

# Linktiger Partners REST API

## Table of Contents

<b>1. Introduction.....</b>	<b>2</b>
<b>1.2 REST API base URL .....</b>	<b>2</b>
<b>1.3 The RESTful web-service management.....</b>	<b>2</b>
<b>2. Accounts.....</b>	<b>3</b>
<b>2.1 Account resource.....</b>	<b>3</b>
<b>2.2 Account creation .....</b>	<b>3</b>
<b>2.3 Account deletion.....</b>	<b>5</b>
<b>2.4 Account modification.....</b>	<b>6</b>
<b>2.5 Auto login token generation.....</b>	<b>7</b>
<b>3. Tasks.....</b>	<b>9</b>
<b>3.1 Task resource.....</b>	<b>9</b>
<b>3.2 Task creation .....</b>	<b>10</b>
<b>3.3 Task deletion.....</b>	<b>11</b>
<b>3.4 Task modification.....</b>	<b>12</b>
<b>3.5 Querying .....</b>	<b>15</b>

# 1. Introduction

This API is intended for those Linktiger customers, who are allowed to use it. Those customers can be distinguished from other ones by enabling them in the Linktiger database by setting partner option enabled in the field **is\_partner\_api\_available** of the table **customer**.

These partners, granted with some extra privileges than other simple customers, are allowed to make some basic manipulations by means of the given RESTful service API, such as creating new account, deleting account, modification of the created account, to retrieve the account details and its dashboard tasks list, modifying them and adding some more extra ones.

## 1.2 REST API base URL

To make a call to the Linktiger REST API , partner should use a special base URL :

<http://linktiger.com/services/partner/api/1.0>

where **1.0** points to the number of version of the API and it is needed for migration from one API version to another.

## 1.3 The RESTful web-service management

Each request will pass HTTP Basic Authentication which is described in the RFC 2617 <http://www.ietf.org/rfc/rfc2617.txt>. All the requests will be responded by the HTTP status if they fail otherwise by the specified output format in the GET request URL. A connector in our REST service will represent an element that manages network communication for a component by implementing a network protocol HTTPS. A client connector initiates communication with a Linktiger server by creating a request. A server connector listens for connections (from clients of any kind), transmits the request to the component that performs the request processing, creates the response and sends it to the client. The format of the URI which will be sent by the partner will contain a root URL, resource and valid parameters for each case.

## 2.1 Account resource

For account manipulations we are using web resources:

- **user** – is the resource type, which represents account. The valid scheme: [/user/{email address}/](#)

The partner can use this type of resource in order to:

1. create account by sending PUT request with the valid input URI. For example creating account tiger@linktiger.com with the resource : [/user/tiger@linktiger.com](#)

In such way the new account will be created, but with no task

2. delete account by sending DELETE request with valid input URI. For example delete account tiger@linktiger.com with the resource: [/user/tiger@linktiger.com](#)

3. change subscription of the account by sending a POST request with URI. For example changing subscription of the tiger@linktiger.com account with the resource:

[/user/tiger@linktiger.com?subscription={subscription type}](#)

4. to get an auto login URL with token to the dashboard and account pages by sending a GET request with the URI. For example getting the auto login URL with token with the resource:

[/user/tiger@linktiger.com?destination={destination page}](#)

- **url** – is the resource type, which represents the task URL. The partner can use this type of the resource while creating a new account and adding the task URL to it at the same time. The valid scheme is:

[/user/{email address}/url/{defined url task}](#)

For example partner wants to create account **tiger@linktiger.com** with the task URL **http://www.website.com** for it, he should send PUT request according the the scheme:

[/user/tiger@linktiger.com/url/http://www.website.com](#)

## 2.2 Account creation

Makes possible to partner to create a new account with the specified subscription. In order to create a new subscription partner should send a PUT request followed by a valid URL with a valid format, where the whole chain will contain root URL, web resource path and parameters. Required REST Input Parameters for creating subscription are listed and described in the following table:

Parameter	Description	Type
subscription (Optional)	The subscription which the created account will obtain. (Represents the name of whole name of the subscription )	String

Here is the example of creating new subscription (For instance: we need to create account

under tiger@linktiger.com, to add http://www.website.com URL task and subscribe the account to Linkzilla subscription ):

```
https://linktiger.com/services/partner/api/1.0/

PUT /user/tiger@linktiger.com/url/http://www.website.com/?subscription=Linkzilla

-->201 Created
Connection: close
```

The partner can also specify only the value of the resource “email” in the URL, so then only the account will be created without any task on the dashboard. The following example show creating the only account with the Linkzilla subscription:

```
https://linktiger.com/services/partner/api/1.0/

PUT /user/tiger@linktiger.com/?subscription=Linkzilla

-->201 Created
Connection: close
```

The cases in which the partner will get some HTTP statuses by executing the command above are described in the table:

HTTP Status	Description
201 Created	The the user was created.
400 Bad Request	The partner didn't specify the correct parameter subscription, such subscription doesn't exist or email is incorrect
401 Unauthorized Explained	The partner authentication is not approved.
403 Forbidden	The parameters specified correctly, but such account or/and task already exists, or you are trying to add task to partner account, but not to the child one.
405 Method not allowed	Wrong method is specified in the HTTP request, for example PUT instead of POST.

As for the fact that the parameter “subscription” is optional, logically, if the partner specified the URL without this parameter, the Linktiger will automatically assign default subscription to the created customer. The password is generated automatically. If the url task for the customer is not predefined, then the customer will get an email template “REGISTRATION”, otherwise the first email will be sent after the first completed task with the template email “SITE REPORT”.

### 2.3 Account deletion

Makes possible to partner to cancel the existing subscription. In order to unsubscribe the specific account the partner needs to send a DELETE request followed by a valid URL in a valid format, where the whole chain will contain root URL, web resource path .

Here is the example of canceling subscription with valid HTTP basic partner authentication. The sample shows how to specify URL, web resource path in order to delete the preferred account from the Linktiger (For instance: we need to cancel account under name tiger@linktiger.com):

```
https://linktiger.com/services/partner/api/1.0/  
DELETE /user/tiger@linktiger.com -->200 OK  
  
Connection: close
```

The cases in which the partner will get some HTTP statuses by executing the command above are described in the table:

HTTP Status	Description
200OK	The request was approved
400 Bad Request	The partner didn't specify the correct 'email'( such account doesn't exist).
401 Unauthorized Explained	The partner authentication is not approved.
403 Forbidden	The specified resource is not valid. The parameters specified correctly, but there was attempt to delete the root partner account or account that is not a child of your partner one

## 2.4 Account modification

Makes possible to partner to change or upgrade from the existing to preferred subscription of the specific account. In order to change or upgrade the subscription for the specific account the partner needs to send a POST request followed by a valid URL in a valid format, where the whole chain will contain root URL, web resource path and parameter “subscription”.

Required REST Input Parameter for creating subscription is listed and described in the following table:

Parameter	Description	Type
subscription	The subscription which the account will be upgraded to. (Represents the name of whole name of the subscription )	String

The sample shows how to specify URL, web resource path and the parameter in order to change the subscription of the preferred account from the Linktiger (For instance: we need to upgrade account to the Professional subscription under name tiger@linktiger.com).

Here is the example of the account upgrading to a Professional subscription. The sample shows how to specify URL, web resource path and the parameters in order to change the subscription of the preferred account from the Linktiger (For instance: we need to upgrade account to the Professional subscription under name [tiger@linktiger.com](mailto:tiger@linktiger.com)):

```
https://linktiger.com/services/partner/api/1.0/
```

```
POST /user/tiger@linktiger.com/?subscription=Professional-->200 OK
```

```
Connection: close
```

The cases in which the partner will get some HTTP statuses by executing the command above are described in the table:

HTTP Status	Description
200OK	The request was approved.
400 Bad Request	The partner didn't specify the correct parameters.
401 Unauthorized Explained	The partner authentication is not approved.
403 Forbidden	The parameters are correctly specified , but such account doesn't exist, or the account is already subscribed to such subscription. Status is also returned in the case of attempt to modify the root partner account or account that is not a child of your partner one.

## 2.5 Auto login token generation

The partner can get the URL with special token, which enables him to auto-login to the created account's dashboard, or to the created account details page. In order to obtain the URL for account details page or to the account dashboard the partner should send the GET request followed by a valid URL in a valid format with the valid parameters, where the whole chain will contain root URL, web resource path and parameter “destination”. Each time of the request the token will be newly generated and will differ from the previous one, in order not to have the same URL for the auto login.

Required REST Input Parameter for getting URL with generated token for the auto login is listed and described in the following table:

Parameter	Description	Type
destination	Represents the destination page of the created account. Currently the supported values are: <a href="#">dashboard</a> and <a href="#">account</a>	String

The sample shows how to specify URL, web resource path and the parameter in order to get URL with generated token for the dashboard page of the created by the API account (For instance: we desire to get the URL with automatic login for the account under email [tiger@linktiger.com](mailto:tiger@linktiger.com)):

Here is the example, which shows how to specify URL, web resource path and the parameter in order to get the URL to the dashboard of the preferred account (For instance: we need to the auto login URL to the dashboard page of the account under the email - [tiger@linktiger.com](mailto:tiger@linktiger.com)):

https://linktiger.com/services/partner/api/1.0/

GET /user/tiger@linktiger.com/?destination=dashboard-->301 Redirect

Location: <http://linktiger.com/services/services//dashboard?key=5447205f309a08ff613728ec08334eac&random=1269953762049&callback=tiger.module.parseDashboardView>

Connection: close

The possible HTTP status codes can be obtained while processing the described request are listed in the following table:

HTTP Status	Description
301 Redirect	The request was approved, and we are redirecting with auto login to portlet.
400 Bad Request	The partner didn't specify the correct format of the request URL, parameter is invalid.
401 Unauthorized Explained	The partner authentication is not approved.
403 Forbidden	Such account doesn't exist or there is attempt to login into root partner account or account that is not it's child

## 3. Tasks

### 3.1 Task resource

The web resources which are used in order to make the manipulations over the existing tasks on the specific account are:

- **user** – resource type which represents customer account, created by the partner, which can give us the whole information about its tasks, their configurations.
- **url** - resource type which represents the tasks of the account. The valid scheme is:

[/user/{email address}/url/{task url}](#)

Partner can use this resource in order to:

1. create additional task url for the account by sending a PUT request. For example creation of the task <http://www.website.com> for the account [tiger@linktiger.com](mailto:tiger@linktiger.com) will look like:

[/user/tiger@linktiger.com/url/http://www.website.com](#)

2. delete specific task from the account by sending a DELETE request. For example deletion of the task <http://www.website.com> from the account [tiger@linktiger.com](mailto:tiger@linktiger.com) will look like:

[/user/tiger@linktiger.com/url/http://www.website.com](#)

3. modify the configurations of the specific task by sending the POST request with URI which will include the valid parameters. For example excluding PDF files from the task url <http://www.website.com> checking of the [tigr@linktiger.com](mailto:tigr@linktiger.com) account will look like:

[/user/tiger@linktiger.com/url/http://www.website.com?excludes=pdf](#)

4. retrieve the information from the dashboard including the information about the account tasks. It is also possible to retrieve the information about single specific task. This can be done by sending a GET request. For example retrieving the whole information from the [tiger@linktiger.com](mailto:tiger@linktiger.com) account dashboard page in a XML format will look like:

[/user/tiger@linktiger.com/url/?format=xml](#)

For example to retrieve the information about single specific task <http://www.website.com> of the [tiger@linktiger.com](mailto:tiger@linktiger.com) account in JSON format will look like:

[/user/tiger@linktiger.com/url/http://www.website.com?format=json](#)

### 3.2 Task creation

Makes possible to a partner to add extra URLs to the existing subscription. In order to add extra URLs for the specific account the partner needs to send a PUT request followed by a valid URL in a valid format, where the whole chain will contain root URL, web resource path and parameters.

Input URI:

[/user/{email address}/url/{task url}](#)

Example of the adding extra URL task <http://www.website.com> to the account tiger@linktiger.com:

```
https://linktiger.com/services/partner/api/1.0/  
  
PUT /user/tiger@linktiger.com/url/http://www.website.com -->200 OK  
Connection: close
```

The cases in which the partner will get some HTTP statuses by executing the command above are described in the table:

HTTP STATUS	Description
200OK	The request was approved, the task url was successfully added.
400 Bad Request	The partner didn't specify the correct format of the task URL.
401 Unauthorized Explained	The partner authentication is not approved.
403 Forbidden	Such account doesn't exist, or such task already exists. User attempts to create task using root partner account or account that is not a child of it.
405 Method not allowed	Wrong method is specified in the HTTP request, for example PUT instead of POST.

### 3.3 Task deletion

Makes possible to partner to delete the specific task on the created account. In order to delete the existing task, the partner needs to send DELETE request followed by a valid URL in a valid format, where the whole chain will contain root URL, web resource path

Here is the example of deleting task url from the specific account with the valid HTTP basic partner authentication. The sample shows how to specify URL, web resource path in order to delete the preferred task url which belongs to one the created by the API account from the Linktiger (For instance: we need to delete task url <http://www.website.com> under account name [tiger@linktiger.com](mailto:tiger@linktiger.com)):

```
https://linktiger.com/services/partner/api/1.0/
```

```
DELETE /user/tiger@linktiger.com/url/http://www.website.com -->200 OK
```

```
Connection: close
```

The cases in which the partner will get some HTTP statuses by executing the command above are described in the table:

HTTP STATUS	Description
200OK	The request was approved.
400 Bad Request	The partner didn't specify the correct 'email'( such account doesn't exist) or such task url doesn't exist for the deletion.
401 Unauthorized Explained	The partner authentication is not approved.
403 Forbidden	The specified resource is not valid. User attempts to delete task using root partner account or account that is not a child of it.

### 3.4 Task modification

The partner is allowed to modify the existing task URLs of the created account by the API. The partner should send a POST request followed by a valid URL in a valid format, where the whole chain will contain root URL, web resource path and valid parameters.

Required REST Input Parameter for modifying the specific task URL configurations are described in the following table:

Parameter	Description	Type
<b>excludes (Optional)</b>	In case and the parameter is defined it will indicate the file pattern to exclude from the report, each pattern can be separated by comma	String
<b>includes (Optional)</b>	In case and the parameter is defined it will indicate the file pattern to include into the report, each pattern can be separated by comma	String
<b>status (Optional)</b>	The HTTP status codes to include into the report, the redirect HTTP status codes are ignored. The values are the integers separated by comma.	List of integers separated by comma
<b>user (Optional)</b>	In case the server requires authentication (for example FTP), partner can define username for the authentication	String
<b>passwd (Optional)</b>	In case the server requires authentication (for example FTP), partner can define password for the authentication	String
<b>forms (Optional)</b>	Indicates if the forms should be submitted on the site while processing the task. By default the forms are not submitted while task processing.	Boolean
<b>redirect (Optional)</b>	In case the value is set to true indicates to mark the found redirect urls in the report as broken link, otherwise as valid	Boolean
<b>use_robots (Optional)</b>	In case the value is set to true crawler will respect the robots.txt during site crawling.	Boolean

Parameter	Description	Type
report_condition (Optional)	Indicates when email report will be send to user. Should take one of these values: “on_errors”, “after_check” and “no_report”	String
email_send_interval (Optional)	Parameter indicates how often report should be sent to user - “daily” or “weekly”.	String

In case there were no any parameters defined, the request will create the defined task in the URL in case that it doesn't exist in the tasks list on the account's dashboard. The parameters are marked as optional , if some of the parameters are not defined the task obtains the previous configuration.

The example show how to modify URL task <http://www.website.com> configurations to the under the account [tiger@linktiger.com](mailto:tiger@linktiger.com):

1. *Include into the report the following file extensions: .JPG, .PDF, .PHP .*
2. *Exclude from the report following file extensions: .ASP, .TXT, .CSS .*
3. *To find broken links only with the following HTTP status codes: 400, 404.*
4. *To mark redirects as broken links.*
5. *Use robots.txt when crawling website.*
6. *Send email report after site checking.*
7. *Send email report weekly.*

```
https://linktiger.com/services/partner/api/1.0/
```

```
POST /user/tiger@linktiger.com/url/http://www.website.com/?
includes=jpg,pdf,php&
excludes=asp,txt,css&
status=400,404&
forms=true&
redirect=true&
use_robots=false&
report_condition=after_check&
email_send_interval=weekly -->200 OK
```

```
Connection: close
```

There is a way to clear some parameter value. To perform this action the empty parameter value should be specified. For example:

```
https://linktiger.com/services/partner/api/1.0/
```

```
POST /user/tiger@linktiger.com/url/http://www.website.com/?  
includes=&  
excludes=&  
user=&  
passwd= -->200 OK
```

```
Connection: close
```

The cases in which the partner will get some HTTP statuses by executing the command above are described in the table:

HTTP STATUS	Description
200OK	The request was approved, the task was configured by the defined parameters.
400 Bad Request	Parameter is invalid, the partner didn't specify the correct 'email'( such account doesn't exist) or such task URL doesn't exist for the modification.
401 Unauthorized Explained	The partner authentication is not approved.
403 Forbidden	The specified resource is not valid or user attempts to modify task using root partner account or account that is not a child of it.

### 3.5 Querying

The partner is likewise allowed to track all the dashboard information of the specific account like:

1. To get all the URL tasks and URL to the report pages of each task
2. To get the amount of the broken, processed and valid links from each URL task
3. To get the information about the upgrade account and to retrieve URL to the upgrade page in the Portlet if such suggestion exists on the dashboard

The partner will send a GET request with the valid url, web resource path and parameters .

Required REST Input Parameters for creating subscription are listed and described in the following table:

Parameter	Description	Type
<b>format</b>	We can specify the format of the response to be returned. It can be XML, JSON or NVP.	String
<b>config (Optional)</b>	If case and this parameter is defined and its value is set to 'true', the response will also contain the configurations for each returned task URL, by default the value is set to false	Boolean

*The value of the “url” resource is optional, if it is not specified, the partner will get the whole information about all the customer's tasks.*

Here is the example of extracting all the dashboard information of the account tiger@linktiger.com. The sample shows how to specify URL, web resource path and the parameters in order to get all the dashboard information of the preferred account from the Linktiger (For instance: we need to know the state for the account under name tiger@linktiger.com in XML format, we specify to extract also the information for one only task http://www.website.com):

https://linktiger.com/services/partner/api/1.0/

GET /user/tiger@linktiger.com/url/http://www.website.com/?format=xml

-->200 OK

Content-Length: 1004

Content-Type: text/xml; charset=utf-8

<?xml version="1.0" encoding="UTF-8" ?>

<dashboard>

<totals>

<validated>852</validated>

<broken>239</broken>

<estimated>852</estimated>

</totals>

<subscription>

<current>Free</current>

<recommended>Small</recommended>

<upgrade\_url>

http://linktiger.com/services/portlet\_redirect.jsp?url=

http%3A%2F%2Fapp.linktiger.redwerk.com%2Fservices%2Fservices%2F

account%2F%3Fkey%3D0d7d8b8ee84d49b674363768bc80f396%26

callback%3Dtiger.module.parseAccountDetails

</upgrade\_url>

</subscription>

<websites>

<website url="http://www.website.com">

<totals>

<validated>852</validated>

<estimated>852</estimated>

<broken>239</broken>

</totals>

<last\_checked>Apr 01, 2010</last\_checked>

<report\_url>

http://linktiger.com/services/portlet\_redirect.jsp?

url=http%3A%2F%2Fapp.linktiger.redwerk.com

%2Fservices%2Fservices%2Fstandardreports%3Fkey%3D0d7d8b8ee84d49b67

4363768bc80f396%26website%3D4410%26report\_id%3D1%26callback%3D

tiger.module.parseReportResults

</report\_url>

</website>

</websites>

</dashboard>

Connection: close

The JSON of the same XML output will have a look :

```
https://linktiger.com/services/partner/api/1.0/

GET /user/tiger@linktiger.com/url/http://www.website.com/?format=json
-->200 OK

{"dashboard": {
  "totals": {
    "validated": 852,
    "broken": 239,
    "estimated": 852
  },

  "subscription": {
    "current": "Free",
    "recommended": "Small",
    "upgrade_url": "http://linktiger.com/services/portlet_redirect.jsp?url=http%3A%2F%2Flinktiger.com%2Fservices%2Fservices%2Faccount%2F%3Fkey%3D1bc5ec582f0770789984a3a3f1eb79fb%26callback%3Dtiger.module.parseAccountDetails"
  },
  "websites": [
    {
      "@url": "http://www.website.com",
      "totals": {
        "validated": "852",
        "estimated": "852",
        "broken": "239"
      },
      "last_checked": "Apr 01, 2010",
      "report_url": "http://linktiger.com/services/portlet_redirect.jsp?url=http%3A%2F%2Flinktiger.com%2Fservices%2Fservices%2Fstandardreports%3Fkey%3Dd1ec158e586087060945e999224a350f%26website%3D4410%26report_id%3D1%26callback%3Dtiger.module.parseReportResults"
    }
  ]
}}

Connection: close
```

And NVP output will have a look:

```
https://linktiger.com/services/partner/api/1.0/
```

```
GET /user/tiger@linktiger.com/url/http://www.website.com/?format=nvp  
-->200 OK
```

```
Validated=852,852
```

```
&broken=239,239
```

```
&estimated=852,852
```

```
&current=Free&recommended=Small
```

```
&upgrade_url=http://linktiger.com/services/portlet_redirect.jsp?
```

```
url=http%3A%2F%2Flinktiger.com%2Fservices%2Fservices%2Faccount
```

```
%2F%3Fkey%3D8c576fef3d4a1d4fe7ff1fc9f35915fb%26callback%3D
```

```
tiger.module.parseAccountDetails
```

```
&website_url=http://www.website.com&last_checked="Apr 02, 2010"
```

```
&report_url=http://linktiger.com/services/portlet_redirect.jsp?
```

```
url=http%3A%2F%2Flinktiger.com%2Fservices%2Fservices%2F
```

```
standardreports%3Fkey%3D8c576fef3d4a1d4fe7ff1fc9f35915fb%26website%3D4410%26
```

```
report_id%3D1%26callback%3Dtiger.module.parseReportResults
```

```
Connection: close
```

As for that fact that the value of the “**url**” resource is *optional*, so if not specifying it, the partner will get the information about all the customer's tasks from his dashboard. The example shows how to extract all the information about all the customer's tasks:

https://linktiger.com/services/partner/api/1.0/

GET /user/tiger@linktiger.com/url/?format=xml

-->200 OK

```
<?xml version="1.0" encoding="UTF-8" ?>
<dashboard>
<totals>
  <validated>1704</validated>
  <broken>478</broken>
  <estimated>1704</estimated>
</totals>
<subscription>
  <current>Free</current>
  <recommended>Small</recommended>
  <upgrade_url>
    http://linktiger.com/services/portlet_redirect.jsp?url=
    http%3A%2F%2Fapp.linktiger.redwerk.com%2Fservices%2Fservices%2F
    account%2F%3Fkey%3D0d7d8b8ee84d49b674363768bc80f396%26
    callback%3Dtiger.module.parseAccountDetails
  </upgrade_url>
</subscription>
<websites>
  <website url="http://www.website.com">
  <totals>
    <validated>852</validated>
    <estimated>852</estimated>
    <broken>239</broken>
  </totals>
  <last_checked>Apr 01, 2010</last_checked>
  <report_url>
    http://linktiger.com/services/portlet_redirect.jsp?
    url=http%3A%2F%2Flinktiger.com
    %2Fservices%2Fservices%2Fstandardreports%3Fkey%3D0d7d8b8ee84d49b67
    4363768bc80f396%26website%3D4410%26report_id%3D1%26callback%3D
    tiger.module.parseReportResults
  </report_url>
</website>
<websites>
  <website url="http://www.website1.com">
  <totals>
    <validated>852</validated>
    <estimated>852</estimated>
    <broken>239</broken>
  </totals>
  <last_checked>Apr 01, 2010</last_checked>
  <report_url>
    http://linktiger.com/services/portlet_redirect.jsp?
    url=http%3A%2F%2Flinktiger.com
    %2Fservices%2Fservices%2Fstandardreports%3Fkey%3D0d7d8b8ee84d49b67
    4363768bc80f396%26website%3D4410%26report_id%3D1%26callback%3D
    tiger.module.parseReportResults
  </report_url>
</website>
</websites>
</dashboard>
```

Connection: close

Here is another one example, which describes how to retrieve the information about the specific task and its configurations. In order to get the specific task configurations, the request URL should additionally have the **config** parameter set to true.

Input URI:

</user/{email address}/url/{specific url}/?config=true&format={format of the response}>

The response is represented in the XML format:

```
https://linktiger.com/services/partner/api/1.0/
GET /user/tiger@linktiger.com/url/http://www.website.com/?format=xml&config=true
-->200 OK

<?xml version="1.0" encoding="UTF-8" ?>
<dashboard>
<totals>
  <validated>852</validated>
  <broken>239</broken>
  <estimated>852</estimated>
</totals>
<subscription>
  <current>Free</current>
  <recommended>Small</recommended>
  <upgrade_url>
    http://linktiger.com/services/portlet_redirect.jsp?url=
    http%3A%2F%2Flinktiger.com%2Fservices%2Fservices%2F
    account%2F%3Fkey%3D0d7d8b8ee84d49b674363768bc80f396%26
    callback%3Dtiger.module.parseAccountDetails
  </upgrade_url>
</subscription>
<websites>
  <website url="http://www.website.com">
  <totals>
    <validated>852</validated>
    <estimated>852</estimated>
    <broken>239</broken>
  </totals>
  <last_checked>Apr 01, 2010</last_checked>
  <includes/>
  <excludes/>
  <submit_forms>>false</submit_forms>
  <follow_redirects>>false</follow_redirects>
  <error_codes>
    1,411,400,412,401,413,402,414,403,415,404,416,405,417,
    406,500,407,501,408,502,409,503,410,504,505
  </error_codes>
  <use_robots>>false</use_robots>
  <email_send_interval>daily</email_send_interval>
  <report_condition>after_check</report_condition>
  <report_url>
    http://linktiger.com/services/portlet_redirect.jsp?
    url=http%3A%2F%2Flinktiger.com
    %2Fservices%2Fservices%2Fstandardreports%3Fkey%3D0d7d8b8ee84d49b67
    4363768bc80f396%26website%3D4410%26report_id%3D1%26callback%3D
    tiger.module.parseReportResults
  </report_url>
</website>
</websites>
</dashboard>

Connection: close
```

The element “**total**” will contain the information about the total summary for the tasks. Its sub-element properties are described in the following table:

Name	Value
validated	Will return the value of the summary of the validated links for the whole URL tasks
broken	Will return the value of the summary of the broken links for the whole URL tasks
estimated	Will return the value of the summary of the estimated links for the whole URL tasks

The element “**upgrade**” will be returned with value of the upgrade subscription, in case there exists an upgrade subscription suggestion on the account dashboard. In case there is no any suggestion for an account upgrade, those fields are not included in the format response.

The element “**task**” will be returned with the whole information about the account existing tasks. Each “**task**” element in the XML format will contain property elements which values and names are described in the following table:

Name	Value
validated	The amount of the validated links
broken	The amount of the broken links
estimated	The amount of the estimated links
last_checked	Date of the last check
excludes	The file pattern to exclude from the report
includes	The file pattern to include in the report
submit_forms	If the value is true then the forms are submitted on the site while task processing, otherwise are not submitted
username	If the task configured to process with the username , then the username value returned from the 'Secure Area'
password	If the task configured to process with the username and the password is applied to it, then the password value is returned from the 'Secure Area'
error_codes	Error types to be reported for the task
follow_redirects	If set to true, then the task processing marks found redirects as broken in the report, otherwise as a valid links
upgrade_url	The URL to the upgrade subscription page, in case the there is a suggestion to upgrade
report_url	The URL to the full report page of the specific task

Name	Value
use_robots	If the value is true, crawler will respect robots.txt during site crawling
email_send_interval	Shows how often email reports should be sent to user - "daily" or "weekly".
report_condition	Indicates when email report will be send to user. Should take one of these values: "on_errors", "after_check" and "no_report"

In case some task were not processed, the amounts of broken, processed and valid links will be 0. In case there are no any tasks bounded to the account, the element is omitted. In case the answer was approved, the partner will get an HTTP status 200 OK and the response in the specified format, otherwise the partner didn't specify the correct parameters or specified a wrong response format, he will obtain an HTTP status of 400 Bad Request. In case that the parameters specified correctly, but the partner authentication is not approved or such account doesn't exist, he will get an HTTP status of 403 Forbidden.

HTTP Status	Description
200OK	The request was approved.
400 Bad Request	The partner didn't specify the correct format of the task URL or parameter is invalid
401 Unauthorized Explained	The partner authentication is not approved.
403 Forbidden	Such account doesn't exist, such task doesn't exists or user attempts to browse dashboard data of the root partner account or account that is not a child of it.