



Domain Edit API Dokumentation

Für DomainEdit Server-Version 1.0.0

Inhaltsverzeichnis

- Einleitung3
- API Methoden.....4
 - authenticate4
 - getSettings5
 - saveSettings6
 - getUsers7
 - getUser8
 - saveUser9
 - deleteUser10
 - getDomains11
 - getUserDomains12
 - saveUserDomains13
 - getServerVersion14
- Objekte15
 - User.....15
 - Domain16
 - Userdomain17
- Soap Faults.....18
 - 118
 - 1018
 - 2018
 - 2118
 - 2218
 - 3018
 - 4018
 - 4118
 - 6018
 - 8018
 - 8118
 - 10018
 - 10118

Einleitung

Bei der DomainEdit Reseller Schnittstelle handelt es sich um eine SOAP Schnittstelle. Bezüglich SOAP können Sie sich [hier](#) informieren.

Alle Anfragen zur DomainEdit API werden via SOAP beantwortet.

In einem Fehlerfall werden SoapFaults zurückgegeben, andernfalls der in der Methode angegebene Rückgabewert.

Jede Methode kann den [SoapFault 1](#) zurückliefern, sollte der Reseller nicht (mehr) authentifiziert sein. Weitere mögliche (SOAP) Fehlerfälle werden im Abschnitt [Soap Faults](#) behandelt.

In dieser Dokumentation werden Beispiele der Abfragen zur API mittels PHP Zend 1.10.8 SOAP Client durchgeführt.

Weitere Informationen über PHP Zend 1 SOAP finden Sie [hier](#).

API Beispiele folgen folgendem Muster:

{Methodenname}

Methode: {Methodenname}	{Version}
Beschreibung: {Beschreibung}	
Parameter:	
{Variablen Name}	{Datentyp} {Beispiel}
Rückgabe:	
{Variablen Name}	{Datentyp} {Beispiel}
Beispiel: <pre><?php try { // code beispiel der Abfrage in PHP } catch (Exception \$e) { print_r (\$e->getMessage()); }</pre>	
Mögliche Soap Faults: 1 , {weitere mögliche Soap Faults}	
Zusätzliches: {Zusätzliche Informationen}	

API Methoden

authenticate

Methode: authenticate			Version: 1.0
Beschreibung: Authentifizierung mittels der hosttech Domainreselling Zugangsdaten			
Parameter:			
	Benutzername	String	"MeinBenutzername"
	Passwort	String	"MeinPasswort"
Rückgabe:			
	SessionID	String	"km5uep9tteuwn5h23j1oagc5r4"
Beispiel:			
<pre> <?php try { // definieren der Zugangsdaten \$sURL = 'http://domainedit.ch/wsdl'; \$sUser = 'MeinBenutzername'; \$sPassword = 'MeinPasswort'; // holen der Session \$oSession = new Zend_Session_Namespace('api'); // instanzieren des Soap Clients \$oSoapClient = new Zend_Soap_Client(\$sURL); // wenn bereits eingeloggt, setzen des Session Cookies if(isset(\$oSession->sessid)) { \$oSoapClient->setCookie('PHPSESSID', \$oSession->sessid); } // authentifizieren \$sSessionID = \$oSoapClient->authenticate(\$sUser, \$sPassword); // setzen der SessionID in die Session für spätere Abfragen \$oSession->sessid = \$sSessionID; } catch(Exception \$e) { print_r(\$e->getMessage()); } </pre>			
Mögliche Soap Faults:			
<u>1</u>			
Zusätzliches:			
<p>Durch Speichern und Übergabe der Session mittels Cookie an Soap, ist eine interne neue Authentifizierung nicht jedes mal notwendig. Dies erspart Zeit und Ressourcen bei der Abfrage.</p> <p>Sollte die Session nicht mehr gültig sein, wird automatisch ein neuer Anmeldeversuch durchgeführt und die neue SessionID wieder zurückgegeben.</p>			

getSettings

Methode: getSettings		Version: 1.0
Beschreibung: Holen der DomainEdit Reseller Einstellungen		
Parameter: Keine		
Rückgabe:		
Settings	Array	Array ([address] => Street [city] => City [color] => FF00F0 [companyname] => hosttech GmbH [contactperson] => [email] => e@mail.tld [footerrow] => Footer [layout] => with_header [logo] => [logoutlink] => http://www.hosttech.ch [phone] => 0123456789 [zip] => 4444 [introduction] => Text)
Beispiel:		
<pre><?php try { // authentifizierung // \$oSoapClient->auth.... // // hole reseller einstellungen \$oSoapClient->getSettings () } catch (Exception \$e) { print_r (\$e->getMessage ()) } </pre>		
Mögliche Soap Faults: <u>1</u>		
Zusätzliches: Der Rückgabewert von Settings['layout'] besteht aus „with_header“ oder „no_header“ Der Rückgabewert von Settings['color'] ist ein Hexdezimaler Farbcode ohne „#“ (Raute)		

saveSettings

Methode: saveSettings		Version: 1.0
Beschreibung: Speichern verschiedener Reseller Einstellungen		
Parameter:		
Settings	Array	Array ([address] => Street [city] => City [color] => FF00F0 [companyname] => hosttech GmbH [contactperson] => [email] => e@mail.tld [footerrow] => Footer [layout] => with_header [logo] => [logoutlink] => http://www.hosttech.ch [phone] => 0123456789 [zip] => 4444 [introduction] => Text)
Rückgabe:		
Saved	Boolean	true / false
Beispiel:		
<pre> <?php try { // authentifizierung // \$oSoapClient->auth.... // // speichere reseller einstellungen \$oSoapClient->saveSettings (array('logoutlink' => 'http://www.google.ch', 'footerrow' => 'just a footer test', 'companyname' => 'Testing GmbH', 'address' => 'Adresse 123', 'email' => 'skaiser@hosttech.ch', 'zip' => '112233', 'city' => 'TestCity 1', 'phone' => '0190666666', 'color' => 'FF00F0', 'layout' => 'with_header', 'logo' => '', 'introduction' => 'Text')); } catch (Exception \$e) { print_r (\$e->getMessage ()) } </pre>		
Mögliche Soap Faults: 1 , 20		
Zusätzliches: Settings['logo'] wird der Inhalt eines Bildes ()		

Settings['layout'] darf nur „with_header“ oder „no_header“ beinhalten

getUsers

Methode: getUsers		Version: 1.0
Beschreibung: Hole alle Benutzer des Resellers		
Parameter: keine		
Rückgabe:		
Users	Array von Usern	Siehe Objekte
Beispiel:		
<pre> <?php try { // authentifizierung // \$oSoapClient->auth.... // // hole alle benutzer \$aUsers = \$oSoapClient->getUsers () } catch (Exception \$e) { print_r (\$e->getMessage ()) } </pre>		
Mögliche Soap Faults:		
<u>1</u>		
Zusätzliches:		

getUser

Methode: getUser		Version: 1.0
Beschreibung: Hole einzelnen Benutzer anhand ID		
Parameter:		
UserID	Int	BenutzerID (Bsp 385)
Rückgabe:		
User	User	Siehe Objekte
Beispiel:		
<pre> <?php try { // authentifizierung // \$oSoapClient->auth.... // // hole benutzer mit id 385 \$oUser = \$oSoapClient->getUser(385); } catch (Exception \$e) { print_r(\$e->getMessage()); } </pre>		
Mögliche Soap Faults: 1 , 10 , 40		
Zusätzliches:		

saveUser

Methode: saveUser		Version: 1.0	
Beschreibung: Speichere einen (neuen) Benutzer			
Parameter:			
User	User	Siehe Objekte	
Rückgabe:			
UserID	Int	BenutzerID Bsp 385	
Beispiel:			
<pre> <?php try { // authentifizierung // \$oSoapClient->auth.... // // erstelle neuen benutzer \$oUser = new stdClass(); \$oUser->name = 'Neuer Benutzer'; \$oUser->username = 'Benutzername'; \$oUser->password = 'Passwort'; \$oUser->email = 'mail@domain.tld'; \$oUser->blocked = false; \$oUser->userdatachange = true; // speichere benutzer \$iUserID = \$oSoapClient->saveUser(\$oUser); } catch (Exception \$e) { print_r(\$e->getMessage()); } </pre>			
Mögliche Soap Faults: 1 , 10 , 21 , 40 , 60 , 80 , 81 , 100 , 101			
Zusätzliches: Für das Bearbeiten eines bestehenden Benutzers im ÜbergabeObjekt User vorher die ID des zu bearbeitenden Benutzers setzen			

deleteUser

Methode: deleteUser		Version: 1.0	
Beschreibung: Lösche einen bestehenden Benutzer			
Parameter:			
userID	Int	Benutzer ID Bsp: 385	
Rückgabe:			
Deleted	boolean	true / false	
Beispiel:			
<pre> <?php try { // authentifizierung // \$oSoapClient->auth.... // // lösche benutzer \$bDeleted = \$oSoapClient->deleteUser(385); } catch (Exception \$e) { print_r(\$e->getMessage()); } </pre>			
Mögliche Soap Faults: 1 , 10 , 30 , 40			
Zusätzliches:			

getDomains

Methode: getDomains	Version: 1.0
Beschreibung: Erhalte eine Liste aller zur Verfügung stehenden Domains	
Parameter: keine	
Rückgabe: Domains Array von Domains Siehe Objekte	
Beispiel: <pre><?php try { // authentifizierung // \$oSoapClient->auth.... // // hole alle möglichen domains \$aDomains = \$oSoapClient->getDomains (); } catch (Exception \$e) { print_r (\$e->getMessage ()) } </pre>	
Mögliche Soap Faults: 1	
Zusätzliches: Mit dem Laden aller Domains wird zusätzlich die Liste evtl. veralteter Domains aktualisiert. Das aktualisieren einer Domain erfolgt ebenso, sobald der Benutzer eine Domain lädt.	

getUserDomains

Methode: getUserDomains		Version: 1.0
Beschreibung: Erhalte eine Liste aller Benutzer zugewiesenen Domains des gewählten Benutzers		
Parameter:		
UserID	Int	Benutzer ID (Bsp: 385)
Rückgabe:		
UserDomains	Array von Userdomains	Siehe Objekte
Beispiel:		
<pre><?php try { // authentifizierung // \$oSoapClient->auth.... // // hole alle domain zuweisungen von benutzer 385 \$aUserDomains = \$oSoapClient->getUserDomains (385); } catch (Exception \$e) { print_r (\$e->getMessage ()) } </pre>		
Mögliche Soap Faults:		
<u>1</u>		
Zusätzliches:		

saveUserDomains

Methode: saveUserDomains		Version: 1.0
Beschreibung: Speicher eine neue Liste von Domainzuordnungen eines Benutzers		
Parameter:		
UserID	Int	Benutzer ID (Bsp: 385)
Userdomains	Array von Userdomains	Siehe Objekte
Rückgabe:		
Saved	Boolean	true / false
Beispiel:		
<pre> <?php try { // authentifizierung // \$oSoapClient->auth.... // // domain \$oDomain = new stdClass(); \$oDomain->id = 63695; // domainid muss vorher ausgelesen werden // erstelle userdomain \$oUserDomain = new stdClass(); \$oUserDomain->domain = \$oDomain; \$oUserDomain->rights = 'contactdata;details'; \$aUserDomains = array(); \$aUserDomains[] = \$oUserDomain; // speicher userdomain \$bSaved = \$oSoapClient->saveUserDomains(385, \$aUserDomains); } catch(Exception \$e) { print_r(\$e->getMessage()); } </pre>		
Mögliche Soap Faults: 1 , 10 , 22 , 40 , 41		
Zusätzliches: Eine Liste der möglichen Domain kann mit getDomains vorher ausgelesen werden. Eine Liste der möglichen Benutzer kann mit getUsers vorher ausgelesen werden.		

getServerVersion

Methode: getServerVersion	Version: 1.0	
Beschreibung: Laden des Versionsstrings des Servers		
Parameter: keine		
Rückgabe:		
Version	String	1.0.0
Beispiel: <pre><?php try { // authentifizierung // \$oSoapClient->auth.... // // laden der Serverversion \$sVersion = \$oSoapClient->getServerVersion(); } catch (Exception \$e) { print_r(\$e->getMessage()); }</pre>		
Mögliche Soap Faults: 1		
Zusätzliches:		

Objekte

User

```
/**
 * DomainEdit User
 */
class User
{
    /**
     * @var int
     */
    public $id = 0;

    /**
     * @var string
     */
    public $name = '';

    /**
     * @var string
     */
    public $username = '';

    /**
     * @var string
     */
    public $password = '';

    /**
     * @var string
     */
    public $email = '';

    /**
     * @var bool
     */
    public $blocked = false;

    /**
     * @var bool
     */
    public $userdatachange = true;
}
```

Domain

```
/**
 * DomainEdit Domain
 */
class Domain
{
    /**
     * @var int
     */
    public $id = 0;

    /**
     * @var string IDNA encoded Domainname
     */
    public $name = ''; // name IDNA encoded. bsp 'xn--tst-sna.ch'

    /**
     * @var Userdomain[]
     */
    public $userdomains = array(); // array of userdomain objects
}
```


Userdomain

```
/**
 * DomainEdit Userdomain
 */
class Userdomain
{
    /**
     * @var int
     */
    public $id = 0;

    /**
     * @var User
     */
    public $user; // user object

    /**
     * @var Domain
     */
    public $domain; // domain object

    /**
     * @var string
     * @notice semicolon seperated list of
     *         following allowed strings:
     *
     *         contactdata
     *         nameserver
     *         dnsrecords
     *         details
     *         authcode
     */
    public $rights = '';
}
```

Soap Faults

Fault Code	FaultString	Beschreibung
1	Ungültige Authentifizierung	Fehlerhafte Zugangsdaten
10	Keine Berechtigung	Die Abfrage greift auf einen Zugangsbeschränkten Datensatz zu, welcher mit Ihren Zugangdaten nicht erfasst werden kann.
20	Fehler beim Abspeichern der Einstellungen	Fehlerhafte / nicht validierbare Übergabeparameter
21	Fehler beim Abspeichern des Benutzers	Fehlerhafte / nicht validierbare Übergabeparameter
22	Fehler beim Abspeichern der Benutzerdomains	Fehlerhafte / nicht validierbare Übergabeparameter
30	Fehler beim Löschen des Benutzers	Fehlerhafte / nicht validierbare Übergabeparameter
40	Benutzer nicht gefunden	Benutzer nicht (mehr) in der Datenbank vorhanden
41	Domain nicht gefunden	Domain nicht (mehr) in der Datenbank vorhanden
60	E-Mail Adresse ist nicht valid	Übergebene E-Mail Adresse ist nicht valid. Prüfung folgt Richtlinien von Zend\Validator\EmailAddress
80	E-Mail Adresse existiert bereits	E-Mail Adressen werden Unique behandelt („Passwort vergessen“ - Funktion anhand E-Mail Adresse)
81	Benutzername existiert bereits	Benutzernamen werden Unique behandelt (Login via Username)
100	Benutzername darf nicht leer sein	Benutzername muss mindestens ein Non-Whitespace-Charakter beinhalten
101	Passwort darf nicht leer sein	Passwort muss mindestens ein Non-Whitespace-Charakter beinhalten