**TF6760 TwinCAT 3 IoT HTTPS/REST使用**

|  |  |
| --- | --- |
|  | 作者：徐良定  职务：华东区 技术工程师  邮箱：ld.xu@beckhoff.com.cn  日期：2025-07-10 |
| **摘 要：**  REST（Representational State Transfer）API在IoT通信中经常由Web Server端提供，通过统一的非监控模式接口进行通信。例如，REST API可用于查询天气数据，与消息服务端进行通信，或作为调用数据库的接口。  TF6760 [TwinCAT](https://www.beckhoff.com.cn/zh-cn/products/automation/twincat/" \l "text_bild_2" \o "Open, PC-based control technology) 3 IoT HTTPS/REST 功能组件为用户提供了在PLC中进行HTTP/HTTPS 通信的基本功能，让用户能够以客户端的形式访问REST API，并提供GET、PUT和POST等 HTTP命令。通信通道（HTTPS）由SSL/TLS提供安全保障机制。可以在PLC中通过XML和 JSON格式来定义通信时的消息内容。  此文档模拟了实际生产环境下的数据追溯系统，PLC可能涉及到的直接通过访问Web API接口，实现对生产数据的增、删、查、改等功能，并提供了简单的HMI界面让用户能够直接看到仿真效果。 | |
| **附 件：**   |  |  |  | | --- | --- | --- | | 序 号 | 文件名 | 备注 | |  |  |  | |  |  |  | |  |  |  | |  |  |  | | |
| **历史版本：**   |  |  |  | | --- | --- | --- | |  |  |  | |  |  |  | |  |  |  | |  |  |  | | |
| **免责声明：**  我们已对本文档描述的内容做测试。但是差错在所难免，无法保证绝对正确并完全满足您的使用需求。本文档的内容可能随时更新，如有改动，恕不事先通知，也欢迎您提出改进建议。 | |
| **参考信息：** | |

目 录

[1. 软硬件版本 3](#_Toc204171433)

[1.1. 倍福Beckhoff 3](#_Toc204171434)

[1.1.1. 控制器硬件 3](#_Toc204171435)

[1.1.2. 控制软件 3](#_Toc204171436)

[1.2. 其它 3](#_Toc204171437)

[1.2.1. 开发软件 3](#_Toc204171438)

[2. Web API项目实现 3](#_Toc204171439)

[2.1. 创建ASP.NET Core Web API项目 3](#_Toc204171440)

[2.2创建Product实体类 4](#_Toc204171441)

[2.3 添加ProductController 5](#_Toc204171442)

[2.4在VS2022里面运行源代码或者直接运行发布后的文件，启动WEB API服务 5](#_Toc204171443)

[2.4确认WEB API服务已启动，记录端口号信息 6](#_Toc204171444)

[3. PLC程序编写 6](#_Toc204171445)

[3.1. FB\_IotHttpClient参数配置 7](#_Toc204171446)

[3.2. Post请求添加Product 7](#_Toc204171447)

[3.3 Delete请求删除指定ID的Product 9](#_Toc204171448)

[3.4 Get请求获取分页查询的产品数据 10](#_Toc204171449)

[3.5 Get请求获取指定BarCode产品的信息 12](#_Toc204171450)

[4. 常见问题 13](#_Toc204171451)

[4.1. PLC程序里面Http请求传递参数的方式如何选择？ 13](#_Toc204171452)

[4.2. 通过 URI 和 Body 传递数据的区别 14](#_Toc204171453)

# 软硬件版本

## 倍福Beckhoff

### 控制器硬件

无

### 控制软件

笔记本Win11

基于TwinCAT 3.1 Build 4024.64版本

TF6760 试用授权

## 其它

文件说明：

* 1. Publish.7z：WEB API可执行程序，解压后可以直接运行里面的exe文件，启动WEB API服务。

文本

AI 生成的内容可能不正确。

* 1. TF6760ApiTest.7z：WEB API C#源代码。

文本

AI 生成的内容可能不正确。

* 1. TF6760PLCSample.7z：TwinCAT测试程序。

图片包含 文本

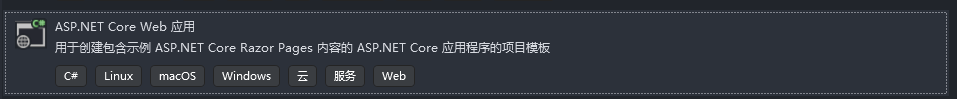
AI 生成的内容可能不正确。

### 开发软件

VS2022 Community

# Web API项目实现

## 创建ASP.NET Core Web API项目



屏幕的截图

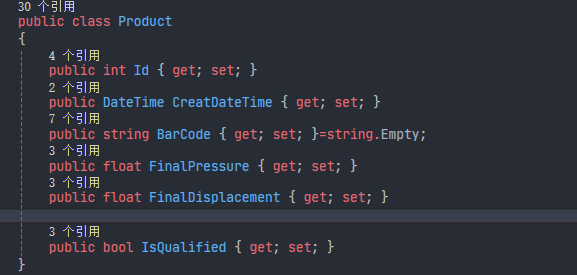
AI 生成的内容可能不正确。

此处选择了.NET 8.0的框架版本。

ASP.NET Core开发教程参考微软官方教程。

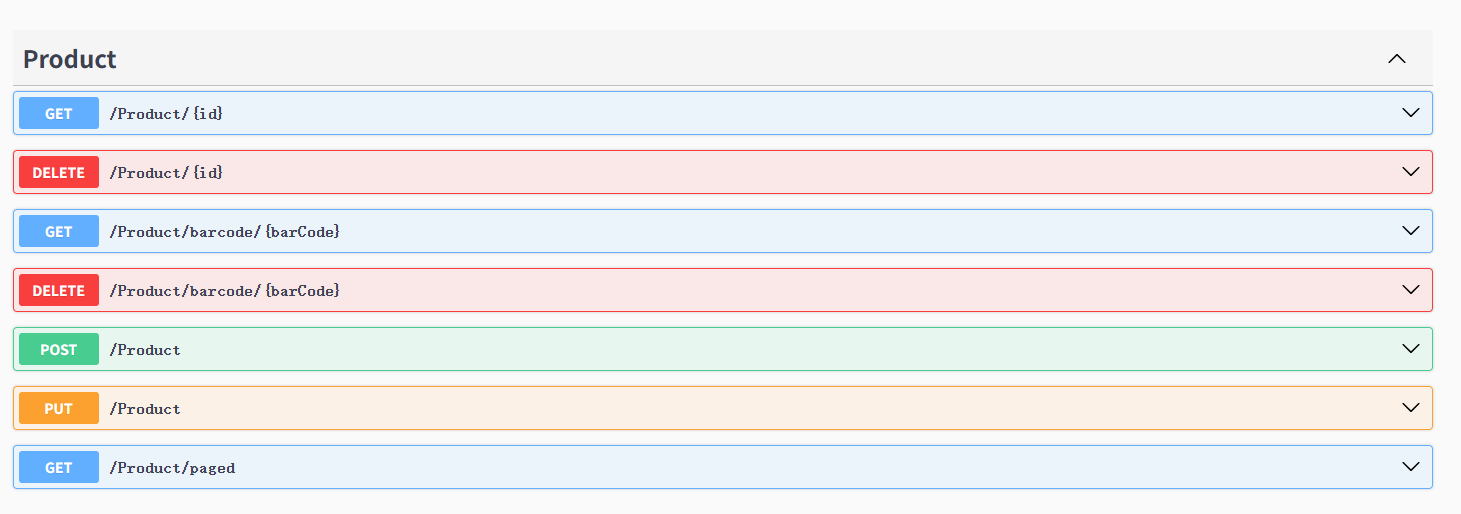
[使用 ASP.NET Core 控制器创建 Web API - Training | Microsoft Learn](https://learn.microsoft.com/zh-cn/training/modules/build-web-api-aspnet-core/)

## 2.2创建Product实体类



实体类里面包含了一些模拟实际产品的属性。

## 2.3 添加ProductController



API接口描述信息如上，主要包含了添加新产品数据、根据条件删除对应产品、分页查询数据库里面的产品数据、更新某个产品数据等功能。具体实现代码源代码以及可运行程序在附件里面。

数据库里面默认生成了100个随机的Product数据，用于客户进行测试。

## 2.4在VS2022里面运行源代码或者直接运行发布后的文件，启动WEB API服务

1. 在VS2022里面启动源代码进行调试；

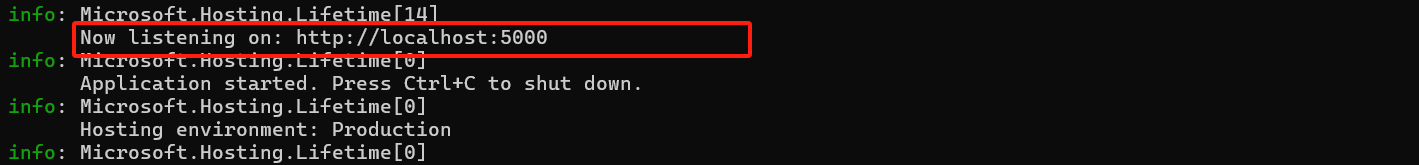
电脑萤幕画面

AI 生成的内容可能不正确。

1. 在发布的Publish文件夹里面，启动可执行文件；



## 2.4确认WEB API服务已启动，记录端口号信息



此处HTTP端口号是5000。

# PLC程序编写

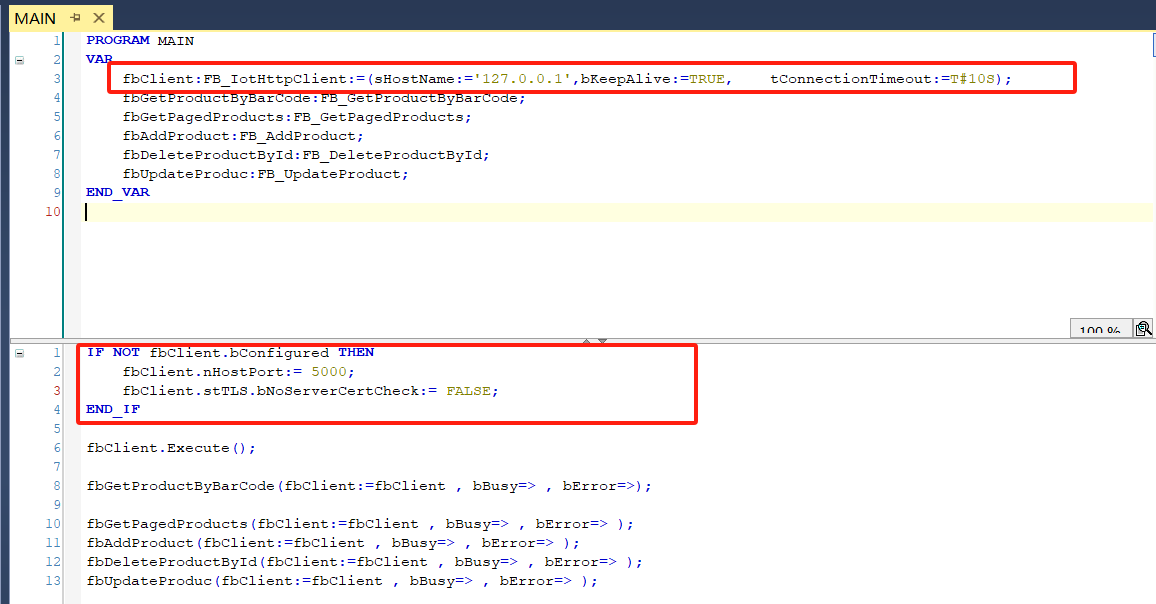


图形用户界面, 文本, 应用程序, 电子邮件

AI 生成的内容可能不正确。

PLC程序里面主要使用到了FB\_IotHttpClient和FB\_IotHttpRequest功能块。前者用于配置连接参数，建立连接。后者用来发送HTTP请求。此外还用到了Tc3\_JsonXml库，用来解析WEB API返回的Json数据和组合Json对象发送给WEB API。

## FB\_IotHttpClient参数配置



对于测试环境，由于WEB API和TwinCAT程序在同一台电脑上面，所以sHostName填写‘127.0.0.1’，nHostPort填写上面WEB API服务里面显示的端口号5000。

FB\_IotHttpClient的stTLS输入为ST\_IotSocketTls类型，里面bNoServerCertCheck这个值表示禁用服务器证书有效性验证。如果通信不使用 TLS 加密（HTTP/WebSockets），该值必须保持为 FALSE。

上面演示的WEB API后端项目里面没有TLS功能，所以这个值设置为FALSE。

## Post请求添加Product

图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。

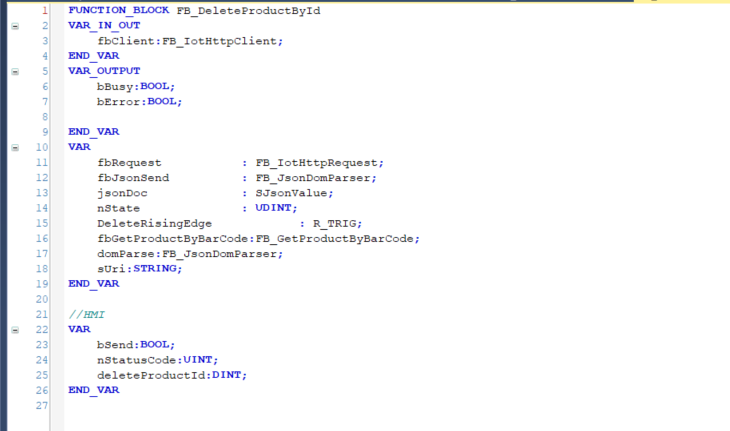


WEB API后端增加Product的逻辑里面，会判断BarCode是否重复。当BarCode不重复时，返回状态码201，表示添加成功。当BarCode和数据库里面数据重复时，拒绝添加Product，返回状态码409。状态码 **409** 表示 **Conflict（冲突）。**

HMI演示效果如下：



## Delete请求删除指定ID的Product



图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。

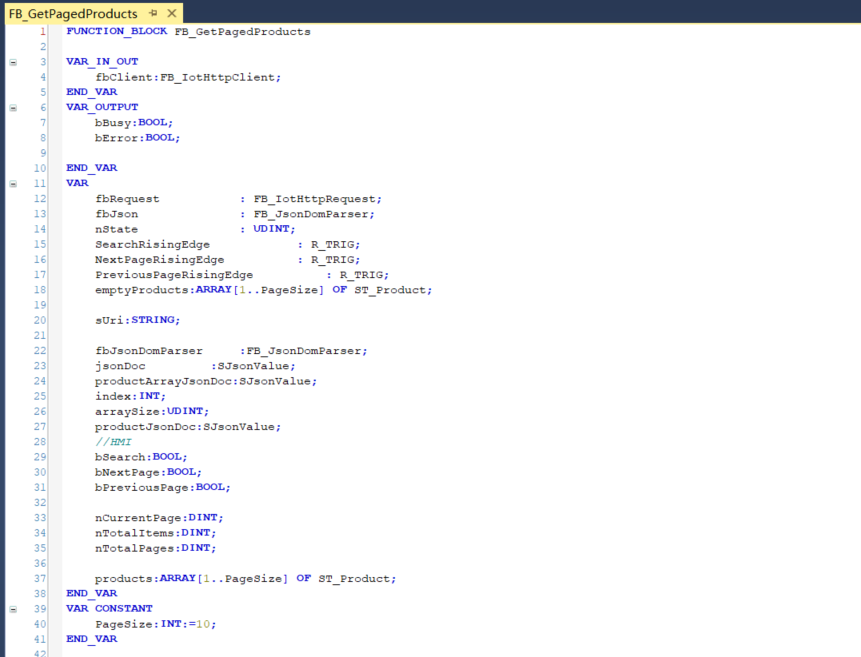
WEB API后端删除Product的逻辑里面，会判断当前要删除的对应ID的产品是否存在。当对应ID的产品存在时，返回状态码204，表示删除成功。不存在时，拒绝删除Product，返回状态码404。

HMI演示效果如下：



## Get请求获取分页查询的产品数据

从性能、网络传输、用户体验等方面考虑，通过不需要返回数据库里面的所有数据。对于UI展示页面，用户通常只需要查看部分数据，分页可以让用户更快地浏览和定位所需内容。

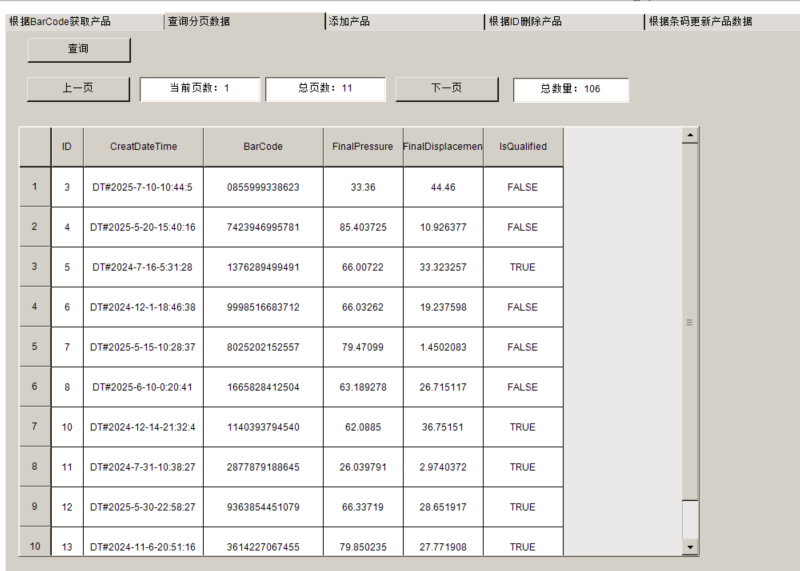


图形用户界面, 文本

AI 生成的内容可能不正确。

当点击查询按钮的时候，会把当前页数设置成1，每页的数量设置为10。WEB API会返回当前的页数、总页数、总产品数量以及当前页数的产品信息，用于让前端进行显示。

HMI演示效果如下：



## Get请求获取指定BarCode产品的信息

对于实际设备上面，通常会安装一个读码器，获取产品的条码/二维码等信息。该功能可以获取对应产品的信息，然后根据是否合格或者其它信息，设备判断继续生产或者放行等操作。

文本

AI 生成的内容可能不正确。

图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。

WEB API后端查询具有指定BarCode的Product的逻辑里面，会判断条码是否存在。条码存在的话，返回状态码200。条码不存在时，返回状态码404Not Found。

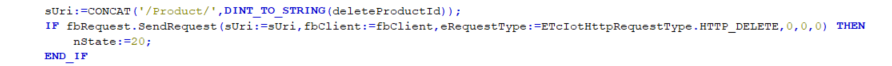
HMI显示效果如下：



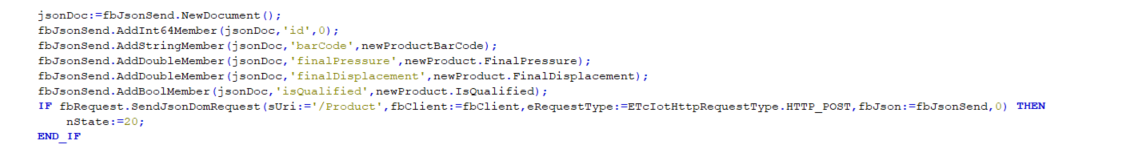
# 常见问题

## PLC程序里面Http请求传递参数的方式如何选择？

在DeleteProductById的示例中，我们使用如下在Uri里面传递了需要删除的ID参数。

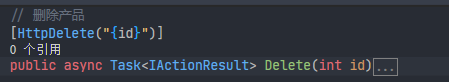


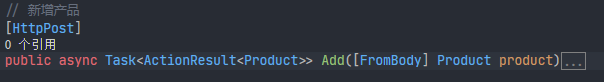
在AddProduct示例中，我们使用了如下在**请求体（Request Body）**里面传递新增的Product参数。在PLC代码里面，通过FB\_JsonDomParser来组合Json对象。



PLC程序里面使用哪种方式，**取决于WEB API（后端）项目里面如何定义这个HTTP请求**。PLC程序发送HTTP请求的格式，必须要和后端定义的一致。

后端定义如下：





## 通过 URI 和 Body 传递数据的区别

1. URI（包括路径参数和查询字符串）

* **数据位置**：数据放在请求的 URL 中，如 **/api/products/123** 或 **/api/products?name=abc**
* **常用场景**：GET、DELETE、部分 PUT 请求，适合简单、标识性的数据（如ID、筛选条件）
* **优点**：易于缓存、书签、日志记录，参数直观
* **缺点**：不适合传递复杂或大量数据，URL长度有限

1. Body（请求体）

* **数据位置**：数据放在 HTTP 请求体中，通常为 JSON、XML 或表单格式
* **常用场景**：POST、PUT、PATCH，适合传递复杂对象（如新增、更新实体）
* **优点**：可传递结构化、复杂、大量数据，无长度限制
* **缺点**：不易被缓存，参数不直观，GET 请求不能用 Body

总结：

* **URI** 适合简单、标识性参数（如ID、筛选条件）
* **Body** 适合复杂、结构化数据（如对象、列表）
* RESTful 推荐：查询/标识用 URI，新增/更新用 Body

（以上来自AI回答）

**上海（ 中国区总部）**

中国上海市静安区汶水路 299 弄 9号（市北智汇园）

电话: 021-66312666

**北京分公司**

北京市海淀区魏公村路6号院1号楼丽金智地中心西塔901室

电话: 010-82200036 邮箱: beijing@beckhoff.com.cn

**广州分公司**

广州市天河区珠江新城珠江东路32号利通广场1303室

电话: 020-38010300/1/2 邮箱: guangzhou@beckhoff.com.cn

**成都分公司**

成都市锦江区东御街18号 百扬大厦2305 室

电话: 028-86202581 邮箱: chengdu@beckhoff.com.cn

|  |  |
| --- | --- |
| 请用微信扫描二维码  通过公众号与技术支持交流 | 倍福官方网站：  https://www.beckhoff.com.cn  在线帮助系统：  https://infosys.beckhoff.com/index\_en.htm |
| 倍福虚拟学院：  https://tr.beckhoff.com.cn/ |
| 招贤纳士：job@beckhoff.com.cn  技术支持：support@beckhoff.com.cn  产品维修：service@beckhoff.com.cn  方案咨询：sales@beckhoff.com.cn |
|  |