VISIT-X.net Signup Api V1.0

This documentation provides an overview of the functionality of the VISIT-X.net signup API. To use the API signup, you need a key associated with your webmaster ID. You get this key from your sales contact.

Structure of the form

You can use the API with a standard form. In this form every field you utilise must have an ID attribute assigned to it which the API can access. Such a form could look like this:

```
<form name="register" id="register" method="post" action="">
   <label>username: <input type="text" name="username"
id="username"></label><br>
   <label>email: <input type="text" name="email" id="email"></label><br>
   <label>password: <input type="text" name="password"
id="password"></label><br>
   <input type="submit" value="Abschicken">
</form>
```

The attribute “action” of the form remains blank.

Embedding of the API in your form

To embed the API in your form you add a JavaScript tag first, which then loads the API:

```
<script
src="https://www.wazazu.com/VX/SI/API?w=XXXXXX&ws=YYYYYY&pfm=1502&wt=ZZZZZZ
&key=YOURKEY"></script>
```

Please replace XXXXXX with your webmaster ID and YYYYYYYY with your campaign ID.

With wt=ZZZZZZZ you can commit its own token if you have set up a S2S-Callback with us.

For a normal registering with VISIT-X.NET please leave the parameter "pfm" at 1502 or remove it completely.

Should you want to link to one of your VISIT-X.net kits, please enter pfm=2107 instead.

The function pfm only works with the right combinations of w= and ws=. For the correct data please refer to the page "Kampagnen verwalten" (Manage campaigns) in your VX-CASH account.

Then you extend your form tag with an onSubmit attribute as follows:

```
<form name="register" id="register" method="post" action=""
     onsubmit="return window.vxApi.checkSignup('username', 'email',
'password')">
```

The function 'window.vxApi.checkSignup' always receives the IDs of the fields "username", "email" and "password" and always returns a "false" to prevent the sending of the form. After the API has tested the field values, an attempt is made to create a new VISIT-X.net user with the data given.

If this is successful the user will then be redirected to the VISIT-X.net homepage with an access token or, alternatively, to your own kit.

The API Functions

The API offers some functions that you can use to determine its behavior.

The possible functions are:

window.vxApi.checkUsername('usernameId')

Commit the ID of the field for the username to this function to check whether a username meets all the basic requirements (minimum and maximum length, etc.). This function will show an error message and return Boolean 'false' when the requirements aren't met, otherwise it returns 'true'.

This can, for example, be added to an "onblur" event of the username field.

window.vxApi.checkEmail('emailId')

Commit the ID of the field for the email address to this function to check whether the email given meets the basic requirements. This function will show an error message and return Boolean 'false' when the requirements aren't met, otherwise it returns 'true'.

window.vxApi.checkPassword('passwordId')

Commit the ID of the field for the password to this function to check whether the chosen password meets the basic requirements. This function will show an error message and return Boolean 'false' when the requirements aren't met, otherwise it returns 'true'.

window.vxApi.checkSignup('usernameId', 'emailId', 'passwortId')

Commit the IDs of the fields for username, email and password to this function. The function performs the above three functions for each field. If no errors occur a new VISIT-X.net user with the data given is created.

If errors occur (e.g. because the username has been assigned already) the error message will, in the default implementation, be issued as alert box.

If you want to show the error messages yourself (e.g. in your own modal box) you can use the following function:

window.vxApi.setAlert(callable)

With this function you can set up your own function for the API that takes care of the issuing of the error messages. When errors occur the function receives a JavaScript object containing one or more keys that show in which fields errors have occurred. The values of the keys are an array of **german** error messages. A typical object might look like this:

```
{
  "username": [
    "Bitte geben Sie mindestens 5 Zeichen ein."
  ],
  "email": [
    "Bitte geben Sie eine valide E-Mail-Adresse ein."
  ]
}
```

window.vxApi.setHandler(callable)

If you want to not only show the standard messages in your own way, but handle the whole error management yourself you can use this function. It commits its own function to the API, taking care of the management of the error handling.

Please note that **only** errors coming from the **backend** go through this function. Error messages from the simple checks above continue to be handled directly in the alert function.

Your handling function should accept three values:

1. An error ID
2. An error object
3. If necessary, an alternative name

The error ID is an integer value which numerically presents the error found:

Error ID	Error type
1	Username too short
2	Username too long
4	Username already assigned
8	Incorrect characters in username (umlauts, etc.)
16	Server connection error
32	Invalid email address (e.g. disposable email addresses, non-existent domains, etc.)
128	Invalid username (unwanted terms, etc.)
256	Invalid password

The error object is equivalent to the form described already in window.vxApi.setAlert(callable).

Should the error refer to the username our server tries to create one alternative suggestion. This is committed as 3rd parameter. If no suitable alternative suggestion could be found this variable contains the original username entered.

The function must return a Boolean value.

If set to "true", the original error messages are issued as usual via the alert function. If set to “false” the alert function is not activated, not even if you embedded your own via setAlert. In this case the handling function has to take care of issuing a notice to the user itself.