

SubDevice.SendEx

Метод можно использовать только с драйверами [AV & Custom Systems](#) (HTTP). Позволяет отправить оборудованию массив данных с указанием на родительский объект (контекст исполнения). Внутри SendEx можно добавить ряд событий, передающих в функции данные о своей работе. События позволяют следить за получением данных от HTTP сервера и обрабатывать различные этапы работы с сервером.

Синтаксис

SubDevice.SendEx(Data, Object)

Название	Пример	Описание
Data	[a, b, c]	type: array массив данных для отправки
Object	this	type: Object объект родитель

На выходе

-

Пример

```
var l_oDevice = module.GetDevice("Server REST"); //Получаем устройство
var l_oSubDevice = module.AddSubDevice("SubDevice 1", l_oDevice); //создаем
подустройство
l_oSubDevice.SendEx({
    Type: "POST",
    Url: "/html/login",
    Data: ["Password=2007&name=authform&Login=admin"],
    Headers: {"Content-Type": "application/x-www-form-urlencoded"},
    cbReceiveText: function(text, code, headers)
{IR.Log("cbReceiveText "+text+code+headers);},
    cbReceiveData: function(text, code, headers)
{IR.Log("cbReceiveData "+text+code+headers);},
    cbReceiveCode: function(code)
{IR.Log("cbReceiveCode "+code);},
    cbReceiveKey: function(key,value)
{IR.Log("cbReceiveKey "+key+value);},
    cbReceiveStartBody: function(stream)
{IR.Log("cbReceiveStartBody "+stream);},
    cbReceivePartBody: function(stream)
{IR.Log("cbReceivePartBody "+stream);},
    cbReceiveEndBody: function(size)
{IR.Log("cbReceiveEndBody "+size);},
    cbTimeOut: function()
{IR.Log("cbTimeOut");},
    cbReceiveStream: function(stream, code, headers)
{IR.Log("cbReceiveStream "+stream+code+headers);}
```

```
});  
l_oSubDevice.SendEx({  
    Type: "GET",  
    Url:  "/json/ok/server/channel",  
    Headers: {"Cookie":"ir-session-id=o8lKdEvw2ZlU"},  
    cbReceiveText:  function(text, code, headers)  
{IR.Log("cbReceiveText "+text);} }  
});
```