API Documentation
For WebSocket Structure Devices

Communication Protocol v2.1.4

20181101

Index

1. WebSocket Protocol Description

WebSocket protocol was standardized as RFC 6455 by IEIF in Year 2011. WebSocket was firstly standardized in HTML5 as TCP connection. WebSocket is a standardized protocol, enables real-time full-duplex communication between Server and Client. WebSocket has become a protocol providing cross-platform real-time communication between Server and Client.

Biggest difference between WebSocket and traditional HTTP:

- WebSocket is a two-way communication protocol, after establishing the connection, WebSocket Server and Browser (Client Agent) can actively send or receive data from each other, like a Socket.
- WebSocket requires TCP-like Client and Server to connect through handshake, only after connecting successfully can they achieve the intercommunication.

For the Time Attendance Terminal we mentioned in this manual, it uses the WebSocket for the communication. Fully using the Web Socket’s fast and low resource occupancy, it’s a better choice for the Time Attendance Terminal and Access Control Terminal application scene.

NOTE: we could refer to https://www.websocket.org for more details for WebSocket protocol.

2. DEMO Running Environment

Because the WebSocket comes along with the Windows System only support Win8 or above, or Server 2012 or above, there would be some limitation for the User to get familiar with the product and the development. Therefore, We provide an unlimited Windows version as winform Server, click to run the WebSocket API of the test device.

NOTE: This demo uses a third party component supersocket, need to run in the computer with .NET Framework 4.0 or above.

Because the demo is mainly to test the Terminal API, for the parts that we don't release to the web client, if you need to show on every computer's webpage, please develop your own front end code.

3. API Function List for different Hardware

Judge the <TerminalType> in Register commands, √ Support。 × Not Support。

TerminalType: if no suffix, means support both FACE and FP, -FP supprot Fingerprint only, -FACE support Face only.

	Hardware Platform	4900	4900	2960	4960	5900	**5928**		
No	**API item** **TerminalType**	PFS100	PFS100-FP	YY100-FP	YY100-FACE	JF100	**ZC100-FACE**		**More...**
1	**Register**	√	√	√	√	√	√		
2	**Login**	√	√	√	√	√	√		
3	**GetAllUserID**	×	×	×	×	×	√		
4	**GetUserAllInfo**	√	×	×	√	√	√		
5	**GetUserSampleInfo**	√	×	×	√	√	√		
6	**SetUserSampleInfo**	×	×	×	×	√	√		
7	**GetUserData**	√	√	√	√	√	√		
8	**SetUserData**	√	√	√	√	√	√		
9	**GetFirstUserData**	×	×	×	×	√	√		
10	**GetNextUserData**	×	×	×	×	√	√		
11	**GetUserPassword**	√	√	√	√	√	√		
12	**GetUserCardNo**	√	√	√	√	√	√		

13	**GetUserPhoto**	×	×	×	×	×	√		
14	**SetUserPhoto**	×	×	×	×	×	√		
15	**GetFingerData**	√	√	√	×	√	×		
16	**SetFingerData**	√	√	√	×	√	×		
17	**GetFaceData**	√	×	×	√	√	√		
18	**SetFaceData**	√	×	×	√	√	√		
19	**RemoteEnroll**	×	×	×	×	√	√		
20	**ExitRemoteEnroll**	×	×	×	×	√	√		
21	**TakeOffManager**	×	×	×	×	√	√		
22	**EnableDevice**	×	×	×	×	√	√		
23	**GetTime**	√	√	√	√	√	√		
24	**SetTime**	√	√	√	√	√	√		
25	**GetDepartment**	√	√	√	√	√	√		
26	**SetDepartment**	√	√	√	√	√	√		
27	**GetProxyName**	×	×	×	×	×	√		
28	**SetProxyName**	×	×	×	×	×	√		
29	**GetBellTime**	×	×	×	×	√	√		
30	**SetBellTime**	×	×	×	×	√	√		
31	**GetDeviceSetting**	×	×	×	×	√	√		
32	**SetDeviceSetting**	×	×	×	×	√	√		

33	**RestoreDevice**	×	×	×	×	√	√		
34	**GetPowerSetting**	×	×	×	×	√	√		
35	**SetPowerSetting**	×	×	×	×	√	√		
36	**FirmwareUpgradeByCloud**	√	×	×	√	√	√		
37	**GetEthernetSetting**	×	×	×	×	√	√		
38	**SetEthernetSetting**	×	×	×	×	√	√		
39	**GetWifiSetting**	×	×	×	×	√	√		
40	**SetWifiSetting**	×	×	×	×	√	√		
41	**GetMobileNetSetting**	×	×	×	×	√	×		
42	**SetMobileNetSetting**	×	×	×	×	√	×		
43	**GetVPNServer**	×	×	×	×	√	√		
44	**SetVPNServer**	×	×	×	×	√	√		
45	**GetGPS**	×	×	×	×	√	×		
46	**SetGPS**	×	×	×	×	√	×		
47	**GetCloudServer**	×	×	×	×	√	√		
48	**SetCloudServer**	×	×	×	×	√	√		
49	**GetStreamingServer**	×	×	×	×	×	√		
50	**SetStreamingServer**	×	×	×	×	×	√		
51	**GetLocalServer**	×	×	×	×	×	√		
52	**SetLocalServer**	×	×	×	×	×	√		

53	**EmptyTimeLog**	√		√		√		√		√		√			
54	**EmptyManageLog**	√		√		√		√		√		√			
55	**EmptyUserEnrollmentData**	√		√		√		√		√		√			
56	**EmptyAllData**	√		√		√		√		√		√			
57	**TimeLog**	√		√		√		√		√		√			
58	**AdminLog**	√		√		√		√		√		√			
59	**GetAttendanceLog**	√		×		√		√		√		√			
60	**GetLogSetting**	×		×		×		×		√		√			
61	**SetLogSetting**	×		×		×		×		√		√			
62	**GetAttendanceRule**	×		×		×		×		×		×			
63	**SetAttendanceRule**	×		×		×		×		×		×			
64	**AccessStatus**	×		×		×		×		√		√			
65	**GetAccessSetting**	×		×		×		×		√		√			
66	**SetAccessSetting**	×		×		×		×		√		√			
67	**GetAccessList**	×		×		×		×		√		√			
68	**SetAccessList**	×		×		×		×		√		√			
69	**GetStoreStatus**	×		×		×		×		√		√			
70	**GetDeviceInfo**	×		×		×		×		√		√			

4. Handshake Process

【Device】and【Server】Connection Steps:

1、【Device】MENU – Network. – Cloud Server (Different Device, it might be different in the MENU), This URL is pointing to the 【Server】Address. Format: ws://IP:port/path , for example：ws://192.168.1.199:443/wssample。If the Server SSL encryption, then need use wss://

2、【Device】sends the WebSocket handshake request to the pointing【Server】.

3、【Server】receives the request from【Device】, and then judge the WebSocket handshake (Please refer to the official website of WebSocket for more details https://www.websocket.org).

4、【Device】and【Server】WebSocket handshake connection is established successfully. And now, we have established an independent TCP channel for the【Device】and【Server】.【Device】and【Server】can send message to each other.

5、【Device】send <Register> command to【Server】. (command details refer to the following **Specific API**) , register the【Device】with the correct serial number to the 【server】.

6、【Server】receives <Register> command, and save the serial number and related parameters of the Time Attendance Device, then returns a token value (this value is the sessionID of the WebSocket). And [cloudID] parameter is a unique string that agreed by the Device Supplier and the Server Supplier, Server could use this string to judge if this Device belongs to them(Judge the machines was Sold by them).

7、【Device】receive the token value, and send a <Login> command to request login. After that, each time after connecting to the network,

the device just need to send the <Login> request.

8、【Server】judges the receiving <Login> command parameter, and judge if the Device is allowed to login. If allowed, then returns the corresponding parameter.

9、Login successfully, 【Device】and【Server】connect successfully in API. Both of them can operate the "Upload User Info", "Attendance Record" and "Access control" etc., all other operations.

10、If want to disconnect, can disconnect WebSocket connection directly.

5. Specific API

1) Register: Register Request

Device Request	
<?xml version="1.0"?> *<Message>* *<Request>Register</Request>* *<Rrid>0</Rrid>* *<Version>ZD4900 v2.0.180308</Version>* *<TerminalType>PFS100-FP</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<ProductName>M70</ProductName>* *<DeviceSerialNo>sn123</DeviceSerialNo>* *<CloudId>cloudid12345678</CloudId>* *</Message>*	<Request>:Register, request to register, after the first time the new device connect the Server, it sends Register request to the Server every 10 seconds, until the Serve returns a response. <Rrid>: Session sequence number, long type, increment by increase 1. <Version>: Firmware Version <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <ProductName>: Device Product Name <DeviceSerialNo>: Device unique serial number <CloudId>: Cloud ID, an identifier of the Device's default connecting Cloud Server. Cloud Server could use it to judge if allow the Device to connect to the Cloud Server or not.
Server Response	
<?xml version="1.0"?> *<Message>* *<Response>Register</Response>* *<Actid>0</Actid>* *<Time>2018-05-09-T09:42:45Z</Time>* *<DeviceSerialNo>wb2018042802</DeviceSerialNo>* *<Token>e0677693-47f7-45ed-8276-bd9f5759b4fe</Token>* *<Result>OK</Result>* *</Message>*	<Response>: Register, respond the register request <Actid>: Actid is Rrid <Time>: Server Time, Device receive this Time, Synchronize to the Device. <DeviceSerialNo>: Device unique Serial Number <Token>: Cloud Server returns a session ID, we can use the WebSocket's sessionID. <Result>: result. OK/Fail.

2) Login: Login Request

Device Request	
<?xml version="1.0"?> *<Message>* *<Request>Login</Request>* *<Rrid>1</Rrid>* *<Version>ZD4900 v2.0.180308</Version>* *<DeviceSerialNo>wb2018042802</DeviceSerialNo>* *<Token>e0677693-47f7-45ed-8276-bd9f5759b4fe</Token>* *</Message>*	<Request>: *Login, Login request, after the Device has been registered in the Server successfully, it sends login request to the server every 30 seconds, until the server return a response. Device re-connect to the network, it will not send register request, it will send login request directly.* <Rrid>: Session sequence number, long type, increment by increase 1. <Version>: Firmware Version. <DeviceSerialNo>: Device unique serial number. *<Token>: it's the token value which is saved during Register.*
Server Response	
<?xml version="1.0"?> <Message> <Response>Login</Response> <Actid>1</Actid> <Time>2018-05-09-T09:42:53Z</Time> <DeviceSerialNo>wb2018042802</DeviceSerialNo> <Result>OK</Result> </Message>	<Response>: Login, respond the Login Request <Actid>: Actid is Rrid <Time>: Server Time, Device receive this Time, Synchronize to the Device. <DeviceSerialNo>: Device unique serial number. <Result>>: result. OK/Fail.

3) GetAllUserID: Get All User ID

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>GetAllUserID</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show</Action_ext>* *</Message>*	<Request>:*GetAllUserID, get all User ID* <*Ccid*>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. *<Action_ext>: action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.*
Device Response	
<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>* *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>* *<Action_ext>show/xx</Action_ext>* *<Response>GetAllUserID</Response>* *<UserID>1,2,4,5</UserID>* *<Result>OK</Result>* *</Message>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <UserID>: User ID, int. All User ID in the Device. Use “,” to separate. <Result>: OK/Fail, result

4) GetUserAllInfo: Get All Enrollment Information of one User ID

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>GetUserAllInfo</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show</Action_ext>* *<UserID>1</UserID>* *</Message>*	<Request>:*GetUserAllInfo, get all information of one User ID.* <*Ccid*>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. *<Action_ext>>: action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* *<UserID>：User ID, int.*
Device Response	
<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>* *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>* *<Action_ext>show/xx</Action_ext>* *<Response>GetUserAllInfo</Response>* *<UserID>1</UserID>* *<Name>Peter</Name>* *<Privilege>User</Privilege>* *<Depart>0</Depart>* *<Enabled>Yes</Enabled>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <UserID>: *User ID, int.* <Name></Name>: Username. <Privilege>: User, Register, Manager, Administrator. User level. <Depart>: 0``19. The user's department <Enabled>: Yes/No. This User can use or not? <Card>: Card number,, 0 or NULL means no enrollment.

<Card>0</Card> *<PWD>0</PWD>* *<FingersData Fingers="10">* *<Finger id="1">Data</Finger>* *<Finger id="2">Data</Finger>* *<Finger id="10">Data</Finger>* *</FingersData>* *<FaceData>Data</FaceData>* *<Result>OK</Result>* *</Message>*	<PWD>: Password, 0 or NULL means no enrollment. <FingersData Fingers="10">: Fingers value is the total enrollment Fingerprint for the current User. Max is 10 FP. <Finger id="1">: The first Fingerprint data *<!—4900 Platform，996 Byte -->* <FaceData>: Face data *<!-- 4900Platform，4000Byte -->* <Result>:OK/Fail, result.

5) GetUserSampleInfo: Get brief details for some specified User ID

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>GetUserSampleInfo</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx</Action_ext>* *<UserCounts>6</UserCounts>* *<UserID>1,6,13,53-55</UserID>* *</Message>*	<Request>: GetUserSampleInfo, get brief details for some specified User ID. <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>: *action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* <UserCounts>: count of the requested users <UserID>: *User ID, int. Use English Coma "," to separate the User ID. If the User ID is in consecutive, use "-" for connection. There might be no enrollment for these User ID, there might be no return data.*

Device Response	
<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>* *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>* *<Action_ext>show/xx</Action_ext>* *<Response>GetUserSampleInfo</Response>* *<UserSampleInfo UserCounts="6">* *<Info UserID="1" Name="Peter" Privilege="User" Depart="0" Card="9499024" PWD="1" Fingers="2" Face="Yes"/>* *<Info UserID="2" Name="" Privilege="User" Depart="0" Card="0" PWD="0" Fingers="2" Face="No"/>* *</UserSampleInfo>* *<Result>OK</Result>* *</Message>*	<Actid>: <Actid> is<Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <UserSampleInfo UserCounts="2"> : UserCounts value is the true return count, the requested number might be bigger than this value, but the request User ID might not be enroll in the Device yet, so no need to return this kind of User. UserID="1": User ID Name="Peter": User Name Privilege="User": User Level Depart="0": 0~~19, Department Card="9499024": Card Number PWD="1" : Password Fingers="2" : enrollment fingerprint count. Face="Yes": Yes/No, have already enroll the Face or not <Result>: OK/Fail, result

6) SetUserSampleInfo: Set brief info for some specified User ID

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>SetUserSampleInfo</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx</Action_ext>* *<UserSampleInfo UserCounts="2">* *<Info UserID="1" Name="Peter" Privilege="User" Depart="0" Card="9499024" PWD="1" Fingers="2" Face="Yes"/>* *<Info UserID="6" Name="" Privilege="User" Depart="0" Card="0" PWD="0" Fingers="2" Face="No"/>* *</UserSampleInfo>* *</Message>*	<Request>: *SetUserSampleInfo, set brief info for some specified User I.* <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>: *action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* <UserSampleInfo UserCounts="2"> : UserCounts is the total amount of the requested User ID *<Info >same as above.*
Device Response	
<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>* *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>* *<Action_ext>show/xx</Action_ext>* *<Response>SetUserSampleInfo</Response>*	<Actid>:<Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. *<SetUserID UserCounts="2">：the total amount and User ID that have been set*

<SetUserID UserCounts="2">1,6</SetUserID> *<Result>OK</Result>* *</Message>*	*successfully* <Result>: OK/Fail, result

7) GetUserData: Get User Information of one specified User ID (Not including fingerprint and face)

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>GetUserData</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx</Action_ext>* *<UserID>1</UserID>* *</Message>*	<Request>: *GetUserData, get user information of the specified User ID.* <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>: *action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* <UserID>: *User ID, int.*
Device Response	
<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>* *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>* *<Action_ext>show/xx</Action_ext>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>:Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly.

<Response>GetUserData</Response> *<UserID>1</UserID>* *<UserSN>455dfd54SDF</UserSN>* *<Name></Name>* *<Privilege>User</Privilege>* *<Depart>0</Depart>* *<Enabled>Yes</Enabled>* *<Card>9499024</Card>* *<PWD>1</PWD>* *<Fingers>2</Fingers>* *<FaceEnrolled>Yes</FaceEnrolled>* *<Result>OK</Result>* *</Message>*	<UserID>: *User ID, int.* <UserSN>: User sequence number, similar to User ID, string type. <Name></Name>: User name <Privilege>: User, Register, Manager, Administrator. User Level for this User. <Depart>: 0``19. The user's department. <Enabled>: Yes/No. this user can use or not. <Card>: Card number, 0 or NULL means no enrollment. <PWD>: Password, 0 or NULL means no enrollment. *<Fingers>:* total enrollment Fingerprint for the current User. Max is 10 FP. *<FaceEnrolled>*： Yes/No, have already enroll the Face or not <Result>: OK/Fail, result

8) SetUserData: Set User Information of one specified User ID (Not including fingerprint and face)

Server Request	

<?xml version="1.0"?> *<Message>* *<Request>SetUserData</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *<UserID>1</UserID>* *<Type>Set</Type>* *<Name>/23..</Name>* *<UserSN>45F</UserSN>* *<Privilege>User</Privilege>* *<Depart>0</Depart>* *<Enabled>Yes</Enabled>* *<StartDateTime>2000-01-01-T00:00M</StartDateTime>* *<EndDateTime>2020-12-31-T23:59M</EndDateTime>* *<Card>543453</Card>* *<PWD>012345</PWD>* *</Message>*	<Request>: *SetUserData, set user information of one specified User ID.* <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* <UserID>: *User ID, int.* *<Type>: Set/Delete/Edit*，Set, need with full fields, otherwise it will clear up the info(not including the Fingerprint, Face and Photo), if with NULL value, it will clear up too; Edit, can set some specified fileds, others without , then keep the same value, will not change; Delete, delete the current User. <Name></Name>: user name <UserSN>: User sequence number, similar to User ID, string type. *<Privilege>:*User, Register, Manager, Administrator. User Level for this User *<Depart>*:0``19. The user's department <Enabled>: Yes/No. this user can use or not <Card>:Card number, 0 or NULL means no enrollment <PWD>: Password, 0 or NULL means no enrollment
Device Response	
<?xml version="1.0"?> <Message> <Actid>2</Actid> <TerminalType>PFS100</TerminalType> <HardwareVer>4900</HardwareVer> <TerminalID>1</TerminalID> <DeviceSerialNo>wb2018042801</DeviceSerialNo> <Version>ZD4900 v2.0.180308</Version> <Action_ext>show/xx</Action_ext>	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly.

<Response>SetUserData</Response> <UserID>1</UserID> <UserSN>*45F*</UserSN> <Type>Set</Type> <Result>OK</Result> </Message>	<UserID>: *User ID, int.* <UserSN>: User sequence number, similar to User ID, string type. <Type>: Set/Delete/Edit, same as value as the request command. <Result>: OK/No UserID/Card Duplicate/Fail, result. When delete, if the User is not exist, it returns No UserID; edit or se, the Card Number has been already used by some other User ID, it returns Card Duplicate.

9) GetFirstUserData: get the information of the first User ID

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>GetFirstUserData</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx</Action_ext>* *</Message>*	<Request>: *GetFirstUserData, get the information of the first User ID.* <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.*
Device Response	
<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>* *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>* *<Action_ext>show/xx</Action_ext>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly.

<Response>GetUserData</Response> *<UserID>1</UserID>* *<UserSN>455dfd54SDF</UserSN>* *<Name></Name>* *<Privilege>User</Privilege>* *<Depart>0</Depart>* *<Enabled>Yes</Enabled>* *<StartDateTime>2000-01-01-T00:00M</StartDateTime>* *<EndDateTime>2020-12-31-T23:59M</EndDateTime>* *<Card>9499024</Card>* *<PWD>1</PWD>* *<Fingers>2</Fingers>* *<FaceEnrolled>Yes</FaceEnrolled>* *<More>Yes/No</More>* *<Result>OK</Result>* *</Message>*	<UserID>: *User ID, int.* <UserSN>: User sequence number, similar to User ID, string type. <Name></Name>: User name <Privilege>: User, Register, Manager, Administrator. The user's level. <Depart>:0``19. The user's department <Enabled>: Yes/No. this user can use or not <Card>:Card number, 0 or NULL means no enrollment <PWD>: Password, 0 or NULL means no enrollment *<Fingers>:* total enrollment Fingerprint for the current User. Max is 10 FP. *<FaceEnrolled>*：Yes/No, have already enroll the Face or not *<More>：Yes/No，Yes means there's a next User in the Device. Server should send* GetNextUserData request. <Result>: OK/Fail, result

10) GetNextUserData: Get the information of the next User ID

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>*GetNextUserData*</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx</Action_ext>* *</Message>*	<Request>:GetNextUserData, get the information of the next User ID. <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.*
Device Response	
<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>* *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>* *<Action_ext>show/xx</Action_ext>* *<Response>GetUserData</Response>* *<UserID>1</UserID>* *<UserSN>455dfd54SDF</UserSN>* *<Name></Name>* *<Privilege>User</Privilege>* *<Depart>0</Depart>* *<Enabled>Yes</Enabled>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <UserID>: *User ID, int.* <UserSN>: User sequence number, similar to User ID, string type. <Name></Name>: user name <Privilege>: User, Register, Manager, Administrator. The user's level. <Depart>:0``19. The user's department <Enabled>: Yes/No. this user can use or not <Card>:Card number, 0 or NULL means no enrollment

<Card>9499024</Card> *<PWD>1</PWD>* *<Fingers>2</Fingers>* *<FaceEnrolled>Yes</FaceEnrolled>* *<More>Yes/No</More>* *<Result>OK</Result>* *</Message>*	<PWD>: Password, 0 or NULL means no enrollment *<Fingers>:* total enrollment Fingerprint for the current User. Max is 10 FP. *<FaceEnrolled>：* Yes/No,have already enroll the Face or not *<More>: Yes/No，Yes Yes means there's a next User in the Device. Server should send* GetNextUserData request. <Result>: OK/Fail, result

11) GetUserPassword: Get the password of the specified User ID

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>GetUserPassword</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *<UserID>1</UserID>* *</Message>*	<Request>： *GetUserPassword, get password of the specified User ID.* <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* <UserID>: *User ID, int.*
Device Response	

<?xml version="1.0"?> <Message> <Actid>2</Actid> <TerminalType>PFS100</TerminalType> <HardwareVer>4900</HardwareVer> <TerminalID>1</TerminalID> <DeviceSerialNo>wb2018042801</DeviceSerialNo> <Version>ZD4900 v2.0.180308</Version> <Action_ext>show/xx</Action_ext> <Response>GetUserPassword</Response> <UserID>1</UserID> <Password>4</Password> <Result>OK</Result> </Message>	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <UserID>: *User ID, int.* <Password>: password. <Result>:OK/Fail, result

12) GetUserCardNo: Get the Card number of the specified User ID

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>GetUserCardNo</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *<UserID>1</UserID>* *</Message>*	<Request>: *GetUserCardNo, get the card number of the specified User ID.* <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* <UserID>: *User ID, int.*
Device Response	

<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>* *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>* *<Action_ext>show/xx</Action_ext>* *<Response>GetUserCardNo</Response>* *<UserID>1</UserID>* *<CardNo>0</CardNo>* *<Result>OK</Result>* *</Message>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <UserID>: *User ID, int.* <*CardNo*>: card number. <Result>: OK/Fail, result

13) GetUserPhoto: Get the Enroll Photo of the specified User ID

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>GetUserPhoto</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *<UserID>1</UserID>* *</Message>*	<Request>: *GetUserPhoto, get the enroll photo of the specified User ID.* <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* <UserID>: *User ID, int.*
Device Response	

<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>* *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>* *<Action_ext>show/xx</Action_ext>* *<Response>GetUserPhoto</Response>* *<UserID>1</UserID>* *<PhotoData>Photo data</PhotoData>* *<Result>OK</Result>* *</Message>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <UserID>: *User ID, int.* <*PhotoData*>: enroll photo base64 string. <Result>: OK/Fail, result

14) SetUserPhoto: set Photo to the specified User ID

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>SetUserPhoto</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *<UserID>1</UserID>* *<PhotoSize>5543</PhotoSize>* *<PhotoData>Photo data</PhotoData>* *</Message>*	<Request>: *SetUserPhoto, set photo to the specified User ID* <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* <UserID>: *User ID, int.* <PhotoSize>: photo size <PhotoData>: photo data

Device Response	
<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>* *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>* *<Action_ext>show/xx</Action_ext>* *<Response>SetUserPhoto</Response>* *<UserID>1</UserID>* *<Result>OK</Result>* *</Message>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <UserID>: *User ID, int.* <Result>: OK/Fail, result

15) GetFingerData: Get a specified Fingerprint of the specified User ID

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>GetFingerData</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *<UserID>1</UserID>* *<FingerNo>0</FingerNo>* *</Message>*	<Request>: *GetFingerData, get a specified fingerprint of the specified User ID.* <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* <UserID>: *User ID, int.* <FingerNo>: 0~~9, fingerprint Number
Device Response	

<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>* *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>* *<Action_ext>show/xx</Action_ext>* *<Response>GetFingerData</Response>* *<UserID>1</UserID>* *<FingerNo>1</FingerNo>* *<Duress>Yes</Duress>* *<FingerData>04130</FingerData>* *<Result>OK</Result>* *</Message>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <UserID>: *User ID, int.* <FingerNo>: 0~~9, the Number of the specified fingerprint. <Duress>: Yes/No, the current fingerprint is set for duress fingerprint or not. <FingerData>: fingerprint data. <Result>: OK/Fail, result

16) SetFingerData: Set a specified fingerprint to the specified User ID

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>SetFingerData</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *<UserID>1</UserID>* *<FingerNo>0</FingerNo>* *<DuplicationCheck>Yes</DuplicationCheck>* *<Duress>No</Duress>* *<FingerData>Template data in base64 encoding</FingerData>* *</Message>*	<Request>: *SetFingerData, set a specified fingerprint to the specified User ID* <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* <UserID>: *User ID, int.* <FingerNo>：0~~9, the Number of the specified fingerprint. <DuplicationCheck>: Yes/No, need to check if the enrollment is duplicated? <Duress>：Yes/No, the current fingerprint is set for duress fingerprint or not. <FingerData>: fingerprint data. If NULL, it will delete.
Device Response	
<?xml version="1.0"?> <Message> <Actid>2</Actid> <TerminalType>PFS100</TerminalType> <HardwareVer>4900</HardwareVer> <TerminalID>1</TerminalID> <DeviceSerialNo>wb2018042801</DeviceSerialNo> <Version>ZD4900 v2.0.180308</Version> <Action_ext>show/xx</Action_ext> <Response>SetFingerData</Response>	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly.

<UserID>1</UserID> <FingerNo>0</FingerNo> *<Action>Update</Action>* <Result>OK</Result> </Message>	<UserID>: *User ID, int.* <FingerNo>: 0~~9, the Number of the specified fingerprint. <Action>: Update/Delete, Update – update information, delete - delete. <Result>: OK/Duplicate/Fail, result. When detected the setting fingerprint is duplicated, returns Duplicate.

17) GetFaceData: Get face data of the specified User ID

Server Request	
<?xml version="1.0"?> *<Message>* <*Request*>*GetFace*Data</Request> <Ccid>2</Ccid> <Time>2013-4-11-T11:28:54Z</Time> <Action_ext>show/xx<Action_ext> <UserID>1</User*ID*> *</Message>*	<Request>: *GetFaceData, get face data of the specified User ID.* <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* <UserID>: *User ID, int.*
Device Response	
<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>* *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number

<DeviceSerialNo>wb2018042801</DeviceSerialNo> *<Version>ZD4900 v2.0.180308</Version>* *<Action_ext>show/xx</Action_ext>* *<Response>GetFaceData</Response>* *<UserID>1</UserID>* *<FaceEnrolled>Yes/No</FaceEnrolled>* *<FaceData>sss</FaceData><!-- 40000Byte->* *<Result>OK</Result>* *</Message>*	<Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <UserID>: *User ID, int.* <FaceEnrolled>： Yes/No, have already enroll the Face or not <FaceData>: if already have the enrollment Face, then get the Face data. <Result>: *OK*/Fail, result.

18) SetFaceData: set face data to the specified User ID

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>SetFaceData</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *<UserID>1</UserID>* *<DuplicationCheck>0</DuplicationCheck>* *<FaceData>Template data in base64 encoding</FaceData>* *</Message>*	<Request>: *SetFaceData*, set Face data to the specified User ID. <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* <UserID>: *User ID, int.* <DuplicationCheck>: Yes/No, need to check if the Enrollment is duplicated? <FaceData>: Face data, if NULL, then delete.
Device Response	
<?xml version="1.0"?> *<Message>*	<Actid>: <Actid> is <Ccid>

<Actid>2</Actid> *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>* *<Action_ext>show/xx</Action_ext>* *<Response>SetFaceData</Response>* *<UserID>1</UserID>* *<Action>Update</Action>* *<Result>OK</Result>* *</Message>*	<TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <UserID>: *User ID, int.* <Action>: Update/Delete, Update – update information, delete - delete. <Result>: OK/Duplicate/Fail, result. When detected the setting face is duplicated, returns Duplicate.

19) RemoteEnroll: Remote Enroll

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>RemoteEnroll</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *<UserID>1</UserID>* *<Backup>RemoteEnrollFP</Backup>* *<FingerNo>0</FingerNo>* *</Message>*	<Request>: RemoteEnroll, remote enroll. <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* <UserID>: *User ID, int.* <Backup>: RemoteEnrollFace/RemoteEnrollFP/RemoteEnrollCard, When needs remote to enroll the face through Internet, use RemoteEnrollFace; when needs remote to enroll the fingerprint through Internet, use RemoteEnrollFP; When needs remote to enroll Card throught Internet, use RemoteEnrollCard. <FingerNo>: 0~9, when <Backup>=RemoteEnrollFP, need to use this FingerNo, to specify to enroll that fingerprint.
Device Response	

<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>* *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>* *<Action_ext>show/xx</Action_ext>* *<Response>RemoteEnroll</Response>* *<ResultCode>Success</ResultCode>* *</Message>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <ResultCode>: Success/ EnrollNumberError/ DatabaseFull/ FaceAlreadyEnrolled/ FPAlreadyEnrolled / InvalidFingerNumber / CardAlreadyEnrolled / MenuProcessing / RemoteEnrollAlreadyStarted / Unknown. Success: enable the remote enroll successfully. EnrollNumberError: the input Enroll ID is wrong. DatabaseFull: the enrollment capacity is already full. FaceAlreadyEnrolled: the current User ID has already enrolled the Face. FPAlreadyEnrolled: the current User ID has already enrolled the Fingerprint. InvalidFingerNumber: the input Fingerprint Number is invalid. CardAlreadyEnrolled: the current User ID has already enrolled Card Number. MenuProcessing: the Device is in MENU processing now. RemoteEnrollAlreadyStarted: already started the Remote Enroll. Unknown: Not enable due to unknown reason.

20) ExitRemoteEnroll: Exit the Remote Enroll

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>ExitRemoteEnroll</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *</Message>*	<Request>: *ExitRemoteEnroll, exit the remote enroll.* <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.*
Device Response	
<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>* *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>* *<Action_ext>show/xx</Action_ext>* *<Response>ExitRemoteEnroll</Response>* *<ResultCode>SuccessExitRemoteEnroll/NotStartedRemoteEnroll</Result Code>* *</Message>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <ResultCode>: SuccessExitRemoteEnroll/NotStartedRemoteEnroll。 SuccessExitRemoteEnroll: exist successfully. NotStartedRemoteEnroll: not in the remote enroll status.

21) TakeOffManager: remove the Admin

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>TakeOffManager</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *</Message>*	<Request>: *TakeOffManager, remove the Admin.* <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.*
Device Response	
<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>* *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>* *<Action_ext>show/xx</Action_ext>* *<Response>TakeOffManager</Response>* *<Result>OK</Result>* *</Message>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <Result>: OK/Fail, result.

22) EnableDevice: Enable(Lock) / Disable (Unlock) Device

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>EnableDevice</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *<Enable>Yes/No</Enable>* *</Message>*	<Request>: *EnableDevice ,* <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* <Enable>: Yes/No, Yes: Enable, No: Disable
Device Response	
<?xml version="1.0"?> <Message> <Actid>2</Actid> <TerminalType>PFS100</TerminalType> <HardwareVer>4900</HardwareVer> <TerminalID>1</TerminalID> <DeviceSerialNo>wb2018042801</DeviceSerialNo> <Version>ZD4900 v2.0.180308</Version> <Action_ext>show/xx</Action_ext> <Response>*EnableDevice*</Response> <Result>OK</Result> </Message>	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <Result>: OK/Fail, result.

23) GetTime: Get Device Time

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>GetTime</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *</Message>*	<Request>: *GetTime, get device time.* <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.*
Device Response	
<?xml version="1.0"?> <Message> <Actid>2</Actid> <TerminalType>PFS100</TerminalType> <HardwareVer>4900</HardwareVer> <TerminalID>1</TerminalID> <DeviceSerialNo>wb2018042801</DeviceSerialNo> <Version>ZD4900 v2.0.180308</Version> <Action_ext>show/xx</Action_ext> <Response>GetTime</Response> <Time>2013-4-11-T11:31:18Z</Time> *<UTC>+8:00</UTC>* <Result>OK</Result> </Message>	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <Time>: Device turrent time *<UTC>: timezone of the current time, (some devices not support).* <Result>: OK/Fail, result.

24) SetTime: Set Device Time

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>SetTime</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *<UTC>+8:00</UTC>* *</Message>*	<Request>: *SetTime, Set Device Time.* <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* <UTC>: timezone of the current time, *(some devices not support).*
Device Response	
<?xml version="1.0"?> <Message> <Actid>2</Actid> <TerminalType>PFS100</TerminalType> <HardwareVer>4900</HardwareVer> <TerminalID>1</TerminalID> <DeviceSerialNo>wb2018042801</DeviceSerialNo> <Version>ZD4900 v2.0.180308</Version> <Action_ext>show/xx</Action_ext> <Response>SetTime</Response> <Result>OK</Result> </Message>	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <Result>:OK/Fail, result.

25) GetDepartment: Get the corresponding Department name

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>GetDepartment</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *<DeptNo>0</DeptNo>* *</Message>*	<Request>: *GetDepartment,* Get the corresponding Department name. <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* <DeptNo>: 0~19, Department Number.
Device Response	
<?xml version="1.0"?> <Message> <Actid>2</Actid> <TerminalType>PFS100</TerminalType> <HardwareVer>4900</HardwareVer> <TerminalID>1</TerminalID> <DeviceSerialNo>wb2018042801</DeviceSerialNo> <Version>ZD4900 v2.0.180308</Version> <Action_ext>show/xx</Action_ext> <Response>GetDepartment</Response> <DeptNo>0</DeptNo> <Name></Name> <Result>OK</Result> </Message>	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <DeptNo>: Department Number. <Name>: Department Name. <Result>: OK/Fail, result.

26) SetDepartment: Set the corresponding department name

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>SetDepartment</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *<DeptNo>0</DeptNo>* *<Data>department</Data>* *</Message>*	<Request>: *SetDepartment, set the corresponding department name.* <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* <DeptNo>: 0~19, Department Number. <Data>: Department Name.
Device Response	
<?xml version="1.0"?> <Message> <Actid>2</Actid> <TerminalType>PFS100</TerminalType> <HardwareVer>4900</HardwareVer> <TerminalID>1</TerminalID> <DeviceSerialNo>wb2018042801</DeviceSerialNo> <Version>ZD4900 v2.0.180308</Version> <Action_ext>show/xx</Action_ext> <Response>SetDepartment</Response> <Result>OK</Result> </Message>	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <Result>: OK/Fail, result.

27) GetProxyName: Get the corresponding Proxy Name

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>GetProxyName</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *<ProxyNo>0</ProxyNo>* *</Message>*	<Request>: *GetProxyName, Get the corresponding Proxy Name*. <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* <ProxyNo>: 0~19, Proxy Number.
Device Response	
<?xml version="1.0"?> <Message> <Actid>2</Actid> <TerminalType>PFS100</TerminalType> <HardwareVer>4900</HardwareVer> <TerminalID>1</TerminalID> <DeviceSerialNo>wb2018042801</DeviceSerialNo> <Version>ZD4900 v2.0.180308</Version> <Action_ext>show/xx</Action_ext> <Response>*GetProxyName*</Response> <*ProxyNo*>0</*ProxyNo*> <Name></Name> <Use>Yes<Use> <Result>OK</Result> </Message>	<Actid>: <Actid> is <Ccid> <TerminalType>：Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <*ProxyNo*>: Proxy Number <Name>: Proxy Name <Use>: Yes/No, enable this function or not. <Resu*lt*>: *OK/*Fail, result.

28) SetProxyName: Se the Proxy Name

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>SetProxyName</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *<ProxyNo>0</ProxyNo>* *<Data>Assembling</Data>* <Use>Yes<Use> *</Message>*	<Request>： *SetProxyName, set the Proxy name to the specified Proxy Number.* <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* <*ProxyNo*>: 0~19, Proxy Number <Data>: Proxy name <Use>: use this filed to enable the Proxy function, the filed might not be exist, if not exist, not edit this value.
Device Response	
<?xml version="1.0"?> <Message> <Actid>2</Actid> <TerminalType>PFS100</TerminalType> <HardwareVer>4900</HardwareVer> <TerminalID>1</TerminalID> <DeviceSerialNo>wb2018042801</DeviceSerialNo> <Version>ZD4900 v2.0.180308</Version> <Action_ext>show/xx</Action_ext> <Response>*SetProxyName*</Response> <Result>OK</Result> </Message>	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <Result>: OK/Fail, result.

29) GetBellTime: Get the Bell Time settings

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>GetBellTime</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *</Message>*	<Request>: *GetBellTime*, get the bell time setting of the device. <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.*
Device Response	
<?xml version="1.0"?> *<Message>* <Actid>2</Actid> <TerminalType>PFS100</TerminalType> <HardwareVer>4900</HardwareVer> <TerminalID>1</TerminalID> <DeviceSerialNo>wb2018042801</DeviceSerialNo> <Version>ZD4900 v2.0.180308</Version> <Action_ext>show/xx</Action_ext> *<Response>GetBellTime</Response>* *<BellCount>24</BellCount>* *<Bell_0>23:42, 0, 0, 0</Bell_0>* *<Bell_1>0, 0, 0, 0</Bell_1>*	<Actid>: <Actid> is <Ccid> <TerminalType>：Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <BellCount>: total Bell count <Bell_0>: Group NO.1 bell setting, with 4 values, use "," to separate, ①time,②use,③cycle,④times <Bell_1>

<Bell_2>0, 0, 0, 0</Bell_2> *<Bell_3>0, 0, 0, 0</Bell_3>* *<Bell_4>0, 0, 0, 0</Bell_4>* *...* *<Bell_22>0, 0, 0, 0</Bell_22>* *<Bell_23>0, 0, 0, 0</Bell_23>* <Result>OK</Result> *</Message>*	... <Bell_23> <Result>: OK/Fail, result. ①time:hh:mm。23:42 ②use:use/no use ③cycle:EveryDay/Sun/Mon/Tues/Wed/Thur/Fri/Sat ④times:≤200

30) SetBellTime: set bell time

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>SetBellTime</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *<Bell_0>0, 0, 0, 0</Bell_0>* *<Bell_1>0, 0, 0, 0</Bell_1>* *<Bell_2>0, 0, 0, 0</Bell_2>* *<Bell_3>0, 0, 0, 0</Bell_3>* *<Bell_4>0, 0, 0, 0</Bell_4>* *...* *<Bell_22>0, 0, 0, 0</Bell_22>*	<Request>: *SetBellTime*, set bell time. <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* <Bell_0>: Group NO.1 bell setting, with 4 value, , use "," to separate, ①time,②use,③cycle,④times

<Bell_23>0, 0, 0, 0</Bell_23> *</Message>*	
Device Response	
<?xml version="1.0"?> *<Message>* <Actid>2</Actid> <TerminalType>PFS100</TerminalType> <HardwareVer>4900</HardwareVer> <TerminalID>1</TerminalID> <DeviceSerialNo>wb2018042801</DeviceSerialNo> <Version>ZD4900 v2.0.180308</Version> <Action_ext>show/xx</Action_ext> <Response>*SetBellTime*</Response> *<Result>OK</Result>* *</Message>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <Result>: OK/Fail, result

31) GetDeviceSetting: Get Device Setting

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>*GetDeviceSetting*</Response>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *</Message>*	<Request>: GetDeviceSetting, get device settings <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.*
Device Response	
<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>* *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>* *<Action_ext>show/xx</Action_ext>* *<Response>*GetDeviceSetting*</Response>* *<Language>English</Language>* *<Volume>6</Volume>* *<IdentifyMode>*AnyWay*</IdentifyMode>* *<RealTimePhoto>Yes</RealTimePhoto>* *<Result>OK</Result>* *</Message>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <Language>： *Engl*ish/ChineseSimplified/ChineseTraditional。 <volume>:0~~10, 0 means silent. <IdentifyMode>: <RealTimePhoto>: Yes/No, after verify success, need to take the real-time photo? If need, need to push the real-time photo to the server. <Result>: OK/Fail, set successfully or not.

32) SetDeviceSetting: Set some settings to the Device

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>*SetDeviceSetting*</Response>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *<Language>English</Language>* *<volume>6</volume>* *<IdentifyMode>*AnyWay*</IdentifyMode>* *<RealTimePhoto>Yes</RealTimePhoto>* *</Message>*	<Request>：SetDeviceSetting, set device settings. <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* <Language>：*Engl*ish/ChineseSimplified/ChineseTraditional。 <volume>：0~~10, 0 is silent. <IdentifyMode>: AnyWay/Face/FP/Card/PWD/Face+FP/Face+Card/Face+PWD/FP+Card/FP+PWD/Face+FP+Card/Face+FP+PWD/FP+Card+PWD <RealTimePhoto>:Yes/No,after verify success, need to take the real-time photo? If need, need to push the real-time photo to the server.
Device Response	
<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>* *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>* *<Action_ext>show/xx</Action_ext>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly.

*<Response>*SetDeviceSetting*</Response>* *<Result>OK</Result>* *</Message>*	<Result>: OK/Fail, setting successful or not.

33) RestoreDevice: Restore Device（Factory reset/Restart/Cancel Warning Alarm）

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>RestoreDevice</Response>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *<Action>RestoreFactory</Action>* *</Message>*	<Request>: *RestoreDevice, restore device.* <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* <Action>:RestoreFactory/ReStart/CancelWarning., RestoreFactory – restore to factory setting; Restart: restart the device; CancelWarning- cancel the warning alarm.
Device Response	

<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>* *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>* *<Action_ext>show/xx</Action_ext>* *<Response>RestoreDevice</Response>* *<Action>RestoreFactory</Action>* *<Result>OK</Result>* *</Message>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. *<Action>:* RestoreFactory/ReStart, RestoreFactory - restore to factory settin; restart the device; <Result>: promts that already received this command, Device starts to operate, returns OK while it success, and returns Fail while it failed.

34) GetPowerSetting: Get the Power settings

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>*GetPowerSetting*</Response>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *</Message>*	<Request>: GetPowerSetting, get the Power settings. <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.*
Device Response	
<?xml version="1.0"?> *<Message>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the

<Actid>2</Actid> *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>* *<Action_ext>show/xx</Action_ext>* *<Response>*GetPowerSetting*</Response>* *<RestartTime>00:00</RestartTime>* *<SleepAfter>1</SleepAfter>* *<ScreenSaver>Yes</ScreenSaver>* *<PowerKey>Yes</PowerKey>* *<Result>OK</Result>* *</Message>*	functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <RestartTime>: Schedule restart time <SleepAfter>: Device goes to Sleep mode without operation for some time? <ScreenSaver>: Yes/No Enable Screen Saver or not? <PowerKey>: Yes/No Enable the Power Key or not? <Result>: OK/Fail, result.

35) SetPowerSetting: Set Power settings

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>*SetPowerSetting*</Response>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *<RestartTime>00:00</RestartTime>* *<SleepAfter>1</SleepAfter>* *<ScreenSaver>Yes</ScreenSaver>* *<PowerKey>Yes</PowerKey>*	<Request>: SetPowerSetting, set Power settings. <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* <RestartTime>: Schedule restart time <SleepAfter>: Device goes to Sleep mode without operation for some time? <ScreenSaver>: Yes/No Use the Power Key or not?

</Message>	<PowerKey>: Yes/No Enable the Power Key or not?
Device Response	
<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>* *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>* *<Action_ext>show/xx</Action_ext>* *<Response>*SetPowerSetting*</Response>* *<Result>OK</Result>* *</Message>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <Result>: OK/Fail, result.

36) FirmwareUpgradeByCloud: Upgrade Firmware by Cloud System

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>FirmwareUpgradeByCloud</Response>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *<Size>41</Size>* *<Data>http://download.realandbio.com/d/fw/v2.01</Data>* *</Message>*	<Request>: *FirmwareUpgradeByCloud,* Upgrade Firmware by Cloud System <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* <Size>: length of the firmware address <Data>: the firmware address on the Server CDN.
Device Response	
<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>* *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>* *<Action_ext>show/xx</Action_ext>* *<Response>FirmwareUpgradeByCloud</Response>* *<Result>OK</Result>* *</Message>*	<Actid> : <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <Result>: prompt that already received this command, Device starts to download the firmware on the specified address to the Device, Device will upgrade automatically.

37) GetEthernetSetting: Get Ethernet Settings

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>GetEthernetSetting</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *</Message>*	<Request>: *GetEthernetSetting*, Get Ethernet Settings. <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.*
Device Response	
<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>* *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>* *<Action_ext>show/xx</Action_ext>* *<Response>GetEthernetSetting</Response>* *<MacAddress>00:10:dd:01:08:81</MacAddress>* *<DHCP>Yes</DHCP>* *<IP>192.168.1.102</IP>* *<Subnet>255.255.255.0</Subnet>* *<DefaultGateway>192.168.1.1</DefaultGateway>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <MacAddress>: Ethernet network card MAC physical address <DHCP>: Yes/No, Ethernet is Dynamic IP or not? <IP>: defaulted Ethernet IP while in Static IP. <Subnet>: Defaulted Ethernet Subnet Mask for Static IP. <DefaultGateway>: Defaulted Ethernet Gateway for Static IP. <IP_from_dhcp>: if <DHCP> is Ye, get the current Dynamic IP.

<IP_from_dhcp>192.168.1.15</IP_from_dhcp> *<Subnet_from_dhcp>255.255.255.0</Subnet_from_dhcp>* *<DefaultGateway_from_dhcp>0.0.0.0</DefaultGateway_from_dhcp>* *<Result>OK/Fail</Result>* *</Message>*	<Subnet_from_dhcp>: if <DHCP> is Yes, get the current Dynamic Subnet mask. <DefaultGateway_from_dhcp>: if <DHCP> is yes, get the current Default Gateway. <Result>: OK/Fail, result.

38) SetEthernetSetting: set the Ethernet settings

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>*SetEthernetSetting*</Response>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *<DHCP>Yes/No</DHCP>* *<IP>192.168.1.102</IP>* *<Subnet>255.255.255.0</Subnet>* *<DefaultGateway>192.168.1.1</DefaultGateway>* *</Message>*	<Request>： *SetEthernetSetting,* set the Ethernet settings. <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* <DHCP>: Yes/No, Network is Dynamic IP or not? <IP>: defaulted Ethernet IP while in Static IP. <Subnet>: Defaulted Ethernet Subnet Mask for Static IP. <DefaultGateway>: Defaulted Ethernet Gateway for Static IP.
Device Response	

<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>* *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>* *<Action_ext>show/xx</Action_ext>* *<Response>SetEthernetSetting</Response>* *<Result>OK/Fail</Result>* *</Message>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <Result>: OK/Fail, result

39) GetWifiSetting: Get the WIFI settings

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>GetWifiSetting</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *</Message>*	<Request>：*GetWifiSetting*, Get the WIFI settings <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.*
Device Response	

<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>* *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>* *<Action_ext>show/xx</Action_ext>* *<Response>GetWifiSetting</Response>* *<Use>Yes</Use>* *<Port>5005</Port>* *<SSID>WIFI host</SSID>* *<Key>p123456</Key>* *<DHCP>Yes/No</DHCP>* *<IP>192.168.2.225</IP>* *<Subnet>255.255.255.0</Subnet>* *<DefaultGateway>192.168.2.1</DefaultGateway>* *<IP_from_dhcp>192.168.1.15</IP_from_dhcp>* *<Subnet_from_dhcp>255.255.255.0</Subnet_from_dhcp>* *<DefaultGateway_from_dhcp>0.0.0.0</DefaultGateway_from_dhcp>* *<Result>OK/Fail</Result>* *</Message>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <Use>: Yes/No, if WIFI or not? <Port>5005</Port>: WIFI network port <SSID>: WiFi name <Key>： WiFi password <DHCP>: Yes/No, WIFI network is dynamic IP or not? <IP>: defaulted WIFI network IP while in Static IP. <Subnet>: Defaulted WIFI Subnet Mask for Static IP. <DefaultGateway>: Defaulted WIFI Gateway for Static IP. <IP_from_dhcp>: if <DHCP> is Yes, get the current Dynamic IP. <Subnet_from_dhcp>: if <DHCP> is Yes, get the current Dynamic Subnet mask. <DefaultGateway_from_dhcp> if <DHCP> isYes, get the current Default Gateway. <Result>: OK/Fail, result.

40) SetWifiSetting: Set WIFI network setting

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>SetWifiSetting</Response>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *<Use>Yes/No</Use>* *<SSID>xxxx</SSID>* *<Key>xxxxxx</Key>* *<DHCP>Yes/No</DHCP>* *<IP>192.168.2.225</IP>* *<Subnet>255.255.255.0</Subnet>* *<DefaultGateway>192.168.2.1</DefaultGateway>* *</Message>*	<Request>: SetWifiSetting, Set WIFI network setting <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* <Use>: Yes/No, if WIFI or not? <SSID>: WiFi name <Key>: WiFi password <DHCP>: Yes/No, network is Dynamic IP or not? <IP>: defaulted network IP while in Static IP. <Subnet>: Defaulted Subnet Mask for Static IP. <DefaultGateway>: Defaulted Gateway for Static IP.
Device Response	
<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>* *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly.

<Action_ext>show/xx</Action_ext> *<Response>*SetWifiSetting*</Response>* *<Result>OK/Fail</Result>* *</Message>*	<Response>: respond the requested value directly. <Result>: OK/Fail, result

41) GetMobileNetSetting: Get Mobile network settings

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>*GetMobileNetSetting*</Response>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *</Message>*	<Request>: GetMobileNetSetting, Get Mobile network settings. <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.*
Device Response	
<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>* *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version

<Version>ZD4900 v2.0.180308</Version> *<Action_ext>show/xx</Action_ext>* *<Response>*GetMobileNetSetting*</Response>* *<Use>Yes</Use>* *<IP>10.25.127.12</IP>* *<Operator>CMCC</Operator>* *<Result>OK/Fail</Result>* *</Message>*	<Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. *<Use>: Yes/No, Use or not?* *<IP>: connected to network, IP address, if not connected, it's 0.0.0.0* *<Operator>: mobile network operator. If not connected, NULL.* <Result>: OK/Fail, success or fail.

42) SetMobileNetSetting: Set mobile network settings

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>*SetMobileNetSetting*</Response>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *<Use>Yes</Use>* *</Message>*	<Request>: SetMobileNetSetting, Set mobile network settings. <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* *<Use>: Yes/No, use or not.*
Device Response	
<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>* *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version

<Version>ZD4900 v2.0.180308</Version> *<Action_ext>show/xx</Action_ext>* *<Response>*SetMobileNetSetting*</Response>* *<Result>OK/Fail</Result>* *</Message>*	<Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <Result>: OK/Fail, set success or not.

43) GetVPNServer: Get VPN Virtual Local Area Network setting

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>*GetVPNServer*</Response>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *</Message>*	<Request>: GetVPNServer, Get VPN Virtual Local Area Network setting. <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.*
Device Response	
<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>* *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly.

<Action_ext>show/xx</Action_ext> *<Response>*GetVPNServer*</Response>* *<VPNAddress>sj.realtime.cn</VPNAddress>* *<Account>WENS</Account>* *<Password>123456</Password>* *<IP>10.25.127.12</IP>* *<Result>OK/Fail</Result>* *</Message>*	<Response>: respond the requested value directly. *<VPNAddress>: VPN Server address, IP or Domain name* *<Account>: VPN Account* *<Password>: VPN password* *<IP>: when connected successfully, it shows the DNS as IP.* <Result>: OK/Fail, success or not.

44) SetVPNServer: Set VPN Virtual Local Area Network setting

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>*SetVPNServer*</Response>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *<VPNAddress>sj.realtime.cn</VPNAddress>* *<Account>WENS</Account>* *<Password>123456</Password>* *</Message>*	<Request>: SetVPNServer, set VPN Virtual Local Area Network setting <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* <VPNAddress>: VPN Server Address, IP or Domain name. <Account>: VPN Account <Password>: VPN password
Device Response	
<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>* *<TerminalType>PFS100</TerminalType>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform

<HardwareVer>4900</HardwareVer> *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>* *<Action_ext>show/xx</Action_ext>* *<Response>*GetVPNServer*</Response>* *<Result>OK/Fail</Result>* *</Message>*	<TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <Result>: OK/Fail, set successfully or not.

45) GetGPS: Get the GPS Location Data

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>*GetGPS*</Response>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *<Alive>No</Alive>* *</Message>*	<Request>: GetGPS, Get the GPS Location Data <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* <Alive>: Yes/No,Yes means the Server will not send the request again, GPS need to keep alive to send the real-time data back, for the return data, the Server will not give response of receiveing.
Device Response	
<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not.

<TerminalType>PFS100</TerminalType> *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>* *<Action_ext>show/xx</Action_ext>* *<Response>*GetGPS*</Response* *<Longitude>-25.15465</Longitude>* *<Latitude>-14.445</Latitude>* *<SatelliteCount>19</SatelliteCount>* *<Result>OK/Fail</Result>* *</Message>*	<HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <Longitude>: Longitude, West Longitude as Negative. Example: “-25.12”. <Latitude>: Latitude, South Latitude as Negative. Example: “-12.548”. <SatelliteCount>: the connected Satellite Count. <Result>: OK/Fail, success or not. When Alive=Yes, Device keeps sending this information to the Server. The Longitude and Latitude is real-time value, others keeps the same.

46) SetGPS: Set GPS

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>S*etGPS*</Response>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *<Use>No</Use>* *<Alive>No</Alive>* *</Message>*	<Request>: *S*etGPS, Set the GPS. <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* *<Use>: Enable or not.* <Alive>: Yes/No, Yes means Server will not send the request again, GPS need to

	keep alive to send the real-time data back, for the return data, the Server will not give response of receiveing.
Device Response	
<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>* *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>* *<Action_ext>show/xx</Action_ext>* *<Response>*SetGPS*</Response>* *<Result>OK/Fail</Result>* *</Message>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <Result>: OK/Fail, success or not. When Alive=Yes, Device keeps sending GetGPS Response information to the Server. The Longitude and Latitude is real-time value, others keeps the same.

47) GetCloudServer: Get WebSocket Cloud Server settings

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>*GetCloudServer*</Response>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *</Message>*	<Request>: GetCloudServer, Get WebSocket Cloud Server settings <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.*
Device Response	
<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>* *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>* *<Action_ext>show/xx</Action_ext>* *<Response>*GetCloudServer*</Response>* *<WebsocketServer>wss://ws.realadmin.cn</WebsocketServer>* *<Result>OK/Fail</Result>* *</Message>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <*WebsocketServer*>: WebSocket Server address <Result>: OK/Fail, success or not.

48) SetCloudServer: Set WebSocket Cloud Server settings

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>*SetCloudServer*</Response>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *<WebsocketServer>wss://ws.realadmin.cn</WebsocketServer>* *</Message>*	<Request>: SetCloudServer, Set WebSocket Cloud Server settings. <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* <*WebsocketServer>: W*ebSocket Server address
Device Response	
<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>* *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>* *<Action_ext>show/xx</Action_ext>* *<Response>*SetCloudServer*</Response>* *<Result>OK/Fail</Result>* *</Message>*	<Actid>: <Actid> is <Ccid> <TerminalType>:Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <Result>: OK/Fail, success or not.

49) GetStreamingServer: Get the Streaming Server settings

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>GetStreamingServer</Response>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *</Message>*	<Request>: G*etStreamingServ*er, Get the Streaming Server settings <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.*
Device Response	
<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>* *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>* *<Action_ext>show/xx</Action_ext>* *<Response>GetStreamingServer</Response>* *<StreamingServer>https://streaming.real.alive</StreamingServer>* *<Result>OK/Fail</Result>* *</Message>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <*StreamingServer*>: Streaming Server address. <Result>: OK/Fail, success or not.

50) SetStreamingServer: Set the Streaming Server settings

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>SetStreamingServer</Response>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *<StreamingServer>https://streaming.real.alive</StreamingServer>* *</Message>*	<Request>: *SetStreamingServer,* Set the Streaming Server settings <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* *<StreamingServer>: Streaming Server address*
Device Response	
<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>* *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>* *<Action_ext>show/xx</Action_ext>* *<Response>SetStreamingServer</Response>* *<Result>OK/Fail</Result>* *</Message>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <Result>: OK/Fail, success or not.

51) GetLocalServer: Get Device Local Server settings

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>*GetLocalServer*</Response>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *</Message>*	<Request>: GetLocalServer, Get Device Local Server settings <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.*
Device Response	
<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>* *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>* *<Action_ext>show/xx</Action_ext>* *<Response>*GetLocalServer*</Response>* *<TerminalPassword>0</TerminalPassword>* *<TerminalPort>5500</TerminalPort>* *<Result>OK/Fail</Result>* *</Message>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <TerminalPassword>: Device TCP communication password <TerminalPort>: Device Port <Result>: OK/Fail, success or not.

52) SetLocalServer: Set Device Local Server

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>*SetLocalServer*</Response>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *<TerminalPassword>0</TerminalPassword>* *<TerminalPort>5500</TerminalPort>* *<TerminalID>1</TerminalID>* *</Message>*	<Request>: SetLocalServer, Set Device Local Server. <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* <TerminalPassword>: Device TCP communication password <TerminalPort>: Device Port <TerminalID>: Device ID
Device Response	
<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>* *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>* *<Action_ext>show/xx</Action_ext>* *<Response>*SetLocalServer*</Response>* *<Result>OK/Fail</Result>* *</Message>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <Result>: OK/Fail, success or not.

53) EmptyTimeLog: Remove all the Time Attendance Records

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>EmptyTimeLog</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *<Compulsion>Yes</Compulsion>* *</Message>*	<Request>: *EmptyTimeLog*, Remove all the Time Attendance Record. <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* <Compulsion>: Yes/No. Yes: remove all the data compulsively. No: when there's coming data, not remove.
Device Response	
<?xml version="1.0"?> <Message> <Actid>2</Actid> <TerminalType>PFS100</TerminalType> <HardwareVer>4900</HardwareVer> <TerminalID>1</TerminalID> <DeviceSerialNo>wb2018042801</DeviceSerialNo> <Version>ZD4900 v2.0.180308</Version> <Action_ext>show/xx</Action_ext> <Response>EmptyTimeLog</Response> <Result>OK</Result> </Message>	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <Result>：OK/Exist Unsent TimeLog/Fail, result. When Compulsion is No, Device has some data still not sending, returns Exist Unsent TimeLog.

54) EmptyManageLog: Remove all the admin records

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>EmptyManageLog</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *<Compulsion>Yes</Compulsion>* *</Message>*	<Request>: *EmptyManageLog*, remove all the admin records. <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* <Compulsion>, Yes/No. Yse: remove all the data compulsively. No: when there's coming data, not remove.
Device Response	
<?xml version="1.0"?> <Message> <Actid>2</Actid> <TerminalType>PFS100</TerminalType> <HardwareVer>4900</HardwareVer> <TerminalID>1</TerminalID> <DeviceSerialNo>wb2018042801</DeviceSerialNo> <Version>ZD4900 v2.0.180308</Version> <Action_ext>show/xx</Action_ext> <Response>EmptyManageLog</Response> <Result>OK</Result> </Message>	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <Result>: OK/Exist Unsent ManageLog/Fail, result. When Compulsion is No, Device has some data still not sending, returns Exist Unsent TimeLog.

55) EmptyUserEnrollmentData: Remove all the User Enrollment Data

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>EmptyUserEnrollmentData</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *</Message>*	<Request>: *EmptyUserEnrollmentData,* Remove all the User Enrollment Data. <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.*
Device Response	
<?xml version="1.0"?> <Message> <Actid>2</Actid> <TerminalType>PFS100</TerminalType> <HardwareVer>4900</HardwareVer> <TerminalID>1</TerminalID> <DeviceSerialNo>wb2018042801</DeviceSerialNo> <Version>ZD4900 v2.0.180308</Version> <Action_ext>show/xx</Action_ext> <Response>EmptyUserEnrollmentData</Response> <Result>OK</Result> </Message>	<Actid>: <Actid> is <Ccid> <TerminalType>：Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <Result>: OK/Fail, result.

56) EmptyAllData: Remove all Data

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>EmptyAllData</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *<Compulsion>Yes</Compulsion>* *</Message>*	<Request>: *EmptyAllData*, remove all data. <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* <Compulsion>: Yes/No, Yes: remove all the data compulsively,No: when there's coming data, not remove.
Device Response	
<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>* *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>* *<Action_ext>show/xx</Action_ext>* *<Response>EmptyAllData</Response>* *<Result>OK</Result>* *</Message>*	<Actid>: <Actid> is <Ccid> <TerminalType>:Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <Result>: OK/Exist Unsent Log/Fail, result. When Compulsion is No, Device has some data still not sending. Returns Exist Unsent Log.

57) TimeLog: Time Attendance Record which is Real-time pushed to the Server

Device Request	
<?xml version="1.0"?> *<Message>* *<Rrid>2</Rrid>* *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<ProductName>WO491</ProductName>* *<DeviceSerialNo>wb2018042802</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>* *<Event>TimeLog</Event>* *<LogID>1</LogID>* *<Time>2018-05-09-T20:07:33Z</Time>* *<UserID>1</UserID>* *<Action>FP</Action>* *<AttendStat></AttendStat>* *<APStat>None</APStat>* *<JobCode>0</JobCode>* *<Photo>No</Photo>* *<LogImage>Photo data in base64 encoding</LogImage>* *</Message>*	<*Rrid*>: Session sequence number. <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID *<ProductName>:* <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <*Event*>: respond the requested value directly. <LogID>: Log number <Time>: the time of the attendance Log. <UserID>: the User ID of the attendance log. <Action>: the Attendance mode of the attendance log. <AttendStat>: attendance type. <APStat>: duress <JobCode>: Jode Code <Photo>: has real-time photo or not <LogImage>: real-time photo data *AttendStat:* [Duty On], [Duty Off], [Overtime On], [Overtime Off], [Go Out On], [Go Out Off] <Action>: [FACE], [FP]Fingerprint, [CD]Card, [PWD]Password, [RemoteCardAtt]Remote Card Attendance, the UserID is Card Number,

	[FACE+CD]Face+Card, [FACE+PWD]Face+Password, [FACE+FP]Face+Fingerprint, [CertificateCard]Certificated Card
Server Response	
<?xml version="1.0"?> *<Message>* *<Response>TimeLog</Response>* *<Actid>2</Actid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Result>OK/Fail</Result>* *</Message>*	<*Response*>: *TimeLog, real-time attendance record.* <*Actid*>: <*Actid*> = <*Rrid*> Session sequence number <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Result>: OK/Fail, result.

58) AdminLog: Admin Log which is real-time pushed to the Server

Device Request	
<?xml version="1.0"?> *<Message>* *<Rrid>44250</Rrid>* *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<ProductName>WO491</ProductName>* *<DeviceSerialNo>wb2018042802</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>* *<Event>AdminLog</Event>*	<*Rrid*>: Session sequence number. <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID *<ProductName>:* <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <*Event*>: respond the requested value directly. <LogID>: Log number

<LogID>1</LogID> *<Time>2018-05-10-T12:35:14Z</Time>* *<AdminID>1</AdminID>* *<UserID>2</UserID>* *<Action>BackupFP</Action>* <Stat>8</Stat> *</Message>*	<Time>: the time of the Log. *<AdminID>: Admin User ID, if no Admin to the device, this value is 0.* *<UserID>: User ID* *<Action>*：the action of the admin log, refer to the following [Action Table] for details. <Stat>: *0~10*， when *<Action>=BackupFP, this value is the Enrolled Finger Number FingerNo.*
Server Response	
<?xml version="1.0"?> *<Message>* *<Response>AdminLog</Response>* *<Actid>2</Actid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Result>OK/Fail</Result>* *</Message>*	*<Response>*: *AdminLog : real-time admin log* *<Actid>*: *<Actid>* = *<Rrid>* Session sequence number. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Result>: OK/Fail, result.

Action Table

"Unknown"	
BackupFP	*Enroll Fingerprint*
EnrollUserPWD	*Enroll Password*
EnrollCard	*Enroll Card*
EnrollFace	*Enroll Face*
DeleteFace	*Delete Face*
DeleteFP	*Delete Fingerprint*
DeletePWD	*Delete Password*
DeleteCard	*Delete Card*
DeleteAll	*Delete a specified User*
DeleteAllLog	*Delete All Log(Attendance Log and Admin Log)*

DeleteAllEnoll	*Delete All Enroll*
SettingChanged	*Change Device settings*
SetTime	*Change Device Time*
TakeOffManager	*Remove Admin*
Restore	*Restore factory settings*
boot	*Power on*
DownloadFirmwareSucccess	*Download firmware success*
DoorOpenTimeoutAlarm	*Door Open Time out Alarm*
IllegalOpenAlarm	*Illegal Door Open Alarm(Open Door without verification)*
DuressAlarm	*Duress Alarm*
LinkageAlarm	*Linkage Alarm*
TamperAlarm	*Tamper Alarm*
BlackListAlarm	*BlackList Alarm*
DriveLock	*Open Lock*
NoDriveLock	*Close Lock*
DoorSensorOpen	*Door Sensor Open*
DoorSensorClose	*Door Sensor Close*

59) GetAttendanceLog: Get Attendance Log of the specified User ID in specified time

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>GetAttendanceLog</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *<UserID>1</UserID>* *<StartTime>2017-12-06-T00:00:00Z</StartTime>* *<EndTime>2017-12-16-T23:59:59Z</EndTime>* *<LogIDPos>0</LogIDPos>* *</Message>*	<Request>：*GetAttendanceLog*, Get Attendance Log of the specified User ID in specified time. <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* <UserID>: Get attendance log of the specified User ID. Options. When NULL it's getting attendance log of all User ID. <StartTime>: StartTime of the getting attendance log, Options. When NULL, it's getting all the attendance log before the EndTime. <EndTime>: EndTime of the getting attendance log, Options. When NULL, it's getting all the attendance log begins from the StartTime. <LogIDPos>: Options. 0 or NULL, it's for all log; when it's not NULL or not 0, it returns all the logs that match all the condition mentioned above with the same LogID. This is applied to the following situation: after first time getting mass data, there's some LogID missing, need to re-getting a specified Log, for example sending LogIDPos=78, request Device to send the NO.78 Log that matching the condition mentioned above.
Device Response	
<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not.

<TerminalType>PFS100</TerminalType> *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>* *<Action_ext>show/xx</Action_ext>* *<Response>GetAttendanceLog</Response>* *<AttendanceLog Count="2" LogIDBeginPos="1">* *<Log>* *<LogID>1</LogID>* *<Time>2018-05-08-T19:39:43Z</Time>* *<UserID>1</UserID>* *<Action>FACE</Action>* *<AttendStat></AttendStat>* *<APStat>None</APStat>* *<JobCode>0</JobCode>* *<Photo>No</Photo>* *</Log>* *<Log>* *<LogID>2</LogID>* *<Time>2018-05-08-T19:43:19Z</Time>* *~~~* *<Photo>No</Photo>* *</Log>* *</AttendanceLog>* *<Result>OK</Result>* *</Message>*	<HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <AttendanceLog Count="10" LogIDBeginPos="1">: Count: it shows total count of the sending log, Server use it to judge if it receive the full logs or not. LogIDBeginPos: it will send the matching log begins from this LogID, it use for the following situation: when Server request to get a specified log. For example: Count=”1” LogIDBeginPos=”78”, it means only send logs with LogID=78. <Log>: subelement of log <LogID>: the Log Number <Time>: the time of the Log. <UserID>: the User ID of the attendance log. <Action>: the Attendance mode of the attendance log. <AttendStat>: attendance type. <APStat>: duress <JobCode>: Jod Code. <Photo>: real-time photo <Result>: OK/Exist Unsent Log/Fail, result.

60) GetLogSetting: Get Log settings

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>*GetLogSetting*</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *</Message>*	<Request>: GetLogSetting, get log settings. <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.*
Device Response	
<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>* *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>* *<Action_ext>show/xx</Action_ext>* *<Response>*GetLogSetting*</Response>* *<Att*LogWarning*>1000</Att*LogWarning> *<Manager*LogWarning*>100</Manager*LogWarning> *<ReVerifyTime>3</ReVerifyTime>* *<Result>OK</Result>* *</Message>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <*Att*LogWarning>: Options. It gives warning alarm while the free capacity space of Attendance Log reaches this number. <*Manager*LogWarning>: Options. It gives warning alarm while the free capacity space of Admin Log reaches this number. <*ReVerifyTime*>: Options. During the setting time range, Device will not keep the Repeated Verification log. (minutes) <Result>: OK/Exist Unsent Log/Fail, result.

61) SetLogSetting: Set Log settings

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>*SetLogSetting*</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* <*Att*LogWarning>*1000*</*Att*LogWarning> <*Manager*LogWarning>*100*</*Manager*LogWarning> *<ReVerifyTime>3</ReVerifyTime>* *</Message>*	<Request>: SetLogSetting, set Log settings. <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* <*Att*LogWarning>: Options. It gives warning alarm while the free capacity space of Attendance Log reaches this number. <*Manager*LogWarning>: Options. It gives warning alarm while the free capacity space of Admin Log reaches this number. <*ReVerifyTime*>: Options. During the setting time range, Device will not keep the Repeated Verification log. (minutes)
Device Response	
<?xml version="1.0"?> *<Message>* *<Actid>2</Actid>* *<TerminalType>PFS100</TerminalType>* *<HardwareVer>4900</HardwareVer>* *<TerminalID>1</TerminalID>* *<DeviceSerialNo>wb2018042801</DeviceSerialNo>* *<Version>ZD4900 v2.0.180308</Version>* *<Action_ext>show/xx</Action_ext>* *<Response>*SetLogSetting*</Response>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <Result>: OK/Fail, result.

<Result>OK</Result> *</Message>*	

62) GetAttendanceRule：Get Attendance Rules settings

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>GetAttendanceRule</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *</Message>*	<Request>: *GetAttendanceRule, Get Attendance Rules settings.* <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.*
Device Response	
<?xml version="1.0"?> *<Message>* <Actid>2</Actid> <TerminalType>PFS100</TerminalType> <HardwareVer>4900</HardwareVer> <TerminalID>1</TerminalID> <DeviceSerialNo>wb2018042801</DeviceSerialNo> <Version>ZD4900 v2.0.180308</Version> <Action_ext>show/xx</Action_ext> <Response>*GetAttendanceRule*</Response>	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <TimeSection_0>: StartTime, EndTime, Status.

<TimeSection_0>0, 0, 0</TimeSection_0> <TimeSection_1>0, 0, 0</TimeSection_1> <TimeSection_2>0, 0, 0</TimeSection_2> ... <TimeSection_23>0, 0, 0</TimeSection_23> </Message>	StartTime: 00:00 EndTime: 00:00 Status: "Duty On""Duty Off""Overtime On""Overtime Off""In""Out" <Result>: OK/Exist Unsent Log/Fail, result.

63) SetAttendanceRule: Set Attendance Rules

Server Request	
<?xml version="1.0"?> <Message> <Request>SetAttendanceRule</Request> <Ccid>2</Ccid> <Time>2013-4-11-T11:28:54Z</Time> <Action_ext>show/xx<Action_ext> <TimeSection_0>0, 0, 0</TimeSection_0> <TimeSection_1>0, 0, 0</TimeSection_1> <TimeSection_23>0, 0, 0</TimeSection_23> </Message>	<Request>: *SetAttendanceRule, Set Attendance Rules* <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* <TimeSection_0>: StartTime, EndTime, Status. StartTime: 00:00 EndTime: 00:00 Status: "Duty On""Duty Off""Overtime On""Overtime Off""In""Out"
Device Response	

<?xml version="1.0"?> *<Message>* <Actid>2</Actid> <TerminalType>PFS100</TerminalType> <HardwareVer>4900</HardwareVer> <TerminalID>1</TerminalID> <DeviceSerialNo>wb2018042801</DeviceSerialNo> <Version>ZD4900 v2.0.180308</Version> <Action_ext>show/xx</Action_ext> <Response>*GetAttendanceRule*</Response> *<Result>OK</Result>* *</Message>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <Result>: OK/Fail, result.

64) AccessStatus: Current Access Control Status

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>*AccessStatus*</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *</Message>*	<Request>: AccessStatus, Current Access Control Status. <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.*
Device Response	

<?xml version="1.0"?> *<Message>* <Actid>2</Actid> <TerminalType>PFS100</TerminalType> <HardwareVer>4900</HardwareVer> <TerminalID>1</TerminalID> <DeviceSerialNo>wb2018042801</DeviceSerialNo> <Version>ZD4900 v2.0.180308</Version> <Action_ext>show/xx</Action_ext> *<Response>*AccessStatus*</Response>* *<LockRelease*Status*>Yes</LockRelease*Status> *<DoorSenso*Status*>Open</DoorSenso*Status> *<Alarm*Status*>None</Alarm*Status> *<Result>OK/Fail</Result>* *</Message>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. *<LockRelease*Status*>: Yes/No. Yes – Open Lock.* *<Door*SensoStatus>: Open/Close. Close – Close Lock. <AlarmStatus>: DoorNotClosed/IllegalOpen/Duress/Tamper/Linkage/BlackList. Alarm Status. DoorNotClosed: Alarm if Door not closed. IllegalOpen: Alarm if Door Open without any legal verification. Duress: Duress Alarm Tamper: Tamper Alarm Linkage: Linkage Alarm. BlackList: Blacklist Alarm. <Result>: OK/Exist Unsent Log/Fail, result.

65) GetAccessSetting: Get Access Control Settings

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>*GetAccessSetting*</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *</Message>*	<Request>: GetAccessSetting, Get Access Contorl Settings <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.*
Device Response	
<?xml version="1.0"?> *<Message>* <Actid>2</Actid> <TerminalType>PFS100</TerminalType> <HardwareVer>4900</HardwareVer> <TerminalID>1</TerminalID> <DeviceSerialNo>wb2018042801</DeviceSerialNo> <Version>ZD4900 v2.0.180308</Version> <Action_ext>show/xx</Action_ext> *<Response>*GetAccessSetting*</Response>* *<LockReleaseTime>*CustomOpen*</LockReleaseTime>* *<VerifyPassLockReleaseTime>5</VerifyPassLockReleaseTime>* *<PassTime>00:00-23:59</PassTime>* *<AllWeek>No</AllWeek>* *<StartWeek>Sun</StartWeek>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <LockReleaseTime>:AlwayClose/AlwayOpen/Custom. AlwayClose - Normal Closed, no matter what rules, door will not open. AlwayOpen- Normal Open. CustomOpen - Customize the Door Open Time. *<VerifyPassLockReleaseTime>: when* <LockReleaseTime>=CustomOpen is valid, the time duration to keep door opening after verify success. <PassTime>: when <LockReleaseTime>=CustomOpen is valid, the door keeps

<EndWeek>Sun</EndWeek> *<DoorOpenTimeout>20</DoorOpenTimeout>* *<DoorSensorType>None</DoorSensorType>* *<DuressAlarm>Yes</DuressAlarm>* *<LinkageAlarm>Yes</LinkageAlarm>* *<TamperAlarm>Yes</TamperAlarm>* *<BlackListAlarm>Yes</BlackListAlarm>* *<WGOutputFormat>26</WGOutputFormat>* *<WGOutputContent>UserID</WGOutputContent>* *<Result>OK/Fail</Result>* *</Message>*	opening during this time duration. *<AllWeek>: Yes/No, when* <LockReleaseTime>=CustomOpen is valid, If valid the whole week or not? If yes, Door can be open everyday during the <PassTime>. *<StartWeek>: when* <LockReleaseTime>=CustomOpen is valid, Starts at which day? (Monday - Sunday) *<EndWeek>: when* <LockReleaseTime>=CustomOpen is valid, Ends at which day? (Monday - Sunday) <DoorOpenTimeout>: Door Open Time Out <DoorSensorType>:None/NormallyClose/NormallyOpen. None – No use; NormallyClose – Normally Close Type; NormallyOpen – Normally Open Type. <DuressAlarm>:Yes/No, Enable Duress Alarm or not. <LinkageAlarm>:Yes/No, Enable Linkage Alarm or not. <TamperAlarm>:Yes/No, Enable Tamper Alarm or not. <BlackListAlarm>:Yes/No, Enable BlackList Alarm or not. <WGOutputFormat>:26/34, Wiegand output format, Wiegand 26 & Wiegand 34 <WGOutputContent>:UserID/UserIDorCard, Wiegand output content, UserID? UserID or Card? <Result>: OK/Exist Unsent Log/Fail, result.

66) SetAccessSetting: Set Access Control Settings

Server Request	
<?xml version="1.0"?> *<Message>*	<Request>: SetAccessSetting, set access control settings. <Ccid>: Session sequence number, long type, increment by increase 1.

*<Request>*SetAccessSetting*</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *<LockReleaseTime>*CustomOpen*</LockReleaseTime>* *<VerifyPassLockReleaseTime>5</VerifyPassLockReleaseTime>* *<PassTime>00:00-23:59</PassTime>* *<AllWeek>No</AllWeek>* *<StartWeek>Sun</StartWeek>* *<EndWeek>Sun</EndWeek>* *<DoorOpenTimeout>20</DoorOpenTimeout>* *<DoorSensorType>None</DoorSensorType>* *<DuressAlarm>Yes</DuressAlarm>* *<LinkageAlarm>Yes</LinkageAlarm>* *<TamperAlarm>Yes</TamperAlarm>* *<BlackListAlarm>Yes</BlackListAlarm>* *<WGOutputFormat>26</WGOutputFormat>* *<WGOutputContent>UserID</WGOutputContent>* *</Message>*	<Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* <LockReleaseTime>:AlwayClose/AlwayOpen/Custom. AlwayClose - Normal Closed, no matter what rules, door will not open. AlwayOpen- Normal Open. CustomOpen - Customize the Door Open Time. *<VerifyPassLockReleaseTime>: when* <LockReleaseTime>=CustomOpen is valid, the time duration to keep door opening after verify success. <PassTime>: when <LockReleaseTime>=CustomOpen is valid, the door keeps opening during this time duration. *<AllWeek>:Yes/No, when* <LockReleaseTime>=CustomOpen is valid, If valid the whole week or not? If yes, Door can be open everyday during the <PassTime>. *<StartWeek>: when* <LockReleaseTime>=CustomOpen is valid, Starts at which day? (Monday - Sunday) *<EndWeek>: when* <LockReleaseTime>=CustomOpen is valid, Ends at which day? (Monday - Sunday) *<DoorOpenTimeout>:* Door Open Time Out *<DoorSensorType>:None/NormallyClose/NormallyOpen.None* None – No use; NormallyClose – Normally Close Type; NormallyOpen – Normally Open Type. *<DuressAlarm>:Yes/No*，Enable Duress Alarm or not. *<LinkageAlarm>:Yes/No*，Enable Linkage Alarm or not. *<TamperAlarm>:Yes/No*，Enable Tamper Alarm or not. *<BlackListAlarm>:Yes/No*，Enable BlackList Alarm or not. *<WGOutputFormat>:26/34*，Wiegand output format, Wiegand 26 & Wiegand 34 *<WGOutputContent>:UserID/UserIDorCard*，Wiegand output content, UserID?

	UserID or Card?
Device Response	
<?xml version="1.0"?> *<Message>* <Actid>2</Actid> <TerminalType>PFS100</TerminalType> <HardwareVer>4900</HardwareVer> <TerminalID>1</TerminalID> <DeviceSerialNo>wb2018042801</DeviceSerialNo> <Version>ZD4900 v2.0.180308</Version> <Action_ext>show/xx</Action_ext> *<Response>*SetAccessSetting*</Response>* *<Result>OK/Fail</Result>* *</Message>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <Result>: OK/Fail, result.

67) GetAccessList: Get Access Control Rules List

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>*GetAccessList*</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *</Message>*	<Request>: GetAccessList, Get Access Control Rules List <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.*
Device Response	
<?xml version="1.0"?> *<Message>* <Actid>2</Actid> <TerminalType>PFS100</TerminalType> <HardwareVer>4900</HardwareVer> <TerminalID>1</TerminalID> <DeviceSerialNo>wb2018042801</DeviceSerialNo> <Version>ZD4900 v2.0.180308</Version> <Action_ext>show/xx</Action_ext> *<Response>*GetAccessList*</Response>* <AccessList *Count="2"*> <List ID=”1”> *<Use>Yes</Use>* *<AllDateTime>No</AllDateTime>* *<StartDateTime>2000-01-01-T00:00M</StartDateTime>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <AccessList *Count="2">:* AccessList – access control rules list; *Count – rules count.* <List ID=”1”>*: rule Number.* *<Use>: Yes/No， Enable the Rule or not?* *<AllDateTime>:Yes/No，Yes – everyday, No – use the date & time range below.* *<StartDateTime>：Start Date & Time(hour:minute) of the Rule.*

<EndDateTime>2020-12-31-T23:59M</EndDateTime> *<AllWeek>Yes</AllWeek>* *<StartWeek>Sun</StartWeek>* *<EndWeek>Sun</EndWeek>* <Department>*1*</Department> <Access>*Yes*</Access> </List> <List ID=”2”> *<Use>No</Use>* *<AllDateTime>No</AllDateTime>* *<StartDateTime>2000-01-01-T00:00M</StartDateTime>* *<EndDateTime>2020-12-31-T23:59M</EndDateTime>* *<AllWeek>Yes</AllWeek>* *<StartWeek>Sun</StartWeek>* *<EndWeek>Sun</EndWeek>* <Department>*2*</Department> <Access>*Yes*</Access> </List> </AccessList> *<Result>OK/Fail</Result>* *</Message>*	*<EndDateTime>: End Date & Time(hour:minute) of the Rule.* *<AllWeek>:Yes/No，Yes – every day，No – use the day range below. (Start from which day of the week. End at which day of the week.)* *<StartWeek>: Start Day of the Rule (Monday – Sunday)* *<EndWeek>: End Day of the Rule (Monday - Sunday)* <Department>*: Department Number.* <Access>*: Yes/No,Yes – can open the door.* <Result>: OK/Exist Unsent Log/Fail, result.

68) SetAccessList: Set Access Control Rules List

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>*SetAccessList*</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *<Action>Add/Edit/Delete<Action>* <AccessListID>*1*<AccessListID> <AccessList *Count="2"*> <List ID=”1”> *<Use>Yes</Use>* *<AllDateTime>No</AllDateTime>* *<StartDateTime>2000-01-01-T00:00M</StartDateTime>* *<EndDateTime>2020-12-31-T23:59M</EndDateTime>* *<AllWeek>Yes</AllWeek>* *<StartWeek>Sun</StartWeek>* *<EndWeek>Sun</EndWeek>* <Department>*1*</Department> <Access>*Yes*</Access> </List> <List ID=”2”> *<Use>Yes</Use>* *<AllDateTime>No</AllDateTime>* *<StartDateTime>2000-01-01-T00:00M</StartDateTime>*	<Request>: SetAccessList, Set Access Control Rules List. <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.* *<Action>: Add/Edit/Delete, Add – add rule, Edit – edit rule, Delete – delete rule.* <AccessListID>*: when Delete, delete the rule directly.* <AccessList *Count="2">:* AccessList – access control rule list, *Count – rules count.* <List ID=”1”>*: rule Number.* *<Use>: Yes/No，Enable the Rule or not?* *<AllDateTime>:Yes/No，Yes – everyday, No – use the date & time range below* *<StartDateTime>：Start Date & Time(hour:minute) of the Rule.* *<EndDateTime>：End Date & Time(hour:minute) of the Rule.* *<AllWeek>:Yes/No，Yes – every day，No – use the day range below. (Start from which day of the week. End at which day of the week.)* *<StartWeek>: Start Day of the Rule (Monday – Sunday)* *<EndWeek>: End Day of the Rule (Monday - Sunday)* <Department>*: Department Number.* <Access>*： Yes/No, Yes - can open the door.*

<EndDateTime>2020-12-31-T23:59M</EndDateTime> <*AllWeek>Yes</AllWeek>* <*StartWeek>Sun</StartWeek>* <*EndWeek>Sun</EndWeek>* <Department>*2*</Department> <Access>*Yes*</Access> </List> </AccessList> *</Message>*	
Device Response	
<?xml version="1.0"?> *<Message>* <Actid>2</Actid> <TerminalType>PFS100</TerminalType> <HardwareVer>4900</HardwareVer> <TerminalID>1</TerminalID> <DeviceSerialNo>wb2018042801</DeviceSerialNo> <Version>ZD4900 v2.0.180308</Version> <Action_ext>show/xx</Action_ext> *<Response>*SetAccessList*</Response>* *<Result>OK/Fail</Result>* *</Message>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <Result>: OK/Fail, result.

69) GetStoreStatus: Get the Device Storage Status

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>GetStoreStatus</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *</Message>*	<Request>: *GetStoreStatus*, Get the Device Storage Status. <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.*
Device Response	
<?xml version="1.0"?> <Message> <Actid>2</Actid> <TerminalType>PFS100</TerminalType> <HardwareVer>4900</HardwareVer> <TerminalID>1</TerminalID> <DeviceSerialNo>wb2018042801</DeviceSerialNo> <Version>ZD4900 v2.0.180308</Version> <Action_ext>show/xx</Action_ext> <Response>*GetStoreStatus*</Response> *<ManagerCount>xxx-yyy</ManagerCount>* *<UserCount>xxx-yyy</UserCount>* *<FaceCount>xx-yyy</FaceCount>* *<FpCount>xxx-yyy</FpCount>* *<CardCount>xxx-yyy</CardCount>*	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <ManagerCount>: xxx – already enrolled manager count; yyy – total manager capacity. <UserCount>: xxx – already enrolled user; yyy – total user capacity. <FaceCount>: xxx – already enrolled Face; yyy – total face capacity. <FpCount>: xxx – already enrolled fingerprint; yyy – total fingerprint capacity. <CardCount>: xxx – already enrolled card; yyy – total card capacity.

<PwdCount>xxx-yyy</PwdCount> *<TimeLogCount>zzz-xxx-yyy</TimeLogCount>* *<ManageLogCount>zzz-xxx-yyy</ManageLogCount>* *<PhotoLogCount>zzz-xxx-yyy</PhotoLogCount>* <Result>OK</Result> </Message>	<PwdCount>: xxx – already enrolled password; yyy – total password capacity. <TimeLogCount>: zzz – attendance log count that Unsent to Server, xxx – already used attendance log capacity, yyy – total attendance log capacity. <ManageLogCount>: zzz – admin log count that Unsent to Server, xxx – already used admin log capacity, yyy – total admin log capacity. <PhotoLogCount>: zzz – photo log count that Unsent to Server, xxx – already used photo log capacity, yyy – total photo log capacity. <Result>: OK/Fail, result.

70) GetDeviceInfo: Get Device Information

Server Request	
<?xml version="1.0"?> *<Message>* *<Request>*GetDeviceInfo*</Request>* *<Ccid>2</Ccid>* *<Time>2013-4-11-T11:28:54Z</Time>* *<Action_ext>show/xx<Action_ext>* *</Message>*	<Request>: GetDeviceInfo, get device information. <Ccid>: Session sequence number, long type, increment by increase 1. <Time>: Server Time, Device receive this Time, Synchronize to the Device. <Action_ext>:*action mark, Cloud Server could define this value, for example, Cloud Server give instruction that this is to show, the value is show, then, the Device returns the value as show, Cloud Server can judge this value to check what this data is used for.*
Device Response	

<?xml version="1.0"?> <Message> <Actid>2</Actid> <TerminalType>PFS100</TerminalType> <HardwareVer>4900</HardwareVer> <TerminalID>1</TerminalID> <DeviceSerialNo>wb2018042801</DeviceSerialNo> <Version>ZD4900 v2.0.180308</Version> <Action_ext>show/xx</Action_ext> <Response>GetDeviceInfo</Response> <ReleasedTime>*2010-4-11*</ReleasedTime> *<ProductName>AK47</ProductName>* <Manufacturer>Russia</Manufacturer> <Result>OK</Result> </Message>	<Actid>: <Actid> is <Ccid> <TerminalType>: Terminal Type, use this to judge if the Device support the functions in API or not. <HardwareVer>: Hardware Platform <TerminalID>: Device ID <DeviceSerialNo>: Device unique serial number <Version>: Firmware Version <Action_ext>: respond the requested value directly. <Response>: respond the requested value directly. <ReleasedTime>: Device manufactured Date <*ProductName*>: Device Model. <Manufacturer>: Device manufacturer. <Result>: OK/Fail, result.