

## VISIT-X.net Signup Api V1.0

Diese Dokumentation gibt Ihnen einen Überblick über die Funktionalität der VISIT-X.net Signup API. Um die Signup API nutzen zu können benötigen Sie einen mit ihrer Webmaster ID verbundenen Schlüssel. Diesen erhalten Sie von Ihrem Sales Kontakt.

### Aufbau des Formulars

Die API kann mit einem Standard-Formular genutzt werden, wobei jedes der verwendeten Felder mit einem ID-Attribut belegt sein muss, damit die API diese ansprechen kann. Ein Formular könnte z.B. wie folgt aufgebaut sein:

```
<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>
```

Das Attribut "action" des Formulars wird leer gelassen.

## Einbinden der API in Ihr Formular

Um die API an Ihr Formular zu binden, fügen Sie zuerst ein Javascript-Tag ein, mit dem die API geladen wird:

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

Ersetzen Sie XXXXXX durch Ihre Webmaster ID und YYYYYYYY durch Ihre Kampagnen ID.

Mit wt=ZZZZZZ können Sie ein eigenes Token übergeben, falls Sie einen S2S-Callback bei uns eingerichtet haben.

Für eine normale Anmeldung an VISIT-X.net lassen Sie den Parameter „pfm“ auf 1502 oder entfernen Sie ihn komplett.

Sollten Sie auf einen Ihrer VISIT-X.net Baukästen linken wollen tragen Sie stattdessen bitte pfm=2107 ein.

Das ändern der pfm funktioniert nur mit den richtigen Kombinationen von w= und ws=. Bitte entnehmen Sie die korrekten Daten der Seite „Kampagnen verwalten“ in ihrem VX-CASH Account.

Anschließend erweitern Sie Ihr Form Tag mit einem onSubmit Attribut wie folgt:

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

Die Funktion “window.vxApi.checkSignup” bekommt immer die IDs der Felder “username”, “email” und “passwort” übergeben und liefert immer ein “false” zurück, um das Absenden des Formulars zu verhindern. Nachdem die API die Feldwerte getestet hat, wird ein Versuch unternommen einen neuen User mit den angegebenen Daten bei VISIT-X.net anzulegen.

Bei Gelingen wird der Besucher mit einem Zugangstoken auf die VISIT-X.net Startseite (alternativ auf Ihren Baukasten) geschickt.

## Funktionen der API

Die API bietet Ihnen einige Funktionen an mit denen Sie das Verhalten selbst bestimmen können.

Die möglichen Funktionen sind:

### **window.vxApi.checkUsername('usernameId')**

Übergeben Sie dieser Funktion die ID des Usernamen-Feldes um zu überprüfen ob ein eingegebener Username den grundsätzlichen Ansprüchen (Mindest- bzw. Maximallänge etc.) entspricht. Gibt bei Fehlern eine Fehlermeldung aus und false zurück, ansonsten true.

Kann z.B. an einen „onblur“ Event des Username-Feldes gehängt werden.

### **window.vxApi.checkEmail('emailId')**

Übergeben Sie dieser Funktion die ID des Email-Feldes um zu überprüfen ob eine eingegebene E-Mail-Adresse den grundsätzlichen Ansprüchen entspricht. Gibt bei Fehlern eine Fehlermeldung aus und false zurück, ansonsten true.

### **window.vxApi.checkPassword('passwordId')**

Übergeben Sie dieser Funktion die ID des Passwort-Feldes um zu überprüfen ob ein eingegebenes Passwort den grundsätzlichen Ansprüchen entspricht. Gibt bei Fehlern eine Fehlermeldung aus und false zurück, ansonsten true.

### **window.vxApi.checkSignup('usernameId', 'emailId', 'passwordId')**

Übergeben Sie dieser Funktion die IDs des Usernamen-, E-Mail- und Passwort-Feldes. Die Funktion führt die obigen drei Funktionen für das jeweilige Feld aus. Falls keine Fehler auftreten wird ein neuer VISIT-X.net Benutzer mit den eingegebenen Daten angelegt.

Sollten dabei Fehler auftreten (z.B. weil der Username bereits vergeben ist) werden in der Standard-Implementation die Fehlermeldung als Alert Box ausgegeben.

Falls Sie die Fehlermeldung(en) selbst darstellen wollen (z.B. in einer eigenen Modal-Box) nutzen Sie die folgende Funktion:

### **window.vxApi.setAlert(callable)**

Mit dieser Funktion können Sie der API eine eigene Funktion reichen, die sich um die Ausgabe der Fehlermeldungen kümmert. Die Funktion bekommt bei Fehlern ein Javascript Object mit einem oder mehreren Keys die zeigen für welches Feld Fehler vorhanden sind. Die Values der Keys sind ein Array von Fehlermeldungen. Ein typisches Objekt könnte z.B. wie folgt aussehen:

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

## window.vxApi.setHandler(callable)

Falls Sie nicht nur die Fehlermeldungen auf Ihre eigene Art ausgeben, sondern gleich das gesamte Fehlerhandling selbst gestalten wollen, können Sie diese Funktion benutzen um eine eigene Funktion in die API zu reichen, die sich um das Fehlerhandling kümmert.

Bitte beachten Sie dabei, dass **nur** Fehler die aus dem **Backend** kommen diese Funktion durchlaufen. Fehlermeldungen aus den einfachen Prüffunktionen oben werden weiterhin direkt in der Alert Funktion abgehandelt.

Ihre Handler-Funktion sollte drei Werte entgegennehmen:

1. Eine Fehler ID
2. Ein Fehler-Objekt
3. Gegebenenfalls einen Alternativnamen

Die Fehler ID ist ein Integer Wert, welcher den gefundenen Fehler numerisch darstellt:

Fehler ID	Fehlertyp
1	Username zu kurz
2	Username zu lang
4	Username bereits vergeben
8	Fehlerhafte Zeichen im Usernamen (Umlaute etc.)
16	Verbindungsfehler zum Server
32	Ungültige E-Mail Adresse (z.B. WegWerfAdresse, nicht existierende Domain etc.)
128	Ungültiger Username (unerwünschte Worte etc.)
256	Ungültiges Passwort

Das Fehler Objekt entspricht der Form wie sie bereits in window.vxApi.setAlert(callable) beschrieben wurde.

Sollte der Fehler sich auf den Usernamen beziehen, versucht unser Server einen alternativen Vorschlag zu erstellen. Dieser wird als dritter Parameter übergeben. Konnte kein passender alternativer Vorschlag gefunden werden enthält diese Variable den original eingegebenen Usernamen.

Ihre Funktion muss einen boolschen Wert zurückgeben.

Bei „true“ werden die Original-Fehlermeldungen wie bisher über die Alert Funktion ausgegeben. Bei false wird die Alert Funktion nicht aufgerufen (auch nicht, falls sie via setAlert eine eigene eingebunden haben) und die Handler Funktion muss sich selbst um die Benachrichtigung des Besuchers kümmern