

机器人

身份认证

使用野狗自定义身份认证

<https://docs.wilddog.com/auth/Web/guide/custom.html>

- 1.安装 Wilddog Auth SDK。
- 2.创建 Wilddog Auth 实例。
- 3.当用户使用api成功登录用户系统后，获取Custom Token。
- 4.客户端收到 Custom Token 后，使用认证方法获取uid。

数据结构

```
| -robots
  | -robotUid1                                //认证过后的机器人uid,客户端收到 CustomToken后，使用
  认证方法进行获取
    | -name:<小火宝>                          //机器人名字，字符串
    | -status:<online/offline>                 //当前离在线状态，字符串
    | -type:<desktop/ground>                   //机器人类型,字符串
    | -electricity:<0~1>                       //当前电量，浮点型
    | -currentUser:<uid>                      //当前正在与机器人交互的用户，字符串
    | -monitor:<true/false>                   //机器人监控状态，bool值true可被监控，false不行
    | -control
      | -user1:<command>                      //控制指令，字符串，见下面说明，user1表示用户id，移动
      控制指令机器人只监听currentUser发出的，风扇控制监听所有用户的
      | -fanState:<high/low/off>              //风扇状态，字符串
      | -airBox
        | -co2:<0~500>                        //一氧化碳含量，整型
        | -ch2o:<0~500>                      //甲醛浓度，整型
        | -temperature:<0~60>                //温度，整型
        | -humidity:<20~90>                  //湿度，整型
        | -pm25:<10>                         //pm2.5指数，整型
        | -gas: <true/false>                //天然气是否超标,bool值，true表示正常,false超标
```

控制指令

```
//移动控制指令,移动端长按时间间隔700ms发送一次,
```

```
move_forward
```

```
move_back
```

```
move_left
```

```
move_right
```

```
move_stop
```

```
//风扇控制指令
```

```
fan_changeState_low
```

```
fan_changeState_high
```

```
fan_changeState_off
```

数据安全性规则

```
{
  "rules":{
    "robots":
      {
        "$robotUid":
          {
            "name":
              {
                ".read": "true"
                ".write": "true"
              }
            "status":
              {
                ".read": "true"
                ".write": "auth.uid==$robotUid"
              }
            "electricity":
              {
                ".read": "true"
                ".write": "auth.uid==$robotUid"
              }
            "currentUser":
              {
                ".read":"true",
                ".write":"auth.uid==$robotUid"
              }
            "monitor":
              {
                ".read":"true",
                ".write":"auth.uid==$robotUid"
              }
          }
      }
  }
}
```

```

        "control":
        {
            "$userId":{
                ".read":"auth.uid==$robotUid"
                ".write":"true"
            }
        }
        "fanState":
        {
            ".read":"true",
            ".write":"auth.uid==$robotUid"
        }
        "airBox":
        {
            ".read":"true",
            ".write":"auth.uid==$robotUid"
        }
    }
}
}
}
}
}

```

机器人端在线状态的设置代码参考：

```

// 记录客户端在线状态的节点
var onlineRef = wilddog.sync().ref("/robots");

// 客户端在线，将状态更改为 online（具体逻辑省略）
onlineRef.child(uid).child('status').set("online");

// 注册离线事件：客户端离线时，将状态更改为 offline（具体逻辑省略）
onlineRef.child(uid).child('status').onDisconnect().set("offline");
onlineRef.child(uid).child('status').on("value", function(snap) {
    if (snap.val() === "online") {
        console.log("current user online");
    } else if (snap.val() === "offline") {
        console.log("current user offline");
        // 一旦监听到当前用户是离线状态，但实际上是在线状态，则修正用户的在线状态
        onlineRef.child(uid).child('status').set("online");
    } else {
        console.log("other status");
    }
});

```

