KIWIRE 2.0 API Documentation

Version 1.0.0 (February 2017)

Implementation Notes

1) The Licensee, e.g. Client of Synchroweb (M) Sdn Bhd, including this API, has the right to use or subcontract the use of this API for work related to their implementation and data.
2) The Licensee will receive a new API account, e.g. authentication key, upon request. Each subcontractor shall use a unique account in all their usage of this API.
3) Each and all distinct projects using this API must be approved and quality assured by Synchroweb prior to implementation and launch, including projects developed internally by Licensee. Synchroweb will make recommendations to change functionalities that are deemed to adversely affect performance and security of any and all Synchroweb related system or services. In the case where the API service is running on a server owned and controlled by Synchroweb, such recommendations must be strictly adhered to.
4) Synchroweb quality assurance and assistance for each and all API projects is not covered by any prior agreements between Synchroweb and Licensee, and a quotation for such services must be requested prior to implementation. Synchroweb will study project scopes and make an assessment of the work deemed reasonable to assure project quality.
5) Synchroweb support to Licensee is restricted to proper working of the functions within this API. Any usage of such functions by Licensee and Subcontractors, where the functions are deemed by Synchroweb to be working in accordance with the specification in this document, is under the sole responsibility and risk by the Licensee. Any loss or damage to Licensee data under such conditions are not the responsibility of Synchroweb, and as such the Licensee shall not hold Synchroweb liable.
6) Synchroweb will assist to best ability in case of damage recovery, and Licensee shall compensate Synchroweb for such work by prior agreed daily service rate, or of no such agreements exist, by a rate quoted by Synchroweb when such damage recovery is ordered by Licensee.

7) All API projects shall be documented in scope and functionality by Licensee and Subcontractor before starting any implementation. These documentation shall be submitted to, and approved by Synchroweb as part of quality assurance project ordered in conjunction with the project. Such documentation shall contain:
    a) Project description and related use cases, e.g. what is the intended usage of this project.
    b) Technology description, e.g. what programming languages and other technologies will be used in this project. This also includes hardware and computers involved in the operations and specifications of the network connections used by any implemented API clients.
    c) List of API functions that are planned to be used in the project, together with frequency estimation, e.g. how frequent are the functions expected to be called.
8) As part of the quality assurance project, Synchroweb will assist Licensee or Subcontractor in producing the above mentioned documentation, by answering questions and provide knowledge, but not in the actual production of such documents.
9) If training is requested, Synchroweb will include such request in the quotation for quality assurance project.

Version History

All changes of this document are recorded here.

Version	**Changes / Additions**
1.0	Create document.

General preface

The KIWIRE API allow 3rd party software to access data gathered by KIWIRE 2.0 platform

Note:

- In this document, we will be using following URL as API URL: **http://192.168.0.209/api/**
- Token is the api key set in the KIWIRE API KEY
- All request must be sent via POST request. Call made via GET,DELETE,PUT will fail. You must provide token for each request for authentication purpose.
- All output are in JSON format.

Response Codes

Errors are returned using standard HTTP code syntax. Response with errors might indicate more error details.

Example of Standard API return codes:

Code	Text	Description
106	Forbidden	Requested method is not allowed.
500	Internal Server Error	Something is broken.
400	Bad Request	Request is invalid and cannot be served.
401	Unauthorized	Authentication credentials are missing or invalid.
404	Not Found	Requested data does not exist.
406	Not Acceptable	Request has wrong format or missing info.
200	Success	Request is successfully processed.

Function

All functions are accessed by url which is

http://192.168.0.209/api/<module>/<method>/

Below are the functions supported by Kiwire and the details

1. User

Action: Add user

URL: http://192.168.0.209/api/user/add/

Required Arguments:

Parameter	Description
token	Kiwire API key
username	Username for the user. A string.
password	Password for the user.
profile	Link to which profile.
email	Email for the user.
fullname	Fullname for the user.
expiry	For user account expiry. For example 3 days fro registered date, just put only 3. A Integer.

Example Output:


```
"result": [
  {
    "status": true,
    "message": "Successfully add new user
    "reference": {
      "username": "Maria"
    }
  }
]
```


Output	Values
status	Boolean. If request is success, it will return as true. Otherwise false.
message	String. If the request is success, it will return the success message. Otherwise error message.
reference/data	JSON. Use for further reference.

Action: Edit user

URL: http://192.168.0.209/api/user/edit/

Required Arguments:

Parameter	Description
token	Kiwire API key.
username	Current username for the user. Note that username cannot be change once the account have been created. The purpose required this params is for to find which account to be update.
fullname	Fullname for the user.
email	Email for the user.
password	Password for the user.
profile	To link the user with the profile.
expiry	To set expired date for the user account.

Example Output:


```
{
  "result": [
    {
      "status": true,
      "message": "Successfully update user",
      "reference": {
        "username": "Maria"
      }
    }
  ]
}
```


Output	Values
status	Boolean. If request is success, it will return as true. Otherwise false.
message	String. If the request is success, it will return the success message. Otherwise error message.
reference/data	JSON. Use for further reference.

Action: Delete user

URL: http://192.168.0.209/api/user/delete/

Required Arguments:

Parameter	Description
token	Kiwire API key.
username	Username for the user.

Example Output:


```
{
  "result": [
    {
      "status": true,
      "message": "Successfully delete user",
      "reference": {
        "username": "Maria"
      }
    }
  ]
}
```


Output	Values
status	Boolean. If request is success, it will return as true. Otherwise false.
message	String. If the request is success, it will return the success message. Otherwise error message.
reference/data	JSON. Use for further reference.

Action: List user

URL: http://192.168.0.209/api/user/list/

Required Arguments:

Parameter	Description
token	Kiwire API key.

Example Output:


```
{
  "result": [
    {
      "status": true,
      "message": "Successfully list all user records",
      "data": [
        {
          "username": "John",
          "profile": "testing",
          "email": "john.doe@gmail.com",
          "fullname": "John Doe",
          "expiry": "2017-03-04"
        }
      ]
    }
  ]
}
```


Output	Values
status	Boolean. If request is success, it will return as true. Otherwise false.
message	String. If the request is success, it will return the success message. Otherwise error message.
reference/data	JSON. Use for further reference.

2. Administrator

Action: Add administrator

URL : http://192.168.0.209/api/admin/add/

Required Arguments:

Parameter	Description
token	Kiwire API key.
username	Username for the user.
password	Password for the admin.
fullname	Fullname for the admin.
email	Email for the admin.
groupname	To group admin based on group have been created.
monitor	Put the value as y or n.

Example Output:


```
{
  "result": [
    {
      "status": true,
      "message": "Successfully add new administrator",
      "reference": {
        "username": "MyAdmin"
      }
    }
  ]
}
```


Output	Values
status	Boolean. If request is success, it will return as true. Otherwise false.
message	String. If the request is success, it will return the success message. Otherwise error message.
reference/data	JSON. Use for further reference.

Action: Edit administrator

URL : http://192.168.0.209/api/admin/edit/

Required Arguments:

Parameter	Description
token	Kiwire API key.
username	Current username for the admin. Note that username cannot be change once the account have been created. The purpose required this params is for to find which account to be update.
fullname	Fullname for the admin.
email	Email for the admin.
groupname	To group admin based on group have been created.
monitor	Put the value as y or n.
password	Password for the admin.

Example Output:


```
{
  "result": [
    {
      "status": true,
      "message": "Successfully update administrator",
      "reference": {
        "username": "MyAdmin"
      }
    }
  ]
}
```


Output	Values
status	Boolean. If request is success, it will return as true. Otherwise false.
message	String. If the request is success, it will return the success message. Otherwise error message.
reference/data	JSON. Use for further reference.

Action: Delete administrator

URL : http://192.168.0.209/api/admin/delete/

Required Arguments:

Parameter	Description
token	Kiwire API key.
username	Username for the user.

Example Output:


```
{
  "result": [
    {
      "status": true,
      "message": "Successfully delete administrator",
      "reference": {
        "username": "MyAdmin"
      }
    }
  ]
}
```


Output	Values
status	Boolean. If request is success, it will return as true. Otherwise false.
message	String. If the request is success, it will return the success message. Otherwise error message.
reference/data	JSON. Use for further reference.

Action: List administrator

URL : http://192.168.0.209/api/admin/list/

Required Arguments:

Parameter	Description
token	Kiwire API key.

Example Output:


```
{
  "result": [
    {
      "status": true,
      "message": "Successfully list all admin records",
      "data": [
        {
          "username": "admin",
          "fullname": "Administrator",
          "email": "john.doe@gmail.com",
          "groupname": "operator",
          "monitor": "y"
        }
      ]
    }
  ]
}
```

3. Device

Action: Add device nas

URL : http://192.168.0.209/api/device/{spesific device}/add/

Note:

{spesific device}. You must mention the device type. For example, nas.

You simply put http://192.168.0.209/api/device/nas/add/.

Required Arguments:

Parameter	Description
token	Kiwire API key.
username	Username for the nas.
password	Password for the nas.
nasname	IP address of the NAS/device must be unique.
shortname	Name must be unique. Note: Mikrotik Nas, please use Hotspot Server Name.
location	Address location for the nas.
secret	Shared secret key.
ports	COA port.
description	Description
device_type	Eg: Miktrotik.

Example Output:


```
{
  "result": [
    {
      "status": true,
      "message": "Successfully add nas record to system",
      "reference": {
        "nasname": "0.02.0.7",
        "shortname": "testapi"
      }
    }
  ]
}
```


Output	Values
status	Boolean. If request is success, it will return as true. Otherwise false.
message	String. If the request is success, it will return the success message. Otherwise error message.
reference/data	JSON. Use for further reference.

Action: Edit device nas

URL : http://192.168.0.209/api/device/{spesific device}/edit/

Note:

{spesific device}. You must mention the device type. For example, nas.

You simply put http://192.168.0.209/api/device/nas/edit/.

Required Arguments:

Parameter	Description
token	Kiwire API key.
id	ID for nas.
username	Username for the nas.
password	Password for the nas.
nasname	IP address of the NAS/device must be unique.
shortname	Name must be unique. Note: Mikrotik Nas, please use Hotspot Server Name.
location	Address location for the nas.
secret	Shared secret key.
ports	COA port.
description	Description
device_type	Eg: Miktrotik.

Example Output:


```
{
  "result": [
    {
      "status": true,
      "message": "Successfully updated nas",
      "reference": {
        "id": "68",
        "nasname": "0.02.0.7"
      }
    }
  ]
}
```


Output	Values
status	Boolean. If request is success, it will return as true. Otherwise false.
message	String. If the request is success, it will return the success message. Otherwise error message.
reference/data	JSON. Use for further reference.

Action: Delete device nas

URL : http://192.168.0.209/api/device/{spesific device}/delete/

Note:

{spesific device}. You must mention the device type. For example, nas.

You simply put http://192.168.0.209/api/device/nas/delete/.

Required Arguments:

Parameter	Description
token	Kiwire API key.
id	ID for nas.

Example Output:


```
{
  "result": [
    {
      "status": true,
      "message": "Successfully delete nas",
      "reference": {
        "id": "68",
        "nasname": "0.03.0.8"
      }
    }
  ]
}
```


Output	Values
status	Boolean. If request is success, it will return as true. Otherwise false.
message	String. If the request is success, it will return the success message. Otherwise error message.
reference/data	JSON. Use for further reference.

Action: List device nas

URL : http://192.168.0.209/api/device/{spesific device}/list/

Note:

{spesific device}. You must mention the device type. For example, nas.

You simply put http://192.168.0.209/api/device/nas/list/.

Required Arguments:

Parameter	Description
token	Kiwire API key.

Example Output:


```
{
  "result": [
    {
      "status": true,
      "message": "Successfully list all nas record",
      "data": [
        {
          "id": "75",
          "username": "testing",
          "nasname": "192.168.1.1",
          "shortname": "testing",
          "location": "Malaysia",
          "secret": "00:00:43:BC",
          "ports": "443",
          "description": "",
          "device_type": "mikrotik"
        }
      ]
    }
  ]
}
```

4. Radacct

Action: List all radacct

URL : http://192.168.0.209/api/radacct/list/

Required Arguments:

Parameter	Description
token	Kiwire API key.
start_time	Must be in format Y-m-d H:i:s
end_time	Must be in format Y-m-d H:i:s

Example Output:


```
{
  "result": [
    {
      "status": true,
      "message": "Successfully list all radacct records",
      "data": [
        {
          "acctuniqueid": "b8f17f8a6045b384",
          "username": "FB_10206192781657682",
          "nasipaddress": "192.168.0.74",
          "acctstarttime": "2017-02-14 09:44:08",
          "acctstoptime": "2017-02-14 09:44:16",
          "acctinputoctets": "7942",
          "acctoutputoctets": "90757",
          "calledstationid": "D4:CA:6D:76:5B:A4",
          "callingstationid": "F4:31:C3:AD:52:22",
          "acctterminatecause": "Idle-Timeout"
        }
      ]
    }
  ]
}
```


Output	Values
status	Boolean. If request is success, it will return as true. Otherwise false.
message	String. If the request is success, it will return the success message. Otherwise error message.
reference/data	JSON. Use for further reference.