

回给调用者。

业务系统在请求 IotHub 给指定设备下发指令时，需要提供设备的 ProductName、DeviceName、指令的名称、指令数据以及指令的 TTL。如果指令数据为二进制数据，那么业务系统需要在请求前将指令数据使用 Base64 进行编码，并在请求时指明编码格式 (Encoding) 为 Base64。

```
1. router.post("/:productName/:deviceName/command", function (req, res) {
2.   var productName = req.params.productName
3.   var deviceName = req.params.deviceName
4.   Device.findOne({"product_name": productName, "device_name": deviceName},
      function (err, device) {
5.     if (err) {
6.       res.send(err)
7.     } else if (device != null) {
8.       var requestId = device.sendCommand({
9.         commandName: req.body.command,
10.        data: req.body.data,
11.        encoding: req.body.encoding || "plain",
12.        ttl: req.body.ttl != null ? parseInt(req.body.ttl) : null
13.      })
14.      res.status(200).json({request_id: requestId})
15.    } else {
16.      res.status(404).send("device not found")
17.    }
18.  })
19. })
```

指令发送成功以后，IotHub 会把指令的 RequestID 返回给业务系统，业务系统应该保存这个 RequestID，以便在收到设备对指令的回复时进行匹配。

例如，用户可以远程让家里的路由器下载一个文件，并希望下载完成后在手机上能收到通知，那么业务系统在调用 IotHub Server API 下发指令到路由器后应该保存 RequestID 和用户 ID，路由器下载后便回复该指令，业务系统收到后用回复里的 RequestID 去匹配它保存的 RequestID 和用户 ID，如果匹配成功，则使用对应的用户 ID 通知用户。

因为我们需要用 command 参数拼接主题名，所以 command 参数不应该包含有“#”“/”“+”以及 IotHub 预留的一些字符，这里为了演示，跳过了输入参数的校验，但是在实际项目中是需要加上的。

9.3.4 服务器订阅

在 DeviceSDK 里面没有任何订阅接收指令的主题代码，因为我们会使用 EMQ X 的服务器订阅来实现设备的自动订阅。

EMQ X 的服务器订阅是在“<EMQ X 安装目录>/emqx/etc/emqx.conf”里进行配置的，下面我把这些配置项讲解一下。