

ZODIANET API (ZAPI2)



ZODIANET API “ZAPI 2”

Description of information exchange between the Zodianet platform and third-parties

Document : specification

Source: ZODIANET

Mail: support@zodianet.com

Revision:

1.0 : 01/04/2014 - publicly available.

1.1 : 18/04/2014 - 'ZiBASE 'URL Parameter has been switched with 'zibase'

1.2 : 19/04/2014 - 'zapi.php 'URL file has been switched with 'ZAPI.php'

Table of Contents

Introduction	3
What is Zodianet?	3
Zodianet actors	3
Zodianet API (ZAPI)	4
The Input of ZAPI2 : How to access functions?	5
1. HTTPS GET Request URL	5
2. HTTPS POST Request URL	5
3. Authentication parameters	5
4. Basic parameters.....	5
The Output of ZAPI2 functions	7
1. Response format	7
2. Error code	8
Using ZAPI2 functions in details.....	9
ZAPI2 targets for get service.....	9
1. Get ZIBASE id and token: Service=get->target=token	9
2. Get user data: Service=get->target=user.....	10
3. Get home data: Service=get->target=home	10
4. Get actuator data: Service=get->target=actuator.....	11
5. Get sensor data: Service=get->target=sensor.....	12
6. Get probe data: Service=get->target=probe->historic=day/month/year	12
7. Get scenario data: Service=get->target=scenario	14
8. Get remote data: Service=get->target=remote	14
9. Get camera data: Service=get->target=camera	15
10. Get variable data: Service=get->target=variable.....	16
ZAPI2 targets for execute service	17
1. Execute an actuator: service=execute->target=actuator	17
2. Execute a sensor: service=execute->target=sensor	18
3. Execute a remote: service=execute->target=remote.....	18
4. Execute a scenario: service=execute->target=scenario	19
5. Execute a camera: service=execute->target=camera	19

Introduction

This document provides a description of the second version of Zodianet API (ZAPI2) based on the REST architectural style.

What is Zodianet?

Zodianet is a company specialized in the development of home automation system, allowing the users to be connected permanently to their residence in order to manage the devices in his home using a PC or a mobile phone from anywhere in the world. We can classify the devices as follows: actuator, remote, sensor, probe, IP/TCP camera and scenario. A scenario is a virtual device which is defined as a series of actions according to user requirements.

Currently, Zodianet system offers the following functionalities:

- **Control sensors:** The user can control the detected events along with the time of detection, reported by sensors.
- **Consult probes:** The user can consult the probe measurements as well as the measured time (current measurements, historic measurements).
- **Take pictures/videos:** The user has the ability to view live security video stream.
- **Execute actuators/remotes/scenarios:** The user can execute an actuator/remote/scenario. For instance, on/off a light, open/close a window, adjust a heater, etc.
- **Manage devices:** The user can manage the connected devices (add/remove/modify).

Zodianet actors

Server side

1. **ZiBASE box** which collects the data from all smart objects of any house. ZiBASE is the heart of Zodianet system and operates as a home gateway to connect the home smart objects with the service platform of Zodianet via internet.
2. **Service platform** which collects the data from all ZiBASE boxes. Service platform operates as a server and communicates with all the user PCs and Mobile phones. It manages also the user accounts (login, sms credits, etc.).

Client side

Client applications allow users to set and display ZiBASE box(es) status. The setting application is run on PC whereas the displaying application is run on both PC (HTML¹) and mobile systems (Android², iPhone³, Windows phone⁴).

¹ http://ZiBASE.net/zodianet/zodiaWeb/html/ZiBASE_login.php

² <http://play.google.com/store/apps/details?id=com.zodianet>

³ <http://itunes.apple.com/fr/app/zodianet/id410311840?mt=8>

⁴ <http://www.windowsphone.com/en-us/store/app/zodianet/bc4506c1-8793-4be9-a221-762dce3e7867et>

Zodianet API (ZAPI)

The mentioned service platform performs also Zodianet API (ZAPI2) which is designed to connect the Zodianet platform with third-parties. In other words, this API gives the user the opportunity to access the functionalities of his ZiBASE system through a simple URL. For instance, the user can get the characteristics of all the devices attached to his ZiBASE box or he can execute an actuator, a scenario, a camera, etc.

Given this API, the user could also create its own applications to access its ZiBASE data.

Figure 1 presents an overview of the different actors of Zodianet system and ZAPI2.



Figure 1 Zodianet actors and ZAPI2

The Input of ZAPI2 : How to access functions?

ZAPI2 recommends to use HTTPS as the communication protocol rather than HTTP. Although requests to Zodianet Platform may be send through HTTP, HTTPS allows secured communication between you and ZAPI2.

You can access ZAPI2 functions on the web using the following HTTPS GET or POST requests.

1. HTTPS GET Request URL

<https://zibase.net/api/get/ZAPI.php>

2. HTTPS POST Request URL

<https://zibase.net/api/post/ZAPI.php>

3. Authentication parameters

In order to use ZAPI2, you need to have a ZiBASE account. If you didn't yet create your account, please go to https://zibase.net/zodianet/zodiaWeb/html/zibase_login.php and create a new account user.

The authentication for ZAPI2 is done with your ZiBASE id and your token. These two parameters are mandatory and should be featured for each request.

[zibase] is the identifier of your ZiBASE.

[token] is the secured password of your account.

The following URL allows you to get your ZiBASE id and your token in term of your login and your password:

[https://zibase.net/api/get/ZAPI.php?login=\[your login\]&password=\[your password\]&service=get&target=token](https://zibase.net/api/get/ZAPI.php?login=[your login]&password=[your password]&service=get&target=token).

4. Basic parameters

In addition of your ZiBASE id and your token, you need also to provide the following basic parameters:

[service] is the type of service you want to invoke from Zodianet platform. Actually, there are two types of service for HTTPS GET request: "get" and "execute". The "get" service gets the data from the featured target. The "execute" service executes an action on the featured target.

The services for HTTPS POST will be available in the near future.

[target] is the device on which you would like to perform a service. The target can be in the following types: user, home, actuators, sensors, probes, scenarios, remotes, cameras, variables.

An actuator is a type of device which activate/deactivate itself and knows its state.

A remote is a type of device that performs an action ON/OFF on an appliance and doesn't know in which state the appliance is.

A sensor is a type of device which informs a detected event along with the time of detection.

A probe is a type of device which measures a physical quantity (for instance: temperature, humidity). The measurements (current and historical) are stored in a variable. Some types of probe could measure two type of quantity (for instance: temperature and humidity). In this case, the first measurement is stored in a variable v1 and the second one is stored in another variable v2.

A scenario is a script of a series of actions, defining by a user.

A variable is an attribute which is defined by a user and could have different values. For instance: the user may desire to save the output of a device in a variable or a linear combination of some device's outputs.

The Output of ZAPI2 functions

1. Response format

Responses to your HTTPS request are in JSON format by default or in XML format if you specify. Each response is composed of a head field and a body field. The “head” field returns “success” in the case your request succeeds and “error” in the case your request fails.

The “body” field returns “data” according to the featured target if the request succeeds; otherwise it returns an error code.

Example for a JSON response

```
{
  "head": "success",
  "body": {
    "sensor": {
      "id": "xxx",
      "name": "xxx",
      "icon": "xxx"
    }
  }
}
```

```
{
  "head": "error",
  "body": 108
}
```

Example for a XML response

```
<head>success</head>
<body>
  <sensor>
    <id>xxx</id>
    <name>xxx</name>
    <icon>xxx</icon>
  </sensor>
</body>
```

```
<head>success</head>
<body>108</body>
```

2. Error code

Error code	Description
98	Authorization denied
99	Invalid authentication
100	Missing parameters
102	Missing ZiBASE
103	Missing Token
104	Missing Service
105	Missing Target
106	Missing Probe id
107	Missing Actuator id
108	Missing Sensor id
109	Missing Scenario id
110	Missing Camera id
111	Missing Remote id
112	Missing Data
113	Missing Period
114	Missing Action
115	Missing Variable id
202	Wrong ZiBASE
203	Wrong Token
204	Wrong Service
205	Wrong Target
206	Wrong Probe id
207	Wrong Actuator id
208	Wrong Sensor id
209	Wrong Scenario id
210	Wrong Camera id
211	Wrong Remote id
212	Wrong Data
213	Wrong Period
214	Wrong Action
215	Wrong Variable id

Using ZAPI2 functions in details

The aim of this section is to explain how to use ZAPI2 functions in details. As we explained before, user can request two types of services on one of his target for HTTPS GET method: get and execute.

The defined targets are explained hereafter:

ZAPI2 targets for get service

1. Get ZiBASE id and token
2. Get user data
3. Get home data
4. Get actuator data
5. Get sensor data
6. Get probe data
7. Get scenario data
8. Get remote data
9. Get camera data
10. Get variable data

1. Get ZiBASE id and token: Service=get->target=token

The token target returns user's ZiBASE id and token.

Request format

<https://zibase.net/api/get/zapi.php?login=xxx&password=xxx&service=get&target=token>

Request parameters description

Mandatory	Name	value	Description
Mandatory	login	string	User login
Mandatory	password	string	User password
Mandatory	service	get	Service type
Mandatory	target	token	ZiBASE token
Optional	format	JSON/XML	Response format. JSON by default

Response Example

```
{
  "head": "success"
  "body": {
    "zibase": "xxx",
    "token": "xxx",
  }
}
```

2. Get user data: Service=get->target=user

The user target returns general characteristics of your account.

Request format

<https://zibase.net/api/get/zapi.php?zibase=xxx&token=xxx&service=get&target=user>

Request parameters description

Mandatory	Name	value	Description
Mandatory	zibase	string	ZiBASE identifier
Mandatory	token	string	ZiBASE token
Mandatory	service	get	Service type
Mandatory	target	user	Target name
Optional	format	JSON/XML	Response format. JSON by default

Response Example

```
{
  "head": "success"
  "body": {
    "zibase": "xxx",
    "login": "xxx",
    "email": "xxx",
    "rights": "1",
    "lastConnection": "day/month/year"
  }
}
```

3. Get home data: Service=get->target=home

The home target returns a collection of all device data attached to your ZiBASE.

Request format

<https://zibase.net/api/get/ZAPI.php?zibase=xxx&token=xxx&service=get&target=home>

Request parameters description

Mandatory	Name	Value	Description
Mandatory	zibase	string	ZiBASE identifier
Mandatory	token	string	ZiBASE token
Mandatory	service	get	Service type
Optional	target	home	Target name
Optional	format	JSON/XML	Response format. JSON by default

Response Example

```
{
  "head": "success"
  "body": {
    "zibase": "xxx",
    "actuators": [
      {
        "id": "xxx",
        "name": "xxx",
        "icon": "xxx",
        "protocol": "xxx",
        "status": "xxx"
      },
      etc.
    ],
    "sensors": [
      {
        "id": "xxx",
        "name": "xxx",
        "icon": "xxx",
        "protocol": "xxx",
        "status": "xxx"
      },
      etc.
    ],
    etc.
  }
}
```

4. Get actuator data: Service=get->target=actuator

The actuator target returns the characteristics of a specific actuator attached to your ZiBASE.

Request format

<https://zibase.net/api/get/ZAPI.php?zibase=xxx&token=xxx&service=get&target=actuator&id=xxx>

Request parameters description

Mandatory	Name	Value	Description
Mandatory	zibase	string	ZiBASE identifier
Mandatory	token	string	ZiBASE token
Mandatory	service	get	Service type
Mandatory	target	actuator	Target name
Mandatory	id	string	Actuator id
Optional	format	JSON/XML	Response format. JSON by default

Response Example

```

{
  "head": "success"
  "body": {
    "actuator": {
      "id": "xxx",
      "name": "xxx",
      "icon": "xxx",
      "protocol": "xxx",
      "status": "xxx"
    }
  }
}

```

5. Get sensor data: Service=get->target=sensor

The sensor target returns the characteristics of a specific sensor attached to your ZiBASE.

Request format

<https://zibase.net/api/get/ZAPI.php?zibase=xxx&token=xxx&service=get&target=sensor&id=xxx>

Request parameters description

Mandatory	Name	Value	Description
Mandatory	zibase	string	ZiBASE identifier
Mandatory	token	string	ZiBASE token
Mandatory	service	get	Service type
Mandatory	target	sensor	Target name
Mandatory	id	string	Sensor id
Optional	format	JSON/XML	Response format. JSON by default

Response Example

```

{
  "head": "success"
  "body": {
    "sensor": {
      "id": "xxx",
      "name": "xxx",
      "icon": "xxx",
      "protocol": "xxx",
      "status": "xxx"
    }
  }
}

```

6. Get probe data: Service=get->target=probe->historic=day/month/year

Probe target returns the data, current or past, of a specified probe connected to your ZiBASE.

Request format

<https://zibase.net/api/get/ZAPI.php?zibase=xxx&token=xxx&service=get&target=probe&id=xxx&historic=day/month/year>

Request parameters description

Mandatory	Name	Value	Description
Mandatory	zibase	string	ZiBASE identifier
Mandatory	token	string	ZiBASE token
Mandatory	service	get	Service type
Mandatory	target	probe	Target name
Mandatory	id	string	Probe id
Optional	historic	day/month/year	Historic data
Optional	format	string	Response format. JSON by default

Response Example

```
{
  "head": "success"
  "body": {
    "id": "xxx",
    "zibase": "xxx",
    "data1Hours": [
      [time, value],
      [time, value],
      etc
    ],
    "data2Hours": [
      [time, value],
      [time, value],
      etc
    ],
    etc.
  }
}
```

7. Get scenario data: Service=get->target=scenario

Scenario target returns the characteristics of a specific scenario attached to your ZiBASE.

Request format

<https://zibase.net/api/get/ZAPI.php?zibase=xxx&token=xxx&service=get&target=scenario&id=xxx>

Request parameters description

Mandatory	Name	Type	Description
Mandatory	zibase	string	ZiBASE identifier
Mandatory	token	string	ZiBASE token
Mandatory	service	get	Service type
Mandatory	target	scenario	Target name
Mandatory	id	string	Scenario id
Optional	format	JSON/XML	Response format. JSON by default

Response Example

```
{
  "head": "success"
  "body": {
    "scenario": {
      "id": "xxx",
      "name": "xxx"
    }
  }
}
```

8. Get remote data: Service=get->target=remote

Remote target returns the characteristics of a specific remote attached to your ZiBASE.

Request format

<https://zibase.net/api/get/ZAPI.php?zibase=xxx&token=xxx&service=get&target=remote&id=xxx>

Request parameters description

Mandatory	Name	Type	Description
Mandatory	zibase	string	ZiBASE identifier
Mandatory	token	string	ZiBASE token
Mandatory	service	get	Service type
Mandatory	target	remote	Target name
Mandatory	id	string	Remote id
Optional	format	JSON/XML	Response format. JSON by default

Response Example

```
{
  "head": "success"
  "body": {
    "remote" : {
      "id" : "xxx",
      "name" : "xxx",
      "icon" : "xxx",
      "idON" : "xxx",
      "idOFF" : "xxx"
    }
  }
}
```

9. Get camera data: Service=get->target=camera

Camera target returns the characteristics of a specific camera attached to your ZiBASE.

Request format

<https://zibase.net/api/get/ZAPI.php?zibase=xxx&token=xxx&service=get&target=camera&id=xxx>

Request parameters description

Mandatory	Name	Type	Description
Mandatory	zibase	string	ZiBASE identifier
Mandatory	token	string	ZiBASE token
Mandatory	service	get	Service type
Mandatory	target	camera	Target name
Mandatory	id	string	Camera id
Optional	format	JSON/XML	Response format. JSON by default

Response Example

```
{
  "head": "success"
  "body": {
    "camera" : {
      "id" : "xxx",
      "name" : "xxx",
      "videoURL" : "https://",
      "commandUpURL" : "xxx",
      "commandDownURL" : "xxx",
      "commandLeftURL" : "xxx",
      "commandRightURL" : "xxx",
    }
  }
}
```

```

    "command1URL" : "xxx",
    "command2URL" : "xxx",
    "command3URL" : "xxx",
    "command4URL" : "xxx",
    "command5URL" : "xxx",
    "command6URL" : "xxx",
    "command7URL" : "xxx",
    "command8URL" : "xxx"
  }
}

```

10. Get variable data: Service=get->target=variable

The variable target returns the characteristics of a specific variable attached to your ZiBASE.

Note that variable ids begins at index 1.

Request format

<https://zibase.net/api/get/ZAPI.php?zibase=xxx&token=xxx&service=get&target=variable&id=xxx>

Request parameters description

Mandatory	Name	Type	Description
Mandatory	zibase	string	ZiBASE identifier
Mandatory	token	string	ZiBASE token
Mandatory	service	get	Service type
Mandatory	target	variable	Target name
Mandatory	id	string	Variable id
Optional	format	JSON/XML	Response format. JSON by default

Response Example

```

{
  "head": "success"
  "body": {
    "scenario" : {
      "id" : "xxx",
      "name" : "xxx"
    }
  }
}

```


ZAPI2 targets for execute service

1. Execute an actuator
2. Execute a sensor
3. Execute a remote
4. Execute a scenario
5. Execute a camera

1. Execute an actuator: service=execute->target=actuator

Activate/deactivate an actuator according to the featured action (1: activate, 0: deactivate)

Request format

<https://zibase.net/api/get/ZAPI.php?zibase=xxx&token=xxx&service=execute&target=actuator&id=xxx&action=xxx>

Request parameters description

Mandatory	Name	Value	Description
Mandatory	zibase	string	ZiBASE identifier
Mandatory	token	string	ZiBASE token
Mandatory	service	execute	Service type
Mandatory	target	actuator	Target name
Mandatory	id	string	actuator id
Mandatory	action	0/1	Action on an actuator: Activate(1) or deactivate(0)
Optional	format	JSON/XML	Response format. JSON by default

Response Example

```
{
  "head": "success"
}
```

2. Execute a sensor: service=execute->target=sensor

Read a detected event of a sensor.

Request format

<https://zibase.net/api/get/ZAPI.php?zibase=xxx&token=xxx&service=execute&target=sensor&id=xxx>

Request parameters description

Mandatory	Name	Value	Description
Mandatory	zibase	string	ZiBASE identifier
Mandatory	token	string	ZiBASE token
Mandatory	service	execute	Service type
Mandatory	target	sensor	Target name
Mandatory	id	string	Sensor id
Optional	format	JSON/XML	Response format. JSON by default

Response Example

```
{
  "head": "error",
  "body": "108"
}
```

3. Execute a remote: service=execute->target=remote

Execute a remote.

Request format

<https://zibase.net/api/get/ZAPI.php?zibase=xxx&token=xxx&service=execute&target=remote&id=xxx>

Request parameters description

Mandatory	Name	Value	Description
Mandatory	zibase	string	ZiBASE identifier
Mandatory	token	string	ZiBASE token
Mandatory	service	execute	Service type
Mandatory	target	remote	Target name
Mandatory	id	String	Remote id
Optional	format	JSON/XML	Response format. JSON by default

Response Example for the period=day

```
{
  "head": "success"
}
```

4. Execute a scenario: service=execute->target=scenario

Start a scenario script.

Request format

<https://zibase.net/api/get/ZAPI.php?zibase=xxx&token=xxx&service=execute&target=scenario&id=xxx>

Request parameters description

Mandatory	Name	Value	Description
Mandatory	zibase	string	ZiBASE identifier
Mandatory	token	string	ZiBASE token
Mandatory	service	execute	Service type
Mandatory	target	scenario	Target name
Mandatory	id	string	Scenario id
Optional	format	JSON/XML	Response format. JSON by default

Response Example

```
{
  "head": "success"
}
```

5. Execute a camera: service=execute->target=camera

Execute a camera according to the featured action. There are two types of actions:

- up/down/right/left to move the position of a camera
- Numbered command from 1 to 8. Each number presents a chained movements for the camera defined by the user

Request format

<https://zibase.net/api/get/ZAPI.php?zibase=xxx&token=xxx&service=execute&target=camera&id=xxx&action=xxx>

Request parameters description

Mandatory	Name	Value	Description
Mandatory	zibase	string	ZiBASE identifier
Mandatory	token	string	ZiBASE token
Mandatory	service	execute	Service type
Mandatory	target	actuator	Target name
Mandatory	id	string	actuator id
Mandatory	action	up/down/right/left	Action on a camera
Optional	format	JSON/XML	Response format. JSON by default

Response Example

```
{  
  "head": "success",  
}
```

END OF DOCUMENT