

WizarView API document

V4.4

2022-11-03

| Version | Date | Author | Description |
|----------------|-------------|---------------|-------------------------------------|
| 4.0 | 2017/12/19 | Lizhou | Add Function supported table. |
| 4.1 | 2019/09/25 | Lizhou | Add SoftSIM interfaces. |
| 4.2 | 2019/11/12 | Lizhou | Add query terminal state interface. |
| 4.3 | 2021-08-26 | Yzg | Update interfaces. |
| 4.4 | 2022-11-03 | Lizhou | Update interface url error |

| | |
|--|----|
| 1. Interfaces Introduction | 4 |
| 1.1. Http invocations | 4 |
| 1.1.1. Usage of token | 5 |
| 1.2. Basic Request Message format | 5 |
| 1.3. Basic response message format | 5 |
| 1.4. Request example | 5 |
| 2. Group interface | 9 |
| 2.1. Query country | 9 |
| 2.1.1. Request address | 9 |
| 2.1.2. Request message | 9 |
| 2.1.3. Response message | 9 |
| 2.2. Query states | 10 |
| 2.2.1. Request address | 10 |
| 2.2.2. Request message | 10 |
| 2.2.3. Response message | 10 |
| 2.3. Query group types | 11 |
| 2.3.1. Request address | 11 |
| 2.3.2. Request message | 11 |
| 2.3.3. Response message | 11 |
| 2.4. Query group | 12 |
| 2.4.1. Request address | 12 |
| 2.4.2. Request message | 12 |
| 2.4.3. Response message | 12 |
| 2.5. Add & Update group | 13 |
| 2.5.1. Request address | 13 |
| 2.5.2. Request message | 13 |
| 2.5.3. Response message | 14 |
| 2.6. Delete group | 15 |
| 2.6.1. Request address | 15 |
| 2.6.2. Request message | 15 |
| 2.6.3. Response message | 15 |
| 3. Terminal interface | 15 |
| 3.1. Query terminal | 15 |
| 3.1.1. Request address | 15 |
| 3.1.2. Request message | 16 |
| 3.1.3. Response message | 16 |
| 3.2. Query terminal state | 17 |
| 3.2.1. Request address | 17 |
| 3.2.2. Request message | 17 |
| 3.2.3. Response message | 18 |
| 3.3. Update terminal | 19 |
| 3.3.1. Request address | 19 |
| 3.3.2. Response message | 20 |
| 4. Application interface | 20 |

| | |
|---|----|
| 4.1. Query application | 20 |
| 4.1.1. Request address | 20 |
| 4.1.2. Request message | 20 |
| 4.1.3. Response message | 21 |
| 4.2. Add application | 22 |
| 4.2.1. Request address | 22 |
| 4.2.2. Request message | 22 |
| 4.2.3. Response message | 23 |
| 4.3. Delete application | 23 |
| 4.3.1. Request address | 23 |
| 4.3.2. Request message | 23 |
| 4.3.3. Response message | 23 |
| 5. Application version interface | 24 |
| 5.1. Query application version | 24 |
| 5.1.1. Request address | 24 |
| 5.1.2. Request message | 24 |
| 5.1.3. Response message | 25 |
| 5.2. Add application version | 26 |
| 5.2.1. Request address | 26 |
| 5.2.2. Request message | 26 |
| 5.2.3. Response message | 27 |
| 5.3. Delete application version | 27 |
| 5.3.1. Request address | 27 |
| 5.3.2. Request message | 27 |
| 5.3.3. Response message | 27 |
| 6. Terminal configure default application interface | 28 |
| 6.1. Add terminal default application | 28 |
| 6.1.1. Request address | 28 |
| 6.1.2. Request message | 28 |
| 6.1.3. Response message | 28 |
| 6.2. Delete terminal default application | 29 |
| 6.2.1. Request address | 29 |
| 6.2.2. Request message | 29 |
| 6.2.3. Response message | 29 |
| 7. Group configure default application interface | 30 |
| 7.1. Add group default application | 30 |
| 7.1.1. Request address | 30 |
| 7.1.2. Request message | 30 |
| 7.1.3. Response message | 30 |
| 7.2. Delete group default application | 31 |
| 7.2.1. Request address | 31 |
| 7.2.2. Request message | 31 |
| 7.2.3. Response message | 31 |
| 8. Audit log interface | 32 |

| | |
|---|----|
| 8.1. Query logs | 32 |
| 8.1.1. Request address | 32 |
| 8.1.2. Request message | 32 |
| 8.1.3. Response message | 32 |
| 9. Push message interface | 33 |
| 9.1. Request address | 33 |
| 9.2. Request message | 33 |
| 9.3. Response message | 34 |
| 10. Additional | 34 |
| 10.1. Status Code | 34 |
| 10.2. The functions supported table | 35 |

1. Interfaces Introduction

1.1. Http invocations

The developer gets the user id and user token from the wizarView administrator to access interface , and then each interface invoking , the data passed to wizarView will use the token as signature.

Developer should provide:

- Basic I.D. (name, company, email), used to mark new developer users in wizarView
- Required access rights, indicates the interface to access

Developer will get:

developer id and access token

1.1.1. Usage of token

Token is a security code that is negotiated in advance between wizarView and the third-party interfaces , which is used to participate in the hash calculation of the message to ensure the security of the communication message. Both the wizarView and the third-party interfaces should keep the token properly to avoid leaks.

Token using steps:

- Convert the body in the request message to a json string (a).
- Attach the token to the beginning of the JSON string (a) converted by the body, and attach Time-Token converted by the HTTP Header. namely, b = token + a + Time-Token.
- The new composition contains token strings is computed by sha256, and the result is encoded by base64, namely, hash = base64(sha256(b)).
- Add the hash to the request header and start the communication.

The example:

```
Hash = Base64( SHA256(Token + JSON string + time token) );
```

1.2. Basic Request Message format

| | |
|-------------|--|
| HTTP Header | Time-Token: 1629107790765 UID: 365 Sig: "The message hash" |
| HTTP Body | {"name":"group_01", "createTime":"1629107790765"} |

1.3. Basic response message format

The response message does not calculate the hash and no signature.

```
HTTP Body {"status": 1, "message":"success", "body": {....}}
```

1.4. Request example

```
public class Helper {  
  
    public static String base64(byte[] bs) {
```

```

String s = null;
if (bs != null) {
    s = org.apache.commons.codec.binary.Base64.encodeBase64String(bs);
}
return s;
}

public static String sha256(String str) {
    MessageDigest digest = null;
    try {
        digest = MessageDigest.getInstance("SHA-256");
        digest.update(str.getBytes("UTF-8"));
    } catch (Exception e) {
        return null;
    }
    return base64(digest.digest());
}

public static String calHash(String body, String token, long timeToken) {
    if (body == null) {
        throw new IllegalArgumentException("body can't be null");
    }
    String sha256String = token + body + timeToken;
    return sha256(sha256String);
}

public static String toJsonString(Object body) {
    return JSON.toJSONString(body, SerializerFeature.SortField);
}

public static ResponseMessage toResponseMessage(String responseJson) {
    JSONObject object = JSON.parseObject(responseJson);
    ResponseMessage message = new ResponseMessage(object.getString("status"));
    if (object.containsKey("message")) {
        message.setMessage(object.getString("message"));
    }
    if (object.containsKey("body")) {
        message.setBody(object.get("body"));
    }
    return message;
}

public static String post(String url, String uid, long timeToken, String sig, String text) {
    System.out.println("Request url: " + url);
    System.out.println("Request msg: " + text);
}

```

```

String result = null;
CloseableHttpClient httpclient = HttpClientBuilder.create().build();
try {
    HttpPost httpPost = new HttpPost(url);
    httpPost.setHeader("UID", uid);
    httpPost.setHeader("Time-Token", timeToken + "");
    httpPost.setHeader("Sig", sig);
    httpPost.setEntity(new StringEntity(text, "UTF-8"));

    HttpResponse response = httpclient.execute(httpPost);
    HttpEntity he = response.getEntity();

    result = EntityUtils.toString(he, "UTF-8");
} catch (Exception e) {
    e.printStackTrace();
} finally {
    try {
        httpclient.close();
    } catch (IOException e) {
    }
}
System.out.println("Response msg: " + result);
return result;
}

public static String post(String url, String uid, long timeToken, String sig, File appFile) {
    System.out.println("Request url: " + url);
    String result = null;
    CloseableHttpClient httpclient = HttpClientBuilder.create().build();
    try {
        HttpPost httpPost = new HttpPost(url);
        httpPost.setHeader("UID", uid);
        httpPost.setHeader("Time-Token", timeToken + "");
        httpPost.setHeader("Sig", sig);
        httpPost.setEntity(new FileEntity(appFile));

        HttpResponse response = httpclient.execute(httpPost);
        HttpEntity he = response.getEntity();

        result = EntityUtils.toString(he, "UTF-8");
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        try {

```

```

        httpclient.close();
    } catch (IOException e) {
    }
}

System.out.println("Response msg: " + result);
return result;
}

}

public class ThirdInterfaceTest {
    private String token = null;
    private long uid = 1;

    public ThirdInterfaceTest() {
        token = "5zrxndzy";
    }

    public static void main(String[] args) throws Exception {
        ThirdInterfaceTest test = new ThirdInterfaceTest();
        test.testIsOnline();
    }

    public void testIsOnline() throws Exception {
        String url = "https://www.wizarview.com/wizarView/api/third/terminal/list";
        Map<String, Object> body = new HashMap<>();
        body.put("sn", "WP15301Q00000014");

        String hash = Helper.calHash(body, token);
        String requestMsg = Helper.toJsonString(new RequestMessage(uid, body, hash));
        System.out.println("Request Message: " + requestMsg);

        String responseJson = HttpHelper.SendHttpRequest(url, requestMsg);
        System.out.println("Response Message: " + responseJson);

        ResponseMessage responseMessage = Helper.toResponseMessage(responseJson);
        System.out.println("status:" + responseMessage.getStatus());
    }
}

```

2. Group interface

2.1. Query country

2.1.1. Request address

<https://www.wizarviw.com/wizarView/api/third/country/list>

2.1.2. Request message

Example:

{}

2.1.3. Response message

Example:

```
{
    "status": "1",
    "message": "Success",
    "body": [
        {
            "id": 1,
            "name": "China",
        },
        {
            "id": 2,
            "name": "Amarica",
        },
        .....
    ]
}
```

Response message are in json format, meaning as follows:

| | |
|------|--------------|
| id | Country id |
| name | Country name |

2.2. Query states

2.2.1. Request address

```
https://www.wizarviw.com/wizarView/api/third/country/listStates
```

2.2.2. Request message

Example:

```
{}
```

2.2.3. Response message

Example:

```
{
    "status": "1",
    "message": "Success",
    "body": [
        {
            "id": 1,
            "name": "Shanghai",
            "countryId": "1",
        },
        {
            "id": 2,
            "name": "Shenzhen",
            "countryId": "1",
        },
        .....
    ]
}
```

Response message are in json format, meaning as follows:

| | |
|-----------|------------|
| id | State id |
| name | State name |
| countryId | Country Id |

2.3. Query group types

2.3.1. Request address

<https://www.wizarviw.com/wizarView/api/third/group/listTypes>

2.3.2. Request message

Example:

```
{}
```

2.3.3. Response message

Example:

```
{
    "status": "1",
    "message": "Success",
    "body": [
        {
            "id": 1,
            "name": "Manufacture",
        },
        {
            "id": 2,
            "name": "Generic",
        },
        .....
    ]
}
```

Response message are in json format, meaning as follows:

| | |
|------|---------------|
| id | Group type id |
| name | Group name |

2.4. Query group

2.4.1. Request address

<https://www.wizarviw.com/wizarView/api/third/group/list>

2.4.2. Request message

Example:

```
{  
    "GTL_id": "292",  
    "name": "wizarpes",  
    "phoneNo": "155387373",  
    "email": "152821857@qq.com"  
    "pageNo": 1,  
    "pageSize": 15,  
}
```

The message is in json format, meaning as follows:

| | |
|---------|---|
| GTL_id | Group id. If want to cache data, the value maybe cached the max id. |
| name | Group name |
| phoneNo | Telephone No. |
| email | Email address |

2.4.3. Response message

Example:

```
{  
    "status": "1",  
    "message": "Success",  
    "body": [  
        {"pageNo": 1,  
         "pageSize": 15,  
         "totalCount": 1,  
         "totalPage": 1,  
         "records": [{"  
             "id": 123,  
             "name": "wizarpes",  
             "parentGroupId": 23,  
             "parentGroupName": "QA",  
             "typeId": "4",  
         }]}  
    ]  
}
```

```

    "typeName": "General",
    "countryName": "China",
    "stateName": "Hainan",
    "address1": "wizarpos",
    "address2": "wizarpos",
    "zipCode": "200065",
    "phoneNo": "155387373",
    "faxNo": "",
    "description": "wizarpos",
    "email": "152821857@qq.com"
    "inheritDefaultApp": true,
  },]
}
}

```

Response message are in json format, meaning as follows:

| | |
|-------------------|-----------------------------------|
| id | Group id |
| name | Group name |
| parentGroupId | Parent group id |
| parentGroupName | Parent group name |
| typeId | Group type id |
| typeName | Group type name |
| countryName | Country name |
| stateName | State name/province name |
| address1 | Address1 |
| address2 | Address2 |
| phoneNo | Telephone No. |
| faxNo | Fax No. |
| email | Email address |
| inheritDefaultApp | Whether inherit group default App |

2.5. Add & Update group

2.5.1. Request address

<https://www.wizarviw.com/wizarView/api/third/group/save>

2.5.2. Request message

Example :

```
{
    "id": "31",
    "parentGroupId": 123,
    "name": "test",
    "typeId": "4",
    "email": "152821857@qq.com",
    "countryId": "8",
    "stateId": "20",
    "address1": "hongmei",
    "address2": "yangpu",
    "zipCode": "078234",
    "phoneNo": "155387373",
    "faxNo": "123567",
    "description": "this is a test",
    "inheritDefaultApp": true
},
```

The message is in json format, meaning as follows:

| | |
|--------------------------------|---|
| <code>id</code> | Group id. If add operation, this field is required. Otherwise is optional |
| <code>name</code> | Group name |
| <code>parentGroupId</code> | Parent group id |
| <code>typeId</code> | Group type id |
| <code>countryId</code> | Country id |
| <code>stateId</code> | State id/province id |
| <code>phoneNo</code> | Telephone No. |
| <code>email</code> | Email address |
| <code>address1</code> | Address1 |
| <code>address2</code> | Address2 |
| <code>zipCode</code> | Zip code |
| <code>faxNo</code> | Fax No. |
| <code>description</code> | Group description |
| <code>inheritDefaultApp</code> | Whether inherit group default App |

2.5.3. Response message

Example:

```
{
    "status": "1",
    "message": "success"
}
```

2.6. Delete group

2.6.1. Request address

<https://www.wizarviw.com/wizarView/api/third/group/delete>

2.6.2. Request message

Example:

```
{  
    "id": "58"  
}
```

The message is in json format, meaning as follows:

| | |
|----|----------|
| id | Group id |
|----|----------|

2.6.3. Response message

Example:

```
{  
    "status": "1",  
    "message": "success"  
}
```

3. Terminal interface

3.1. Query terminal

3.1.1. Request address

<https://www.wizarviw.com/wizarView/api/third/terminal/list>

3.1.2. Request message

Example:

```
{  
    "name": "yang",  
    "sn": "wp100022011",  
    "groupName": "huiyin",  
    "groupIds": "1,2,3,4,5",  
    "pageNo": 1,  
    "pageSize": 15,  
}
```

The message is in json format, meaning as follows:

| | |
|-----------|--|
| name | Terminal name |
| sn | Terminal SN |
| groupName | Group name |
| groupIds | Group id list, which are separate with ',' |

3.1.3. Response message

Example:

```
{  
    "status": "1",  
    "message": "Success",  
    "body": [  
        {"pageNo": 1,  
         "pageSize": 15,  
         "totalCount": 1,  
         "totalPage": 1,  
         "records": [{"  
             "id": 123,  
             "name": "yang",  
             "sn": "wp100022011",  
             "groupId": 234,  
             "groupName": "Demo",  
             "modelId": "12",  
             "modelName": "Q2",  
             "imeiNo1": "862634045001420",  
             "imeiNo2": "862634045001421",  
             "description": "baibaibaibai",  
         }]}]
```

```

        "registerTime": "2012-05-24 14:32:21",
        "online": "true",
        "networkType": "1",
        "gps": "31.244677 121.413956",
        "ip": "192.168.230.102"
    }]
}

```

Response message are in json format, meaning as follows:

| | |
|-------------------|-----------------------------------|
| Id | Terminal id |
| name | Terminal name |
| sn | Terminal sn |
| terminalModelName | Terminal model name |
| groupId | Group id |
| groupName | Group name |
| imeiNo1 | Terminal IMEI NO 1 |
| imeiNo2 | Terminal IMEI NO 2 |
| description | Terminal description |
| registerTime | Register time |
| online | Online state |
| networkType | Network type |
| gps | Terminal GPS (latitude/longitude) |
| ip | Terminal IP |

3.2. Query terminal state

3.2.1. Request address

<https://www.wizarviw.com/wizarView/api/third/terminal/state>

3.2.2. Request message

Example

```
{
}
```

```
        "sn": "WP13192000000047"  
    }
```

The parameter is json format, meaning as fellow:

| | |
|----|--------------------------|
| sn | Terminal sequence number |
|----|--------------------------|

3.2.3. Response message

Example

```
{  
    "status": "1",  
    "message": "Success",  
    "body": {  
        "id": "1234",  
        "sn": "WP13192000000047",  
        "customSn": "13192",  
        "groupId": "47",  
        "groupName": "KKGROUP",  
        "model": "W1",  
        "registerTime": "23412341213",  
        "systemVersion": "wp1.0.0-3496-g833be2c.rs3494",  
        "kernelVersion": "3.10.49-wp1.0.0-3557-g077c590  
builder@dailybuild #4  
Sun Apr 28 15:34:37 CST 2019  
engroot",  
        "splashVersion": "wizarpos",  
        "safeModuleVersion": "PCBB46",  
        "bootloaderVersion": "wp1.0.0-3556-gbd40ab6 pcbc",  
        "basebandVersion": "MPSS.JO.HY.V0.3-20190121-HYBRID",  
        "imei1": "869859021088447",  
        "imei2": "869859021088448",  
        "meid": "",  
        "batterySn": "WP13192000000047",  
        "bluetoothMac": "87:94:15:61:36:46",  
        "ethMac": "87:94:15:61:36:46",  
        "wifiMac": "87:94:15:61:36:46",  
        "wifiSsid": "KK-Wifi",  
        "status": "1",  
        "networkType": "1",  
        "ip": "192.168.200.21",  
        "gps": "31.24475 121.414254",  
        "appCount": "17",  
        "lastUpdateTime": "23452345",  
    }  
}
```

```
}
```

The parameters meaning:

| | |
|-------------------|----------------------------------|
| id | Terminal id |
| sn | Terminal sequence number |
| groupId | Group id |
| groupName | Group name |
| model | Terminal model |
| registerTime | Register time |
| systemVersion | System version |
| kernelVersion | Kernel version |
| splashVersion | Splash version |
| safeModuleVersion | Safe module version |
| bootloaderVersion | Bootloader version |
| basebandVersion | Baseband version |
| imei1 | Imei 1 |
| imei2 | Imei 2 |
| meid | Meid |
| batterySn | Battery sn |
| bluetoothMac | Bluetooth Mac |
| ethMac | Ethernet Mac |
| wifiMac | Wifi Mac |
| wifiSsid | Wifi name |
| status | Online state. 1 Online 0 Offline |
| networkType | Network type |
| ip | ip |
| gps | gps |
| appCount | Terminal application count |
| lastUpdateTime | Last update time |
| | |

3.3. Update terminal

3.3.1. Request address

```
https://www.wizarviw.com/wizarView/api/third/terminal/save
```

Example:

```
{  
    "id": 335,  
    "name": "yang",  
    "sn": "WP13192000000047",  
}
```

```
"groupId": 291,  
"inheritDefaultApp": false,  
"description": "wuwuw"  
}
```

The message is in json format, meaning as follows:

| | |
|-------------------|-----------------------------|
| id | terminal id to be updated |
| name | terminal name |
| sn | terminal sn |
| groupId | group id |
| description | terminal description |
| inheritDefaultApp | whether inherit default App |

3.3.2. Response message

Example:

```
{  
    "status": "1",  
    "message": "success"  
}
```

4. Application interface

4.1. Query application

4.1.1. Request address

```
https://www.wizarviw.com/wizarView/api/third/app/list
```

4.1.2. Request message

Example:

```
{  
    "GTL_id": "1",  
    "name": "cardinfo",  
    "groupName": "huiyin"
```

```

    "pageNo": 1,
    "pageSize": 15,
}

}

```

The message is in json format

| | |
|-----------|------------------------|
| name | application name |
| groupName | application group name |

4.1.3. Response message

Example:

```

{
    "status": "1",
    "message": "Success",
    "body": [
        {
            "pageNo": 1,
            "pageSize": 15,
            "totalCount": 1,
            "totalPage": 1,
            "records": [
                {
                    "id": 292,
                    "name": "cardinfo",
                    "type": "apk",
                    "packageName": "com.wizarpos.payment.xx",
                    "groupId": 123,
                    "groupName": "wizarpos",
                    "description": "",
                    "enabled": "1",
                    "restartPortal": "com.wizarpos.payment.MainActivity",
                    "snRegex": "",
                    "firmwareExpr": "",
                    "createTime": "2021-07-03 14:12:12",
                    "operator": "user",
                    "appVersionId": "5678",
                }
            ]
        }
    ]
}

```

The message is in json format, meaning as follows:

| | |
|------|------------------|
| name | Application name |
| id | Application id |
| type | Application type |

| | |
|---------------|---|
| packageName | Apk package name |
| groupId | Application group id |
| groupName | Application group name |
| description | Description |
| enabled | Application is enabled |
| restartPortal | Restart app portal class when upgrade app |
| snRegex | Terminal sn regex |
| firmwareExpr | Terminal firmware expression |
| createTime | Application create time |
| operator | The user of operate application |
| appVersionId | The application version which is using |

4.2. Add application

4.2.1. Request address

<https://www.wizarviw.com/wizarView/api/third/app/save>

4.2.2. Request message

Example:

```
{
  "id": "123",
  "name": "QA",
  "groupId": 213,
  "appTypeName": "apk",
  "description": "this is a test",
  "snRegex": "",
  "firmwareExpr": "",
  "restartPortal": "",
  "enabled": "1",
}
```

The message is in json format, meaning as follows:

| | |
|---------|--|
| id | Application id. Optinal. If create application, this field should be none. If update, must have value. |
| name | application name |
| groupId | application group id |

| | |
|---------------|--|
| appTypeName | application type name |
| description | description |
| snRegex | Terminal SN regex |
| firmwareExpr | Terminal firmware expression |
| restartPortal | The application on terminal restart portal - a class name with package name. |
| enabled | Enable 1 or 0 |

4.2.3. Response message

Example:

```
{
  "status": "1",
  "message": "success"
}
```

4.3. Delete application

4.3.1. Request address

<https://www.wizarviw.com/wizarView/api/third/app/delete>

4.3.2. Request message

Example:

```
{
  "id": 266
}
```

The message is in json format, meaning as follows:

| | |
|----|------------------------------|
| id | application id to be deleted |
|----|------------------------------|

4.3.3. Response message

example:

```
{  
    "status": "1",  
    "message": "Success"  
}
```

5. Application version interface

5.1. Query application version

5.1.1. Request address

<https://www.wizarviw.com/wizarView/api/third/app/version/list>

5.1.2. Request message

Example:

```
{  
    "name": "8.2.0",  
    "fileName": "alogcat_prod.apk",  
    "appName": "alogcat",  
    "groupName": "wizarpos",  
    "pageNo": 1,  
    "pageSize": 15,  
}
```

The message is in json format, meaning as follows:

| | |
|-----------|---|
| name | application version name |
| fileName | file name |
| appName | application name |
| groupName | Application group name , if set ,to find the terminal in the terminal group and sub-terminal group ,otherwise in the own terminal group and sub-terminal group by default |

5.1.3. Response message

Example:

```
{  
    "status": "1",  
    "message": "success",  
    "body": [  
        "pageNo": 1,  
        "pageSize": 15,  
        "totalCount": 1,  
        "totalPage": 1,  
        "records": [  
            {"id": 123,  
             "name": "8.2.0",  
             "groupId": 124,  
             "groupName": "huiyin",  
             "appId": 345,  
             "appName": "alogcat",  
             "versionCode": "20",  
             "fileName": "alogcat_prod.apk",  
             "fileSize": "162549",  
             "updateDate": "2016-03-21 10:10:10",  
         }]  
    ]  
}
```

The message is in json format, meaning as follows:

| | |
|------------|--------------------------|
| Id | application version id |
| name | application version name |
| groupId | group id |
| groupName | group name |
| fileName | file name |
| fileSize | file size |
| appName | application name |
| appId | application id |
| updateDate | update date |

5.2. Add application version

5.2.1. Request address

```
https://www.wizarviw.com/wizarView/api/third/app/version/save
```

5.2.2. Request message

Example:

```
File appFile = new File("/apk/run/PrinterService201508041348_plat-signed.apk");

UrlHelper helper = new UrlHelper(url);
String queryString = helper.addFirstParam("uid", uid)
    .addParam("appId", 961)
    .addParamEncodeUTF8("fileName", appFile.getName())
    .addParam("fileSize", appFile.length())
    .addParam("versionName", "1.0.1")
    .addParam("versionCode", 123)
    .addParam("paramType", "other")
    .addParam("fileSize", 5)
    .addParamEncodeUTF8("description", "upload test,test")
    .getAppendParam();

String hash = Helper.calHash(queryString, token);

url = helper.getUrl();
String responseJson = Helper.post(url, uid, timeToken, hash, appFile);
```

The file stream of the application version is written to body, and other properties are written after url. The following properties are shown as follows

| | |
|-------------|---|
| appId | application id |
| versionCode | application version code,Valid only when param is uploaded |
| versionName | Application version name,valid only when param is uploaded |
| fileName | file name |
| fileSize | file size |
| description | description |
| paramType | param type, optional value:other valid only when param is uploaded |

5.2.3. Response message

Example:

```
{  
    "status": "1",  
    "message": "success"  
}
```

5.3. Delete application version

5.3.1. Request address

```
https://www.wizarviw.com/wizarView/api/third/app/version/delete
```

5.3.2. Request message

Example:

```
{  
    "id":58  
}
```

The message is in json format, meaning as follows:

| | |
|----|--------------------------------------|
| id | application version id to be deleted |
|----|--------------------------------------|

5.3.3. Response message

Example:

```
{  
    "status": "1",  
    "message": "Success"  
}
```

6. Terminal configure default application interface

6.1. Add terminal default application

6.1.1. Request address

<https://www.wizarview.com/wizarView/api/third/app/terminal/add>

6.1.2. Request message

Example:

```
{  
    "uid": 182,  
    "body": {  
        "terminalId": 123,  
        "appId": 234,  
        "flag": 0  
    },  
    "hash": "pLZVBeHO2mYf3+bLhC1drHccsDTGxqEDpNTH2gwetBc="  
}
```

The message is in json format, meaning as follows:

| | |
|------------|----------------------------|
| terminalId | terminal id |
| appId | application id |
| flag | 0 Market 1 Prompt 2 Silent |

6.1.3. Response message

Example:

```
{  
    "status": "1",  
    "message": "success"  
}
```

6.2. Delete terminal default application

6.2.1. Request address

<https://www.wizarview.com/wizarView/api/third/app/terminal/delete>

6.2.2. Request message

Example:

```
{  
    "uid": 182,  
    "body": {  
        "terminalId": 123,  
        "appId": 234,  
  
    },  
    "hash": "pLZVBeHO2mYf3+bLhC1drHccsDTGxqEDpNTH2gwetBc="  
}
```

The message is in json format, meaning as follows:

| | |
|------------|----------------|
| terminalId | terminal Id |
| appId | application id |

6.2.3. Response message

Example:

```
{  
    "status": "1",  
    "message": "success"  
}
```

7. Group configure default application interface

7.1. Add group default application

7.1.1. Request address

<https://www.wizarview.com/wizarView/api/third/app/owner/add>

7.1.2. Request message

Example:

```
{  
    "uid": 182,  
    "body": {  
        "groupId": 123,  
        "appId": 2345,  
        "flag": 0,  
    },  
    "hash": "pLZVBeHO2mYf3+bLhC1drHccsDTGxqEDpNTH2gwetBc="  
}
```

The message is in json format, meaning as follows:

| | |
|---------|-------------------------------|
| groupId | group id |
| appId | application id |
| flag | 0 Market 1 Prompt 2 Silentres |

7.1.3. Response message

Example:

```
{  
    "status": "1",  
    "message": "success"  
}
```

7.2. Delete group default application

7.2.1. Request address

<https://www.wizarview.com/wizarView/api/third/app/owner/delete>

7.2.2. Request message

Example:

```
{  
    "GTL_id": "123",  
    "username": "lucy",  
    "target": "WP15461Q20002422",  
}
```

The message is in json format, meaning as follows:

| | | |
|----------|----------------|--|
| GTL_id | | |
| username | group id | |
| appId | application id | |

7.2.3. Response message

Example:

```
{  
    "status": "1",  
    "message": "Success",  
}
```

8. Audit log interface

8.1. Query logs

8.1.1. Request address

```
https://www.wizarview.com/wizarView/api/third/auditlog/list
```

8.1.2. Request message

Example:

```
{  
    "GTL_id": "123",  
    "username": "lucy",  
    "target": "WP15461Q20002422",  
}
```

The message is in json format, meaning as follows:

| | | | |
|----------|---|----------|--------|
| GTL_id | The log unique identifier. If contain this field, the output message will order by id desc. Otherwise asc | Optional | Long |
| username | Operator name | Optional | String |
| target | Operate object | Optional | String |

8.1.3. Response message

Example:

```
{  
    "status": "1",  
    "message": "Success",  
    "body":{  
        "pageNo":1,  
        "pageSize": 15,  
        "totalCount": 1,  
        "totalPages": 1,  
        "records": [  
    ]  
}
```

```
{
    "action": "auditLog.action.saveApplication",
    "actionType": 2,
    "id": 1177,
    "ip": "127.0.0.1",
    "logTime": "2021-09-08 14:46:36",
    "operator": "admin",
    "groupId": 1,
    "groupName": "Root",
    "targetObjectName": "home-xxx"
},
]
}
}
```

Response message are in json format, meaning as follows:

| | |
|------------------|----------------------|
| id | Log id |
| action | Action label |
| actionType | Action type id |
| targetObjectName | The target of action |
| groupId | User group id |
| groupName | User group name |
| operator | User |
| logTime | Action time |

9. Push message interface

The third-party customers can push message to terminal by wizarView.

9.1. Request address

<https://www.wizarview.com/wizarView/api/third/push>

9.2. Request message

Example :

```
{
    "uid": 182,
    "body": {
```

```

    "sn": "WP1234567890",
    "packageName": "com.wizarpos.xxxx",
    "className": "com.wizarpos.xxxx.MainActivity",
    "msg": "push message",
},
"hash": "pLZVBeHO2mYf3+bLhC1drHccsDTGxqEDpNTH2gwetBc="
}

```

9.3. Response message

Example:

```

{
  "status": "1",
  "message": "success"
}

```

10. Additional

10.1. Status Code

Status code 1 is success. Others are failures.

| Status code | description |
|-------------|--|
| 1 | success |
| 2 | No request message |
| 3 | Invalid user |
| 4 | No the third-party interface configuration |
| 5 | IP not allowed |
| 6 | Invalid message, HASH validation is not passed |
| 7 | No permission to operate |
| 8 | No permission to operate group |
| 9 | Non-existent group parent |
| 10 | Non-existent group |
| 11 | Non-existent group type |
| 12 | Group name has existed |
| 13 | Non-existent country name |
| 14 | Non-existent state name |
| 15 | Non-existent terminal SN |

| | |
|-----|--|
| 16 | No permission to operate terminal |
| 17 | Non-existent terminal model |
| 18 | Sn that doesn't fit the rules |
| 19 | Non-existent application type |
| 20 | No permission to operate application |
| 21 | Non-existent application id |
| 22 | No permission to operate application version |
| 23 | Use http mode ,but no certificate |
| 24 | Incorrect X500Name |
| 25 | Issuer certificate not found |
| 26 | Certificate validation is not passed |
| 27 | Not supported application type, APK PARAM is currently supported only. |
| 28 | Verify application failed |
| 29 | Id and sn is not match |
| 30 | No configure provider |
| 31 | Terminal not in group[NEW], can not be added |
| 32 | Terminal not in group[NEW], can not be added |
| 33 | Non-existent parameter file |
| 34 | Non-existent ID |
| 35 | Terminal and application parameter is not match |
| 36 | Terminal doesn't have permission to configure this parameter template |
| 37 | Non-existent terminal ID |
| 38 | Non-existent group ID |
| | |
| 500 | Internal server error |
| | |
| | |

10.2. The functions supported table

| Function Name | | Supported |
|----------------------|--------------------|-----------|
| Application | Query | Yes |
| | Add | Yes |
| | View | Yes |
| | Edit | Yes |
| | Delete | Yes |
| | Version Configured | Yes |
| Application Provider | Query | No |
| | Add | No |

| | | |
|---------------------|------------------------|-----|
| | View | No |
| | Edit | No |
| | Delete | No |
| Application Version | Query | Yes |
| | Add | Yes |
| | Edit | Yes |
| | Delete | Yes |
| Terminal | Query | Yes |
| | Add | Yes |
| | View | Yes |
| | Edit | Yes |
| | Delete | No |
| | Default Application | Yes |
| | ApplicationParamConfig | No |
| | TerminalAttribute | No |
| | Certificate | No |
| | Push Apps | No |
| | Change Group | No |
| Group | Query | Yes |
| | Add | Yes |
| | Edit | Yes |
| | View | Yes |
| | Delete | Yes |
| | ApplicationParamConfig | No |
| | Default Application | Yes |
| Model | Query | No |
| | Add | No |
| | Edit | No |
| | View | No |
| | Delete | No |
| Manufacturer | Query | No |
| | Add | No |
| | Edit | No |
| | View | No |
| | Delete | No |
| Register Rules | Query | No |
| | Add | No |
| | Edit | No |
| | View | No |
| | Delete | No |
| Batch Import | Query | No |
| | View Details | No |
| | File Import | No |

| | | |
|-----------------------|--------------------|-----|
| Terminal Map | Query | No |
| Terminal Tag | Query | No |
| | Add | No |
| | Edit | No |
| | View | No |
| | Delete | No |
| | Config | No |
| Cert Change Log | Query | No |
| Parameter | Query | No |
| | Add | No |
| | Edit | No |
| | View | No |
| | Delete | No |
| Parameter Template | Query | No |
| | Add | No |
| | Edit | No |
| | View | No |
| | Delete | No |
| | ConfigDefaultParam | No |
| Download Monitor | Query | No |
| Download History | Query | No |
| Terminal Monitor | Query | No |
| | Latest Apps | No |
| | Latest Events | No |
| | Upload Log | No |
| | Change Password | No |
| | Push Apps | No |
| Terminal Install Apps | Query | No |
| Users | Query | No |
| | Add | No |
| | Edit | No |
| | View | No |
| | Delete | No |
| | Usb Token | No |
| Roles | Query | No |
| | Add | No |
| | Edit | No |
| | View | No |
| | Delete | No |
| | Permission | No |
| Audit Log | Query | Yes |
| | | |

