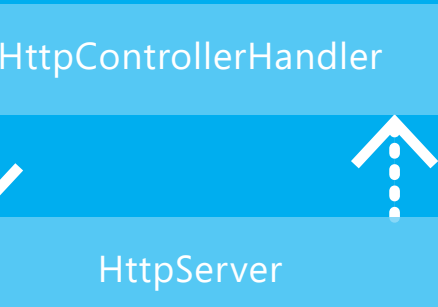


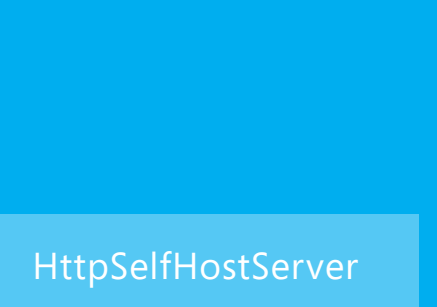
# ASP.NET WEB API: HTTP 消息生命周期

您可以在 ASP.NET 应用程序内承载 Web API，也可以在您自己的过程内承载 Web API（自承载）。在初始入口点后，HTTP 消息将通过相同管道。HTTP 请求消息首先转换为 `HttpRequestMessage` 对象，该对象提供了到 HTTP 消息的强类型访问。

## ASP.NET 承载



## 自承载



ASP.NET Web API 是一个框架，通过这个框架，可以非常轻松地构建可覆盖广泛客户端（包括浏览器和移动设备）的 HTTP 服务。它是在 .NET Framework 上构建支持 REST 的应用程序的理想平台。

本海报展示 HTTP 请求如何通过 Web API 管道传递，以及 HTTP 响应如何传递回来。此图表还展示了可扩展点，您可以在可扩展点添加自定义代码，甚至完全替换默认行为。您可以在 <http://www.asp.net/web-api> 上找到有关 ASP.NET Web API 的文档和教程。

## HTTP 消息处理程序

HTTP 消息处理程序是处理管道的第一个阶段。它们处理传入的 HTTP 请求消息以及传出的 HTTP 响应消息。若要创建自定义消息处理程序，请从 `DelegatingHandler` 类派生。可以添加多个消息处理程序。

消息处理程序可以是全局性的，也可以分配给特定路由。按路由的消息处理程序只有在请求与该路由匹配时才会调用。按路由的消息处理程序在路由表中配置。

### DelegatingHandler

消息处理程序可以跳过管道的其余部分，直接创建响应。

### HttpRoutingDispatcher



否

是

## 按路由的消息处理程序

### Route.Handler

### DelegatingHandler

### HttpMessageHandler

此消息处理程序可以调用 `ApiControllerDispatcher`，并返回到“主”路径，或提供自定义终点。

## 控制器

控制器用来定义在处理 HTTP 请求时要使用的主逻辑。控制器派生自 `ApiController` 类，或实现 `IHttpController` 接口。

### 选择控制器操作

如果请求未被授权，授权筛选器可以创建错误响应并跳过管道的其余部分。

### 授权筛选器

### 异常筛选器

异常！

未经处理的异常将路由到异常筛选器。

OnActionExecuting

OnActionExecuted

### 调用操作

## 控制器操作

## 模型绑定

模型绑定使用请求创建操作的参数值。调用操作时，这些值将会传递给该操作。

### HttpRequestMessage

请求消息

URI

标头

实体正文

### FormatterParameterBinding

### ModelBinderParameterBinding

### HttpParameterBinding

媒体类型格式化程序读取消息正文（如果有）。

默认模型绑定器从 URI 路径和查询字符串读取。

自定义参数绑定可以读取 HTTP 请求的任何部分。

### 媒体类型格式化程序

### IModelBinder

### IValueProvider

复杂类型

简单类型

任何类型

操作参数

## 结果转换

此操作的返回值将被转换为 `HttpResponseMessage`。

### HttpResponseMessage

### 媒体类型格式化程序

### IContentNegotiator

如果返回类型为 `HttpResponseMessage`，允许通过。

如果返回类型为 `void`，创建状态为 204（无内容）的响应。

对于所有其他返回类型，媒体类型格式化程序将值序列化并写入到消息正文。

HttpResponseMessage

void

其他类型

操作返回值

## 图例

- 内置类
- 可扩展点
- 说明
- 请求
- 响应