ:updateFileRequest>  
  <typ:partition>root</typ:partition>  
  <typ:path>/css/images/logo.gif</typ:path>  
  <typ:contentType>image/gif</typ:contentType>  
  <typ:content>Awj0xnio2etL7I1H [... ]</typ:content>  
</typ:updateFileRequest>
```

#### **Response**

```
<updateFileResponse  
  xmlns="http://xmlns.telnic.org/ws/nsp/admin/virtualfile/types-1.0"/>
```

## 14.4. deleteFile

This operation allows an administrator to delete a virtual file.

### ***deleteFileRequest***

Element Name	Description
partition	The partition to which the deleted file belongs.
path	Path and name of the file to be deleted.

### ***deleteFileResponse***

No elements.

### ***Example***

Delete file logo.gif.

### **Request**

```
<typ:deleteFileRequest>
  <typ:partition>root</typ:partition>
  <typ:path>/css/images/logo.gif</typ:path>
</typ:deleteFileRequest>
```

### **Response**

```
<deleteFileResponse
xmlns="http://xmlns.telnic.org/ws/nsp/admin/virtualfile/types-1.0">
```

## 14.5. listFiles

This operation allows an administrator to list the virtual files stored for a given partition.

### *listFilesRequest*

Element Name	Description
partition	The partition for which to list files.

### *listFilesResponse*

Element Name	Description
file	Representation of one file. <b>Sub-elements:</b> <ul style="list-style-type: none"> <li>• path - Path and name of the file.</li> <li>• contentType - MIME type of the file.</li> <li>• modificationDate - The date when the file was last modified.</li> </ul>

### *Example*

#### **Request**

```
<typ:listFilesRequest>
  <typ:partition>root</typ:partition>
</typ:listFilesRequest>
```

#### **Response**

```
<ns4:listFilesResponse
  xmlns:ns4="http://xmlns.telnic.org/ws/nsp/admin/virtualfile/types-
  1.0">
  <ns4:file>
    <ns4:path>/css/images/logo.gif</ns4:path>
    <ns4:contentType>image/png</ns4:contentType>
    <ns4:modificationDate>2008-04-01+02:00</ns4:modificationDate>
  </ns4:file>
  <ns4:file>
    <ns4:path>/css/images/logo2.gif</ns4:path>
    <ns4:contentType>text/plain</ns4:contentType>
    <ns4:modificationDate>2008-04-01+02:00</ns4:modificationDate>
  </ns4:file>
</ns4:listFilesResponse>
```

## 14.6. getFile

This operation allows an administrator to retrieve a file's content.

### *getFileRequest*

Element Name	Description
partition	The partition to which the file belongs.
path	Path and name of the file.

### *getFileResponse*

Element Name	Description
contentType	MIME type of the file.
modificationDate	Date of the file's last modification.
content	Base-64 encoded content of the file.

### *Example*

Get details for the file `logo.gif`.

### **Request**

```
<typ:getFileRequest>
  <typ:partition>root</typ:partition>
  <typ:path>/css/images/logo.gif</typ:path>
</typ:getFileRequest>
```

### **Response**

```
<ns4:getFileResponse
  xmlns:ns4="http://xmlns.telnic.org/ws/nsp/admin/virtualfile/types-
  1.0">
  <ns4:contentType>image/gif</ns4:contentType>
  <ns4:modificationDate>2008-04-01+02:00</ns4:modificationDate>
  <ns4:content>Awj0xnio2etL7I1H [...]</ns4:content>
</ns4:getFileResponse>
```