

VivitiCMS

Partner API Documentation

Trade Secrets / Confidential Information

This material contains trade secrets and/or confidential information of Viviti Technologies, Inc. and may not be used, disclosed, or copied without the permission of Viviti Technologies, Inc.
Copyright © 2015 Viviti Technologies, Inc. All rights reserved.

Table of Contents

- Partner API Documentation 1**
- Getting Started..... 5**
 - Overview..... 5
 - Endpoints 5
 - Authentication 5
 - Status 5
- Server Errors..... 6**
 - Missing Parameter Errors..... 6
- Basic Integration Process 7**
 - Account Creation..... 7
 - Website Creation 7
 - 1)Host Creation (Optional)..... 8
- API Requests 10**
- Accounts 10**
 - Overview..... 10
 - Create Account 11
 - Delete Account..... 13
 - Suspend Account 14
 - Unsuspend Account..... 15
 - Retrieve Account Details 16
 - Retrieve Account Details by Email..... 17
 - Renew Account..... 18

Update Account Package.....	19
Update Account Package Limits	21
List Account Websites	23
Websites	24
Overview.....	24
Create Website.....	25
Update Powered By HTML.....	27
Update Allowed Languages.....	28
Delete Website	29
Suspend Website.....	30
Unsuspend Website	31
Retrieve Website Details	32
Retrieve Login URL For Website.....	33
Apply New Theme To Website.....	34
Themes	35
Retrieve Theme Details	36
Apply New Theme To Website, by Theme ID	37
Hosts.....	38
Overview.....	38
Create A Host	39
Validate a Host	40
Delete A Host	41
Set the host to primary	42
Partners	43

Overview..... 43

List Available Packages 43

Email 44

Overview..... 44

Email Address..... 45

Email Alias..... 45

Domain Key..... 46

Domain Key Signature 46

Blacklist..... 47

List of blacklisted values by type 47

Getting Started

Overview

Viviti allows partners to use our API to create and manage accounts and websites. Calls to the API are made via standard HTTP requests.

Endpoints

Production Server: <https://api.vivitiapp.com>

Test Server: <https://api.staging.vivitiapp.com>

Authentication

Every call to the system requires the following parameter:

Parameter Name	Type	Description
api_key	String	The API Key that is provided to you.

Status

API calls will be responded to with an appropriate HTTP header, and any content in the body will be in XML.

HTTP Header	Cause	Response Body
200	Success	XML
422	Error with request	XML (Error message)
401	Authorization failure (API key or security hash are invalid for the request)	XML (Error message)
404	Requested account or website not found	XML (Error message)
500	There was a problem with your request that has been logged and will be investigated by the Viviti team.	XML (Error message)

Server Errors

In the event that there are issues on our side that are preventing your request from completing, the following response will be returned:

```
<errors>  
  <error>There was an error processing your request.</error>  
</errors>
```

If this response is received the problem on our end will have been logged and we should be looking into it as soon as possible.

Missing Parameter Errors

If you're missing parameters that are required to complete the request, the API will let you know with the following response:

```
<errors>  
  <error>Missing required parameters: id and host</error>  
</errors>
```

Basic Integration Process

Account Creation

The first step to integrating with the Viviti API is to create an account for your user:

Request:

```
POST /account?email=example%40gmail.com&package=3&api_key=API_KEY&term=12
```

Response:

```
<account>
  <created-at type="datetime">2010-02-17T13:30:42-08:00</created-at>
  <email>example@viviti.com</email>
  <id type="integer">8</id>
  <package-id type="integer">3</package-id>
  <payment-due-date type="date">2011-02-17</payment-due-date>
  <security-hash>bd2e6cde672b2f493d1b4efb45cd4d1be1f30900</security-hash>
  <suspended type="boolean">>false</suspended>
  <trial type="boolean">>false</trial>
  <account-limits>
    <storage type="integer">21474836480</storage>
    <pages type="integer">15</pages>
    <websites type="integer">2</websites>
  </account-limits>
  <account-usage>
    <storage type="integer">468591</storage>
    <pages type="integer">5</pages>
    <websites type="integer">10</websites>
  </account-usage>
</account>
```

You must keep a record of both the **id** and **security-hash** elements in the response. Most future requests will require these parameters.

Website Creation

After the account has been created, a website can now be created on the account using the **id** and **security-hash** received from the previous step:

Request:

```
POST
/website?account_id=8&api_key=API_KEY&security_hash=bd2e6cde672b2f493d1b4efb45cd4d1be1f30900&host=example.viviti.com
```

Response:

```
<website>
  <created-at type="datetime">2010-02-17T14:36:26-08:00</created-at>
```

```

<id type="integer">12</id>
<primary-host-id type="integer" nil="true" ></primary-host-id>
<suspended type="boolean">>false</suspended>
<hosts type="array">
  <host>
    <id type="integer">12</id>
    <name>example.viviti.com</name>
  </host>
</hosts>
<ip-address>127.0.0.1</ip-address>
</website>

```

You must keep a record of the **id** element in the response. Most future requests dealing with the website will require this parameter.

Optionally, the **ip-address** should also be stored for future host creation. See the next step for more details regarding this.

When you're using the Viviti API, you're responsible for the creation of the DNS records for the websites you create. The **ip-address** element in the XML response contains the IP address that you'll need for the DNS creation. The following DNS records should be created on your system:

Record Type	Name	Content
A	127.0.0.1	example.viviti.com
CNAME	www.example.viviti.com	example.viviti.com

1) Host Creation (Optional)

If your users would like to point a domain they've registered with you, or a domain they've registered somewhere else then you'll need to add a host for that domain.

Request:

```

POST
/host?api_key=API_KEY&security_hash=bd2e6cde672b2f493d1b4efb45cd4d1be1f30900&website_id=12&host=example.com

```

Response:

```

<host>
  <id type="integer">13</id>
  <name>example.com</name>
  <website-id type="integer">12</website-id>
</host>

```


You must keep a record of the **id** element in the response. Most future requests dealing with the host will require this parameter.

You'll then need to add DNS records using the **ip-address** from the previous step. The following DNS records should be created on your system:

Record Type	Name	Content
A	127.0.0.1	example.com
CNAME	www.example.com	example.com

API Requests

Accounts

Every website in the system is owned by an account. Packages and service limits are set on the account and if the package allows multiple websites, are shared amongst all the websites on the account.

When an account is created a security hash will be returned which should be stored along with the account id. All subsequent requests to do anything with this account will require this security hash to be passed as a parameter.

Accounts will be deactivated several weeks after their expiration date if the account has not been renewed, and trial accounts will be deactivated after 14 days unless they are set to a proper package first. Deactivated accounts can be re-activated, but if left for more than a month will be deleted permanently.

Overview

ID in the URL refers to the ID of the account, which is returned when you create it

HTTP Method	URL	Description
POST	/account	Create a new account.
POST	/account/ ID /delete	Deletes an account.
GET	/account/ ID	Get details for the account.
POST	/account/ ID /renew	Renew the account.
POST	/account/ ID /suspend	Suspends the account.
POST	/account/ ID /unsuspend	Unsuspects the account.
POST	/account/ ID /update_package	Set a new package for the account
POST	/account/ ID /update_package_limits	Set limitations on the account.
GET	/account/ ID /websites	List of websites for the account

Create Account

Create a new account. If no package is specified, the new account will be a trial account that will get automatically removed after 14 days.

HTTP Method: POST

URL: /account

Required Parameters:

Parameter Name	Type	Description
<i>email</i>	String	Email address of the user.

Optional Parameters:

Parameter Name	Type	Description
<i>term</i>	Integer	The term to entitle the user for in months.
<i>package</i>	Integer	The ID of the package to entitle the user with.
<i>limitations[allowed_websites]</i>	Integer	Sets a limit on the account of how many websites it can create.
<i>limitations[allowed_storage]</i>	Integer	Sets a limit on the account of how much storage the site can use.
<i>limitations[allowed_pages]</i>	Integer	Sets a limit on the account of how many websites it can create.

-

Example Request:

POST /account?email=example%40gmail.com&package=3&api_key=API_KEY&term=12

Example Response:

```
<account>
  <created-at type="datetime">2010-02-17T13:30:42-08:00</created-at>
  <email>example@viviti.com</email>
  <id type="integer">8</id>
  <package-id type="integer">3</package-id>
  <payment-due-date type="date">2011-02-17</payment-due-date>
  <security-hash>bd2e6cde672b2f493d1b4efb45cd4d1be1f30900</security-hash>
  <suspended type="boolean">>false</suspended>
  <trial type="boolean">>false</trial>
  <account-limits>
    <storage type="integer">21474836480</storage>
    <pages type="integer">15</pages>
    <websites type="integer">2</websites>
  </account-limits>
  <account-usage>
    <storage type="integer">468591</storage>
    <pages type="integer">5</pages>
    <websites type="integer">10</websites>
  </account-usage>
</account>
```

Delete Account

Permanently removes the account from the system. All websites must be removed from the account before removal is possible as a safeguard for accidental deletion.

Important: The account cannot be recovered after this call is used.

HTTP Method: POST

URL: /account/1/delete

Required Parameters:

Parameter Name	Type	Description
<i>security_hash</i>	String	The security hash for the account that owns this website.

Example Request:

POST

/account/9/delete?api_key=API_KEY&security_hash=e58a0780d910e385af3f118b544a87e10d431e8f

Example Response:

```
<account>
  <created-at type="datetime">2010-02-17T13:30:42-08:00</created-at>
  <email>example@viviti.com</email>
  <id type="integer">8</id>
  <package-id type="integer">3</package-id>
  <payment-due-date type="date">2011-02-17</payment-due-date>
  <security-hash>bd2e6cde672b2f493d1b4efb45cd4d1be1f30900</security-hash>
  <suspended type="boolean">>false</suspended>
  <trial type="boolean">>false</trial>
  <account-limits>
    <storage type="integer">21474836480</storage>
    <pages type="integer">15</pages>
    <websites type="integer">2</websites>
  </account-limits>
  <account-usage>
    <storage type="integer">468591</storage>
    <pages type="integer">5</pages>
    <websites type="integer">10</websites>
  </account-usage>
</account>
```

Suspend Account

Suspends all of the websites under the account and makes it so that the message passed is displayed when attempting to visit any of the websites.

HTTP Method: POST

URL: /account/1/suspend

Required Parameters:

Parameter Name	Type	Description
<i>security_hash</i>	String	The security hash for the account that owns this website.
<i>message</i>	String	The suspend message that will be displayed on the users websites.

Example Request:

```
GET
/account/8/suspend?api_key=API_KEY&security_hash=bd2e6cde672b2f493d1b4efb45cd4d1be1f30900
&message=Has%20not%20paid
```

Example Response:

```
<account>
  <created-at type="datetime">2010-02-17T13:30:42-08:00</created-at>
  <email>example@viviti.com</email>
  <id type="integer">8</id>
  <package-id type="integer">3</package-id>
  <payment-due-date type="date">2011-02-17</payment-due-date>
  <security-hash>bd2e6cde672b2f493d1b4efb45cd4d1be1f30900</security-hash>
  <suspended type="boolean">>false</suspended>
  <trial type="boolean">>false</trial>
  <account-limits>
    <storage type="integer">21474836480</storage>
    <pages type="integer">15</pages>
    <websites type="integer">2</websites>
  </account-limits>
  <account-usage>
    <storage type="integer">468591</storage>
    <pages type="integer">5</pages>
    <websites type="integer">10</websites>
  </account-usage>
</account>
```

Unsuspend Account

This will unsuspend the account and all websites associated with it.

HTTP Method: POST

URL: /account/1/unsuspend

Required Parameters:

Parameter Name	Type	Description
<i>security_hash</i>	String	The security hash for the account that owns this website.

Example Request:

```
GET
/account/8/unsuspend?api_key=API_KEY&security_hash=bd2e6cde672b2f493d1b4efb45cd4d1be1f30900
```

Example Response:

```
<account>
  <created-at type="datetime">2010-02-17T13:30:42-08:00</created-at>
  <email>example@viviti.com</email>
  <id type="integer">8</id>
  <package-id type="integer">3</package-id>
  <payment-due-date type="date">2011-02-17</payment-due-date>
  <security-hash>bd2e6cde672b2f493d1b4efb45cd4d1be1f30900</security-hash>
  <suspended type="boolean">>false</suspended>
  <trial type="boolean">>false</trial>
  <account-limits>
    <storage type="integer">21474836480</storage>
    <pages type="integer">15</pages>
    <websites type="integer">2</websites>
  </account-limits>
  <account-usage>
    <storage type="integer">468591</storage>
    <pages type="integer">5</pages>
    <websites type="integer">10</websites>
  </account-usage>
</account>
```

Retrieve Account Details

Get details for the account, including the ID, email address, payment due date, payment term, and package ID.

HTTP Method: GET

URL: /account/1

Required Parameters:

Parameter Name	Type	Description
<i>security_hash</i>	String	The security hash for the account that owns this website.

Example Request:

```
GET /account/8?api_key=API_KEY&security_hash=bd2e6cde672b2f493d1b4efb45cd4d1be1f30900
```

Example Response:

```
<account>
  <created-at type="datetime">2010-02-17T13:30:42-08:00</created-at>
  <email>example@viviti.com</email>
  <id type="integer">8</id>
  <package-id type="integer">3</package-id>
  <payment-due-date type="date">2011-02-17</payment-due-date>
  <security-hash>bd2e6cde672b2f493d1b4efb45cd4d1be1f30900</security-hash>
  <suspended type="boolean">>false</suspended>
  <trial type="boolean">>false</trial>
  <account-limits>
    <storage type="integer">21474836480</storage>
    <pages type="integer">15</pages>
    <websites type="integer">2</websites>
  </account-limits>
  <account-usage>
    <storage type="integer">468591</storage>
    <pages type="integer">5</pages>
    <websites type="integer">10</websites>
  </account-usage>
</account>
```


Retrieve Account Details by Email

Get details for the account, available for 24 hours after the account has been created.

HTTP Method: GET

URL: /account-details/

Required Parameters:

Parameter Name	Type	Description
<i>email</i>	String	The email for the account

Example Request:

```
GET /account-details/?api_key=API_KEY&email=you@example.com
```

Example Response:

```
<account>
<created-at type="datetime">2013-02-05T11:52:29-08:00</created-at>
<email>someguy22@example.com</email>
<id type="integer">5</id>
<suspended type="boolean">>false</suspended>
<security-hash>48b21da1bc90630efb9f7bf155cb674f535d02ec</security-hash>
<package-id type="integer">13</package-id>
<trial type="boolean">>true</trial>
<payment-due-date type="date">2013-02-19</payment-due-date>
<account-limits>
<storage type="integer">5242880</storage>
<pages type="integer">3</pages>
<websites type="integer">10</websites>
</account-limits>
<account-usage>
<storage type="integer">0</storage>
<pages type="integer">0</pages>
<websites type="integer">1</websites>
</account-usage>
</account>
```

Example Response For Account Over 24 Hours Old:

```
<errors>
<error>Access prohibited</error>
</errors>
```

Renew Account

Extend the accounts expiry date by a number of months.

HTTP Method: POST

URL: /account/1/renew

Required Parameters:

Parameter Name	Type	Description
<i>term</i>	Integer	The term in months to extend the account for.
<i>security_hash</i>	String	The security hash for the account.

Example Request:

POST

/account/8/renew?api_key=API_KEY&security_hash=bd2e6cde672b2f493d1b4efb45cd4d1be1f30900&term=6

Example Response:

```
<account>
  <created-at type="datetime">2010-02-17T13:30:42-08:00</created-at>
  <email>example@viviti.com</email>
  <id type="integer">8</id>
  <package-id type="integer">3</package-id>
  <payment-due-date type="date">2011-08-17</payment-due-date>
  <security-hash>bd2e6cde672b2f493d1b4efb45cd4d1be1f30900</security-hash>
  <suspended type="boolean">>false</suspended>
  <trial type="boolean">>false</trial>
  <account-limits>
    <storage type="integer">21474836480</storage>
    <pages type="integer">15</pages>
    <websites type="integer">2</websites>
  </account-limits>
  <account-usage>
    <storage type="integer">468591</storage>
    <pages type="integer">5</pages>
    <websites type="integer">10</websites>
  </account-usage>
</account>
```

Update Account Package

Update the account to use a new package. This will change the expiry date, and the new expiry date will be returned with the account details in the response. If term is passed, it will also renew the account for the number of months specified (after dealing with pro-rating).

HTTP Method: POST

URL: /account/1/update_package

Required Parameters:

Parameter Name	Type	Description
package	Integer	The ID of the new package to use.
security_hash	String	The security hash for the account.

Optional Parameters:

Parameter Name	Type	Description
term	Integer	Term in months to extend the account for.
<i>limitations[allowed_websites]</i>	Integer	Sets a limit on the account of how many websites it can create.
<i>limitations[allowed_storage]</i>	Integer	Sets a limit on the account of how much storage the site can use.
<i>limitations[allowed_pages]</i>	Integer	Sets a limit on the account of how many websites it can create.

Example Request:

POST

```
/account/8/update_package?api_key=API_KEY&package=2&security_hash=bd2e6cde672b2f493d1b4efb45cd4d1be1f30900
```

Example Response:

```
<account>
  <created-at type="datetime">2010-02-17T13:30:42-08:00</created-at>
  <email>example@viviti.com</email>
  <id type="integer">8</id>
  <package-id type="integer">2</package-id>
  <payment-due-date type="date">2010-11-17</payment-due-date>
  <security-hash>bd2e6cde672b2f493d1b4efb45cd4d1be1f30900</security-hash>
  <suspended type="boolean">>false</suspended>
  <trial type="boolean">>false</trial>
  <account-limits>
    <storage type="integer">21474836480</storage>
    <pages type="integer">15</pages>
    <websites type="integer">2</websites>
  </account-limits>
  <account-usage>
    <storage type="integer">468591</storage>
    <pages type="integer">5</pages>
    <websites type="integer">10</websites>
  </account-usage>
</account>
```

Update Account Package Limits

Update the account to use a new package. This will change the expiry date, and the new expiry date will be returned with the account details in the response. If term is passed, it will also renew the account for the number of months specified (after dealing with pro-rating).

HTTP Method: POST

URL: /account/1/update_package_limits

Required Parameters:

Parameter Name	Type	Description
security_hash	String	The security hash for the account.

Optional Parameters:

Parameter Name	Type	Description
<i>limitations[allowed_websites]</i>	Integer	Sets a limit on the account of how many websites it can create.
<i>limitations[allowed_storage]</i>	Integer	Sets a limit on the account of how much storage the site can use.
<i>limitations[allowed_pages]</i>	Integer	Sets a limit on the account of how many websites it can create.

Example Request:

```
POST /account/8/update_package_limits?
api_key=API_KEY&security_hash=bd2e6cde672b2f493d1b4efb45cd4d1be1f30900&limitations
[allowed_storage]=21474836480&limitations[allowed_websites]=2&limitations[allowed_pages]=
15
```

Example Response:

```
<account>
  <created-at type="datetime">2010-02-17T13:30:42-08:00</created-at>
  <email>example@viviti.com</email>
  <id type="integer">8</id>
  <package-id type="integer">2</package-id>
  <payment-due-date type="date">2010-11-17</payment-due-date>
  <security-hash>bd2e6cde672b2f493d1b4efb45cd4d1be1f30900</security-hash>
  <suspended type="boolean">>false</suspended>
  <trial type="boolean">>false</trial>
  <account-limits>
    <storage type="integer">21474836480</storage>
    <pages type="integer">15</pages>
    <websites type="integer">2</websites>
  </account-limits>
  <account-usage>
    <storage type="integer">468591</storage>
    <pages type="integer">5</pages>
    <websites type="integer">10</websites>
  </account-usage>
</account>
```

List Account Websites

Get a list of all the websites created under the account.

HTTP Method: GET

URL: /account/1/websites

Required Parameters:

Parameter Name	Type	Description
security_hash	String	The security hash for the account that owns this website.

Example Request:

```
GET
/account/8/websites?api_key=API_KEY&security_hash=bd2e6cde672b2f493d1b4efb45cd4d1be1f3090
0
```

Example Response:

```
<websites type="array">
  <website>
    <created-at type="datetime">2010-02-17T14:26:41-08:00</created-at>
    <id type="integer">11</id>
    <primary-host-id type="integer">25</primary-host-id>
    <suspended type="boolean">>false</suspended>
    <hosts type="array">
      <host>
        <id type="integer">11</id>
        <name>example.viviti.com</name>
      </host>
    </hosts>
  </website>
  ...
</websites>
```

Websites

Once an account is created, websites can be created under the account.

Overview

ID in the url refers to the ID of the website, which is returned when you create it

HTTP Method	URL	Description
POST	/website	Create a new website
POST	/website/ ID /delete	Delete a website
POST	/website/ ID /suspend	Suspends a website
POST	/website/ ID /unsuspend	Unsusponds a website
POST	/website/ ID /update_powered_by_html	Updates the powered by html in the footer of a web page.
POST	/website/ ID /update_allowed_languages	Updates which languages a user can use.
POST	/website/ ID /login	Log the user in, by obtaining a single use login url.
POST	/website/ ID /apply_theme	Apply a theme to the website
GET	/website/ ID	Return details for the website

Create Website

Creates a new website. The IP Address will be returned by this call, so that DNS can be correctly setup to ensure the host points at the new site.

HTTP Method: POST

URL: /website

Required Parameters:

Parameter Name	Type	Description
host	String	The hostname of the website you are creating.
account_id	Integer	The ID of the account you are creating a website for.
security_hash	String	The security hash for the account.

Example Request:

```
POST
/website?account_id=8&api_key=API_KEY&security_hash=bd2e6cde672b2f493d1b4efb45cd4d1be1f30900&host=example.viviti.com
```

Example Response:

```
<website>
  <created-at type="datetime">2010-02-17T14:36:26-08:00</created-at>
  <id type="integer">12</id>
  <primary-host-id type="integer" nil="true"></primary-host-id>
  <suspended type="boolean">>false</suspended>
  <hosts type="array">
    <host>
      <id type="integer">12</id>
      <name>example.viviti.com</name>
    </host>
  </hosts>
  <ip-address>127.0.0.1</ip-address>
</website>
```

Optional Parameters:

Parameter Name	Type	Description
powered_by_html	String	The html or text you want to insert into the bottom of the website.
allowed_languages	String	Comma delimited list of languages using two letter abbreviations.

Example Powered By Request:

POST

```
/website?account_id=8&api_key=API_KEY&security_hash=bd2e6cde672b2f493d1b4efb45cd4d1be1f30900&host=example.viviti.com&powered_by_html=%3Ca%20href=%22viviticms.com%22%3EVivitiCMS.com%3C/a%3E
```

Example Allowed Languages Request:

POST

```
/website?account_id=8&api_key=API_KEY&security_hash=bd2e6cde672b2f493d1b4efb45cd4d1be1f30900&host=example.viviti.com&allowed_languages=en
```

Update Powered By HTML

Updates the powered by html at the bottom of a website's footer.

HTTP Method: POST

URL: /website/1/update_powered_by_html

Required Parameters:

Parameter Name	Type	Description
<i>security_hash</i>	String	The security hash for the account that owns this website.
<i>powered_by_html</i>	String	The html displayed at the bottom of a website's footer.

Example Request:

POST

/website/1/update_powered_by_html?api_key=API_KEY&security_hash=bd2e6cde672b2f493d1b4efb45cd4d1be1f30900&powered_by_html=%3Ca%20href=%22viviticms.com%22%3EVivitiCMS.com%3C/a%3E

Update Allowed Languages

Comma delimited list of languages which are allowed when using the builder.

HTTP Method: POST

URL: /website/1/update_allowed_languages

Required Parameters:

Parameter Name	Type	Description
<i>security_hash</i>	String	The security hash for the account that owns this website.
<i>allowed_languages</i>	String	Comma delimited list of languages using two letter abbreviations. Example: en,fr,ar for english, french, arabic

Example Request:

POST

/website?account_id=8&api_key=API_KEY&security_hash=bd2e6cde672b2f493d1b4efb45cd4d1belf30900&host=example.viviti.com&allowed_languages=en

Delete Website

Permanently removes the website from our system.

Important: The website cannot be recovered after this call is used.

HTTP Method: POST

URL: /website/1/delete

Required Arguments:

Parameter Name	Type	Description
security_hash	String	The security hash for the account that owns this website.

Example Request:

```
POST
/website/12/delete?api_key=API_KEY&security_hash=bd2e6cde672b2f493d1b4efb45cd4d1be1f30900
```

Example Response:

```
<website>
  <created-at type="datetime">2010-02-17T14:36:26-08:00</created-at>
  <id type="integer">12</id>
  <primary-host-id type="integer">25</primary-host-id>
  <suspended type="boolean">>false</suspended>
  <hosts type="array">
    <host>
      <id type="integer">12</id>
      <name>example.viviti.com</name>
    </host>
  </hosts>
  <ip-address>127.0.0.1</ip-address>
</website>
```

Suspend Website

Suspends the website and makes it so that the message passed is displayed when attempting to visit the website.

HTTP Method: POST

URL: /website/1/suspend

Required Parameters:

Parameter Name	Type	Description
<i>security_hash</i>	String	The security hash for the account that owns this website.
<i>message</i>	String	The suspend message that will be displayed on the users websites.

Example Request:

```
GET
/account/8/suspend?api_key=API_KEY&security_hash=bd2e6cde672b2f493d1b4efb45cd4d1be1f30900
&message=Has%20not%20paid
```

Example Response:

```
<website>
  <created-at type="datetime">2010-02-17T14:36:26-08:00</created-at>
  <id type="integer">12</id>
  <primary-host-id type="integer">25</primary-host-id>
  <suspended type="boolean">>true</suspended>
  <hosts type="array">
    <host>
      <id type="integer">12</id>
      <name>example.viviti.com</name>
    </host>
  </hosts>
  <ip-address>127.0.0.1</ip-address>
</website>
```

Unsuspend Website

Unsuspects the website and makes it accessible again.

HTTP Method: POST

URL: /website/1/unsuspend

Required Arguments:

Parameter Name	Type	Description
security_hash	String	The security hash for the account that owns this website.

Example Request:

```
POST
/website/12/unsuspend?api_key=API_KEY&security_hash=bd2e6cde672b2f493d1b4efb45cd4d1be1f30900
```

Example Response:

```
<website>
  <created-at type="datetime">2010-02-17T14:36:26-08:00</created-at>
  <id type="integer">12</id>
  <primary-host-id type="integer">25</primary-host-id>
  <suspended type="boolean">>false</suspended>
  <hosts type="array">
    <host>
      <id type="integer">12</id>
      <name>example.viviti.com</name>
    </host>
  </hosts>
  <ip-address>127.0.0.1</ip-address>
</website>
```

Retrieve Website Details

Get details for the website.

HTTP Method: GET

URL: /website/1

Required Parameters:

Parameter Name	Type	Description
security_hash	String	The security hash for the account that owns this website.

Example Request:

```
GET /website/12?api_key=API_KEY&security_hash=bd2e6cde672b2f493d1b4efb45cd4d1be1f30900
```

Example Response:

```
<website>
  <created-at type="datetime">2010-02-17T14:36:26-08:00</created-at>
  <id type="integer">12</id>
  <primary-host-id type="integer">25</primary-host-id>
  <suspended type="boolean">>false</suspended>
  <hosts type="array">
    <host>
      <id type="integer">12</id>
      <name>example.viviti.com</name>
    </host>
  </hosts>
  <ip-address>127.0.0.1</ip-address>
</website>
```


Retrieve Login URL for Website

Get a single use redirect URL that allows the user to login and edit their website.

HTTP Method: POST

URL: /website/1/login

Required Parameters:

Parameter Name	Type	Description
security_hash	String	The security hash for the account that owns this website.

Example Request:

```
POST
/website/12/login?api_key=API_KEY&security_hash=bd2e6cde672b2f493d1b4efb45cd4d1be1f30900
```

Example Response:

```
<login>
```

```
<url>http://example.viviti.com/?website\_id=12&authkey=053048a46c5cf0156630cb8dc32bc909d54eda53</url>
```

```
</login>
```

Apply New Theme to Website

Apply a new theme to the website. This call will be made synchronous, so the theme zip can be deleted after the call is completed.

HTTP Method: POST

URL: /website/1/apply_theme

Required Parameters:

Parameter Name	Type	Description
theme_url	String	The URL of the theme zip file.
security_hash	String	The security hash for the account that owns this website.

Example Request:

```
POST
/website/12/apply_theme?theme_url=http%3A%2F%2Fexample.com%2Ftheme.zip&api_key=API_KEY&security_hash=bd2e6cde672b2f493d1b4efb45cd4d1be1f30900
```

Example Response:

```
<website>
  <created-at type="datetime">2010-02-17T14:36:26-08:00</created-at>
  <id type="integer">12</id>
  <primary-host-id type="integer">25</primary-host-id>
  <suspended type="boolean">>false</suspended>
  <hosts type="array">
    <host>
      <id type="integer">12</id>
      <name>example.viviti.com</name>
    </host>
  </hosts>
  <ip-address>127.0.0.1</ip-address>
</website>
```

Themes

Get a list of themes, store the id of the theme for subsequent API calls.

HTTP Method: GET

Example XML Request:

GET /themes.xml?api_key=API_KEY

Example XML Response

```
<?xml version="1.0" encoding="UTF-8"?>
<themes type="array">
  <theme>
    <author>Jigsy</author>
    <author-url>http://jigsy.com</author-url>
    <editable-header-image type="boolean">false</editable-header-image>
    <id type="integer">424</id>
    <name>blog crazy</name>
    <position type="integer">1</position>
    <responsive-layout type="boolean">true</responsive-layout>
  </theme>
</themes>
```

Example JSON Request:

GET /themes.json?api_key=API_KEY

Example JSON Response

```
[
  {
    "name": "blog crazy",
    "author": "Jigsy",
    "id": 424,
    "editable_header_image": false,
    "position": 1,
    "author_url": "http://jigsy.com",
    "responsive_layout": true
  }
]
```

Retrieve Theme Details

Get details for the theme, including the ID, name, position, editable_header_image, author, author_url and responsive_layout.

HTTP Method: GET

Example XML Request:

GET /themes/424.xml?api_key=API_KEY

Example XML Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<theme>
  <author>Jigsy</author>
  <author-url>http://jigsy.com</author-url>
  <editable-header-image type="boolean">false</editable-header-image>
  <id type="integer">424</id>
  <name>blog crazy</name>
  <position type="integer">1</position>
  <responsive-layout type="boolean">true</responsive-layout>
</theme>
```

Example JSON Request:

GET /themes/424.json?api_key=API_KEY

Example JSON Response:

```
{
  "name": "blog crazy",
  "author": "Jigsy",
  "id": 424,
  "editable_header_image": false,
  "position": 1,
  "author_url": "http://jigsy.com",
  "responsive_layout": true
}
```

Apply New Theme To Website, by Theme ID

Apply a new theme to the website passing in the ID of the theme. This call will be made synchronous, so the theme zip can be deleted after the call is completed.

HTTP Method: POST

URL: /website/1/apply_theme_by_id

Required Parameters:

Parameter Name	Type	Description
theme_id	String	The ID of the theme.
security_hash	String	The security hash for the account that owns this website.

Example Request:

POST

```
/website/12/apply_theme_by_id?theme_id=14&api_key=API_KEY&security_hash=bd2e6cde672b2f493d1b4efb45cd4d1be1f30900
```

Example Response:

```
<website>
  <created-at type="datetime">2010-02-17T14:36:26-08:00</created-at>
  <id type="integer">12</id>
  <primary-host-id type="integer">25</primary-host-id>
  <suspended type="boolean">>false</suspended>
  <hosts type="array">
    <host>
      <id type="integer">12</id>
      <name>example.viviti.com</name>
    </host>
  </hosts>
  <ip-address>127.0.0.1</ip-address>
</website>
```

Hosts

Once a website is created, hosts can be created under the website.

Overview

ID in the url refers to the ID of the host, which is returned when you create it

HTTP Method	URL	Description
POST	/host	Create a new host
POST	/host/ ID /delete	Deletes the host
POST	/host/ ID /set_primary	Sets the host as the primary one for the website.

Create a Host

Create a new host associated with a website.

HTTP Method: POST

URL: /host

Required Parameters:

Parameter Name	Type	Description
<i>host</i>	String	The name of the host you are creating.
<i>website_id</i>	Integer	The ID of the website you are creating a host for.
<i>security_hash</i>	String	The security hash for the account.

Required Parameters:

Parameter Name	Type	Description
<i>primary</i>	Boolean	Set new host as the primary host for the website. Default is false.

Example Request:

```
POST
/host?api_key=API_KEY&security_hash=bd2e6cde672b2f493d1b4efb45cd4d1be1f30900&website_id=1
2&host=example.com
```

Example Response:

```
<host>
  <id type="integer">13</id>
  <name>example.com</name>
  <website-id type="integer">12</website-id>
</host>
```

Validate a Host

Validate a host, this can be called before account creation or after.

HTTP Method: POST

URL: /host

Required Parameters:

Parameter Name	Type	Description
<i>name</i>	String	The name of the host you want to validate.

When the host name is valid:

Example Request:

```
POST /host/validate?api_key=API_KEY
```

Example Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<host>
  <valid type="boolean">true</valid>
</host>
```

When the host name is invalid:

Example Request:

```
POST /host/validate?api_key=API_KEY&name="example.com"
```

Example Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<errors>
  <error>Name has already been taken</error>
</errors>
```


Delete A Host

Deletes a host associated with a website.

HTTP Method: POST

URL: /host/1/delete

Required Parameters:

Parameter Name	Type	Description
website_id	Integer	The ID of the website you are deleting the host for.
security_hash	String	The security hash for the account.

Example Request:

```
POST
/host/delete/13?api_key=API_KEY&security_hash=bd2e6cde672b2f493d1b4efb45cd4d1be1f30900&website_id=12
```

Example Response:

```
<host>
  <id type="integer">13</id>
  <name>example.com</name>
  <website-id type="integer">12</website-id>
</host>
```

Set the host to primary

Sets the host to the primary on the website. This makes it so that all instances of the website within the builder will refer to the website by this host.

HTTP Method: POST

URL: /host/1/set_primary

Required Parameters:

Parameter Name	Type	Description
website_id	Integer	The ID of the website you are deleting the host for.
security_hash	String	The security hash for the account.

Example Request:

```
POST
/host/13/set_primary?api_key=API_KEY&security_hash=bd2e6cde672b2f493d1b4efb45cd4d1be1f30900&website_id=12
```

Example Response:

```
<host>
  <id type="integer">13</id>
  <name>example.com</name>
  <website-id type="integer">12</website-id>
</host>
```

Partners

Provides details specific to your partner account.

Overview

HTTP Method	URL	Description
GET	/partner/packages	List of available packages

List Available Packages

List of all the packages available

HTTP Method: GET

URL: /partner/packages

Required Parameters:

There are no required parameters for this action.

Example Request:

```
POST /partner/packages?api_key=API_KEY
```

Example Response:

```
<packages type="array">
  <package>
    <allow-custom-domains type="boolean">>false</allow-custom-domains>
    <allow-image-library type="boolean">>false</allow-image-library>
    <allow-passwords type="boolean">>false</allow-passwords>
    <allowed-pages type="integer">10</allowed-pages>
    <allowed-storage type="integer">5242880</allowed-storage>
    <allowed-websites type="integer">1</allowed-websites>
    <display-ads type="boolean">>false</display-ads>
    <id type="integer">1</id>
    <name>Starter Plan</name>
    <price-in-cents type="integer">1000</price-in-cents>
  </package>
  ...
</packages>
```

Email

Every email address in the system is owned by an email account. Packages and service limits are set on the email account and if the package allows multiple emails, are shared amongst all the email addresses on the email account.

When an email account is created a security hash will be returned which should be stored along with the email account id. All subsequent requests to do anything with this email account will require this security hash to be passed as a parameter.

Email accounts will be deactivated several weeks after their expiration date if the account has not been renewed.

Overview

ID in the URL refers to the ID of the email account, which is returned when you create it

HTTP Method	URL	Description
POST	/email	Create a new email account.
POST	/email/ ID /delete	Deletes an email account.
GET	/email/ ID	Get details for the email account.
POST	/email/ ID /renew	Renew the email account.
POST	/email/ ID /update_package	Set a new package for the email account.

Email Address

Every email address in the system is owned by an email account. The bold ID in the URL refers to the ID of the email address, which is returned when you create it

HTTP Method	URL	Description
POST	email/id/address	Create a new email address.
POST	email/id/address/ ID /delete	Deletes an email address.
GET	email/id/address/ ID	Get details for the address.
POST	email/id/address/ ID /suspend	Suspends address.
POST	email/id/address/ ID /unsuspend	Unsuspend address.
GET	email/id/addresses	Get a list of addresses for an email account.
GET	addresses	List all or search email addresses (returns an array of email addresses)

Email Alias

An email alias is simply a forwarding email address. To create an alias you will need to choose a source and destination email address.

HTTP Method	URL	Description
POST	email/id/alias	Create a new email alias.
POST	email/id/alias/ ID /delete	Deletes an email alias.
GET	email/id/alias/ ID	Get details for the alias.
GET	email/id/aliases	Get a list of aliases for an email account.

Domain Key

Domain Keys are an email validation method designed to detect email spoofing. Domain Keys provide a way for receiving mail exchangers to validate that incoming mail from a domain is authorized by that domain's administrators and that the email has not been modified during transport.

HTTP Method	URL	Description
POST	email/id/dkim	Create a new domain key.
POST	email/id/dkim/ ID /delete	Deletes a domain key.
GET	email/id/dkim/ ID	Get details for the domain key.
GET	email/id/dkims	Get a list of domain keys for an email account.

Domain Key Signature

A signature is required for the domain key, this can be created using the ID new domain key record.

HTTP Method	URL	Description
POST	email/id/dkim/id/dkim_signature	Create a new domain key signature.
POST	email/id/dkim/id/dkim_signature/ ID /delete	Deletes a domain key signature.
GET	email/id/dkim/id/dkim_signature/ ID	Get details for the domain key signature.
GET	email/id/dkim/id/dkim_signatures	Get a list of domain key signatures for a domain key.

Blacklist

Provides a list of blacklist values which have been identified by the support team.

List of blacklisted values by type

HTTP Method: GET

URL: /blacklist.xml or /blacklist.json

Required Parameters:

Parameter Name	Type	Description
type	String	Types include BlockedHostname, BlockedIp, BlockedLabel and BlockedEmail.

Optional Parameters:

Parameter Name	Type	Description
page	String	The current page number.
per_page	String	How many paged results returned.

Example Request:

```
GET /blacklist.xml?type=BlockedLabel&api_key=API_KEY&per_page=25&page=1
```

Example XML Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<blocked-labels type="array" current_page="1" per_page="5" total_entries="61">
  <blocked-label>
    <created-at type="datetime">2014-07-25T14:45:37-07:00</created-at>
    <id type="integer">200</id>
    <value>master\W*?card</value>
  </blocked-label>
  <blocked-label>
    <created-at type="datetime">2014-07-25T14:45:37-07:00</created-at>
    <id type="integer">199</id>
    <value>visa</value>
  </blocked-label>
  <blocked-label>
    <created-at type="datetime">2014-07-25T14:45:37-07:00</created-at>
    <id type="integer">198</id>
```

```
    <value>card\W*?number</value>
</blocked-label>
<blocked-label>
  <created-at type="datetime">2014-07-25T14:45:37-07:00</created-at>
  <id type="integer">197</id>
  <value>card\W*?type</value>
</blocked-label>
<blocked-label>
  <created-at type="datetime">2014-07-25T14:45:37-07:00</created-at>
  <id type="integer">196</id>
  <value>cvv</value>
</blocked-label>
</blocked-labels>
```

Example JSON Response:

```
[
  {
    "id": 205,
    "value": "Passphrase",
    "created_at": "2014/07/25 14:45:37 -0700"
  },
  {
    "id": 204,
    "value": "Salasana",
    "created_at": "2014/07/25 14:45:37 -0700"
  },
  {
    "id": 203,
    "value": "parola",
    "created_at": "2014/07/25 14:45:37 -0700"
  },
  {
    "id": 202,
    "value": "verification\\W*?code",
    "created_at": "2014/07/25 14:45:37 -0700"
  },
  {
    "id": 201,
    "value": "expiration\\W*?date",
    "created_at": "2014/07/25 14:45:37 -0700"
  }
]
```