

.NET Conf China 2023

2023/12/16
09:30 - 18:00

中国 · 北京



中国·北京

.NET Conf China 2023

.NET下AOT项目实战指南

桂素伟
NETSTARS架构师



什么项目
适合AOT

AOT开发
注意事项

AOT发布
优化技巧

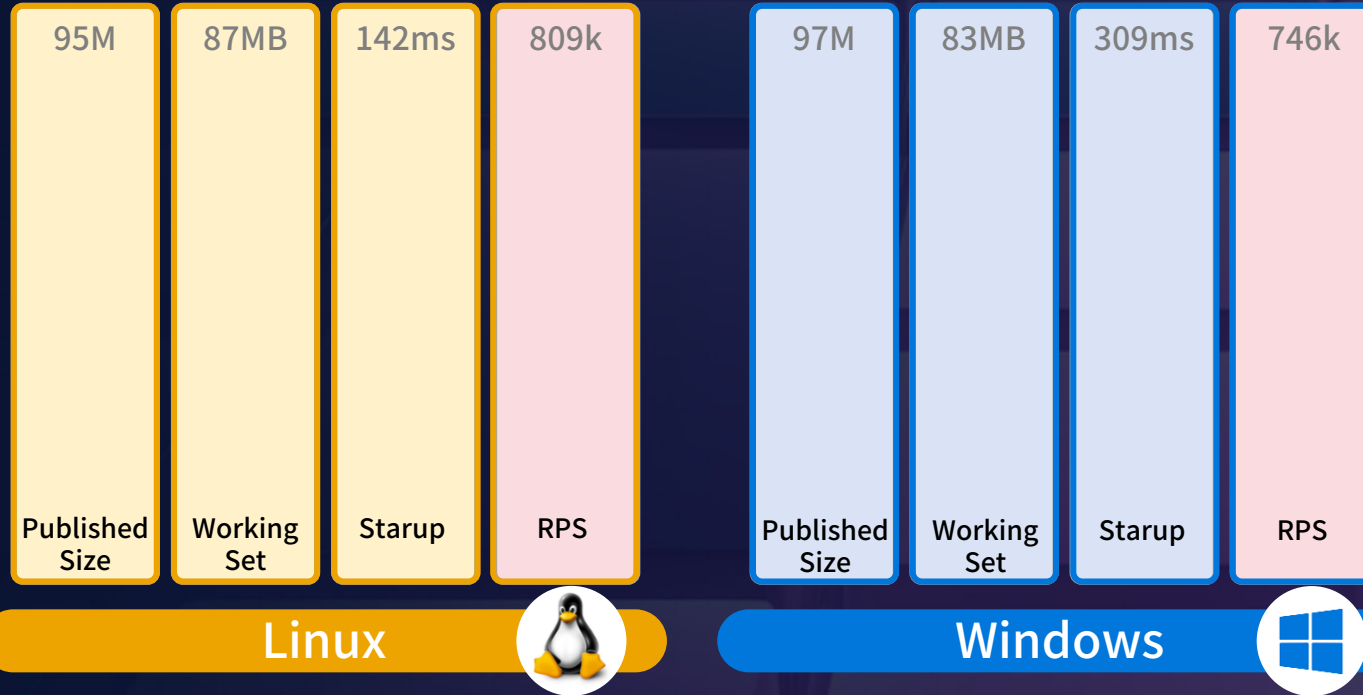


什么项目
适合AOT

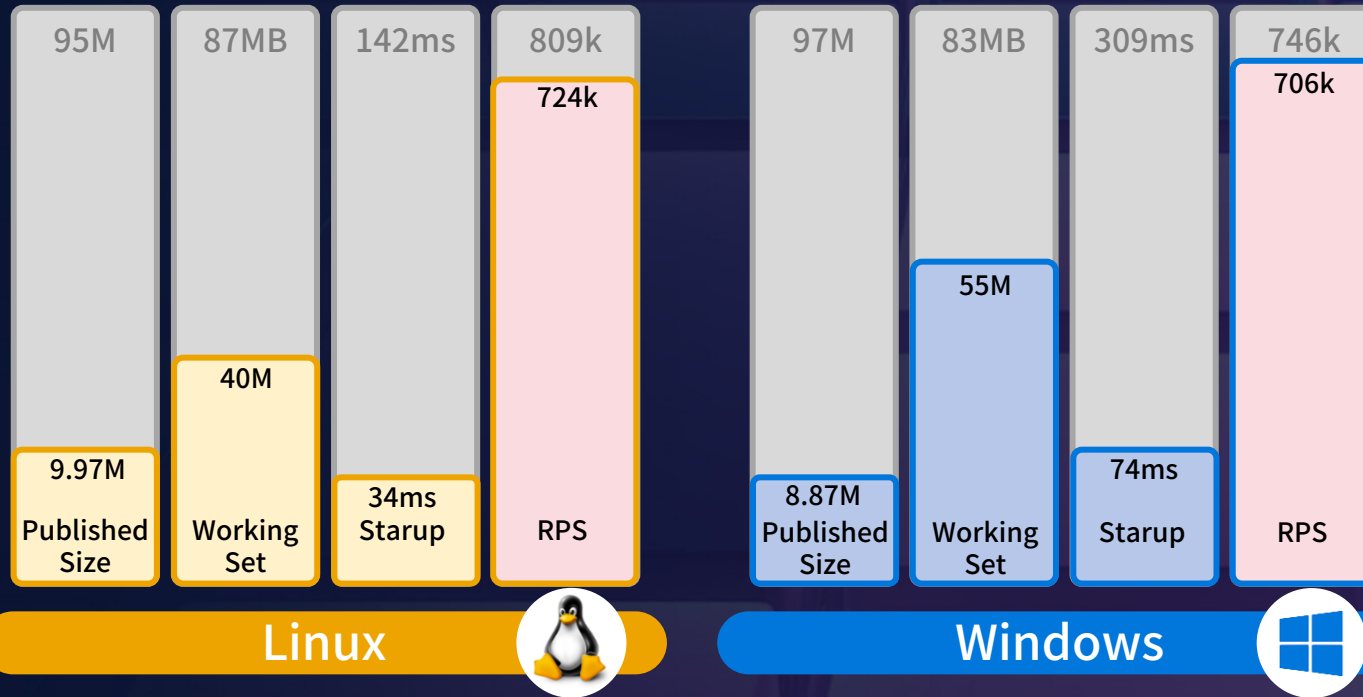




.NET Performance



.NET & Native AOT Performance



适合AOT项目

Function as a Service

复杂签名算法

高性能基础设施

API网关, 鉴权服务

单一功能业务应用

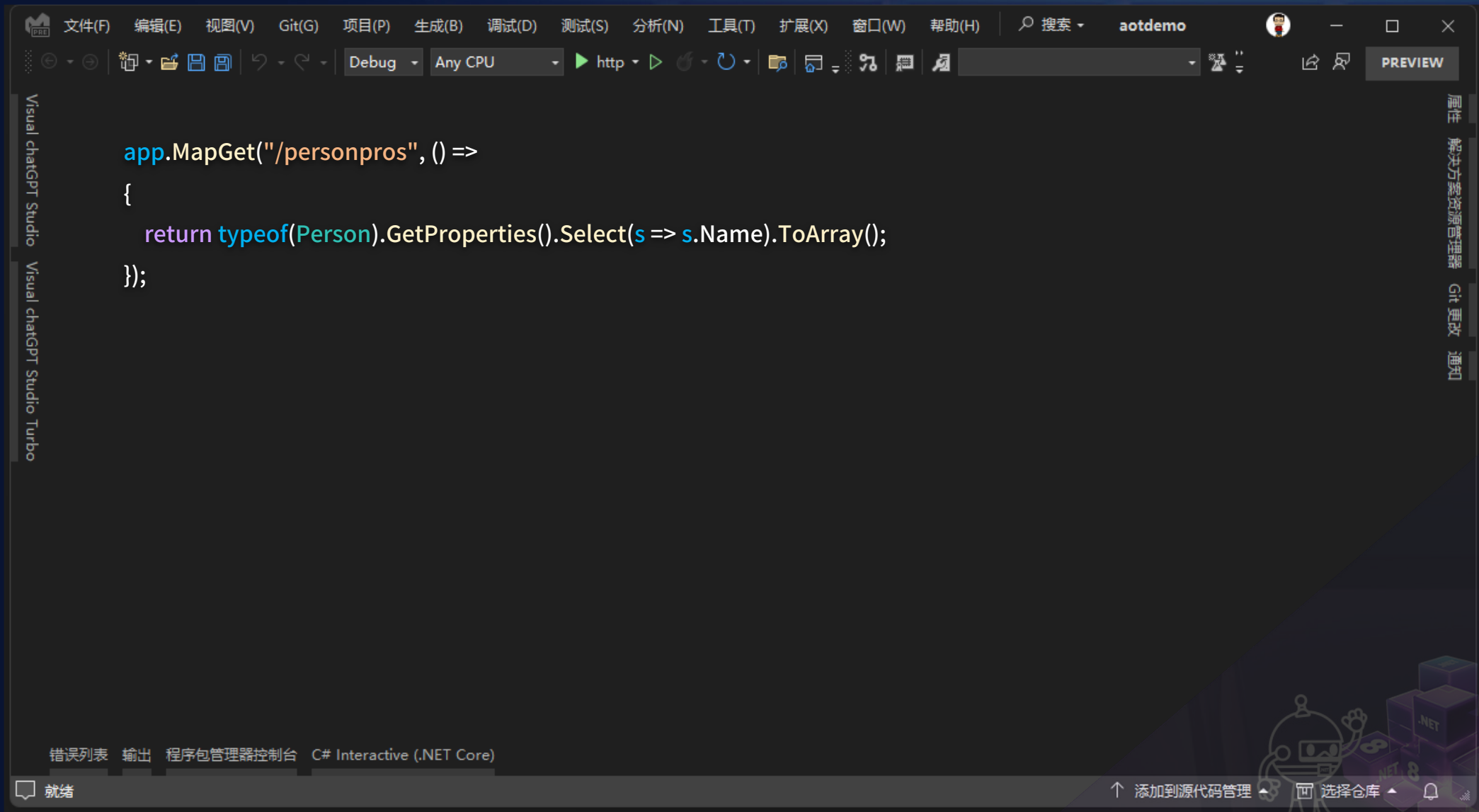
订单ID生成服务, 交易风控系统

AOT



AOT开发
注意事项

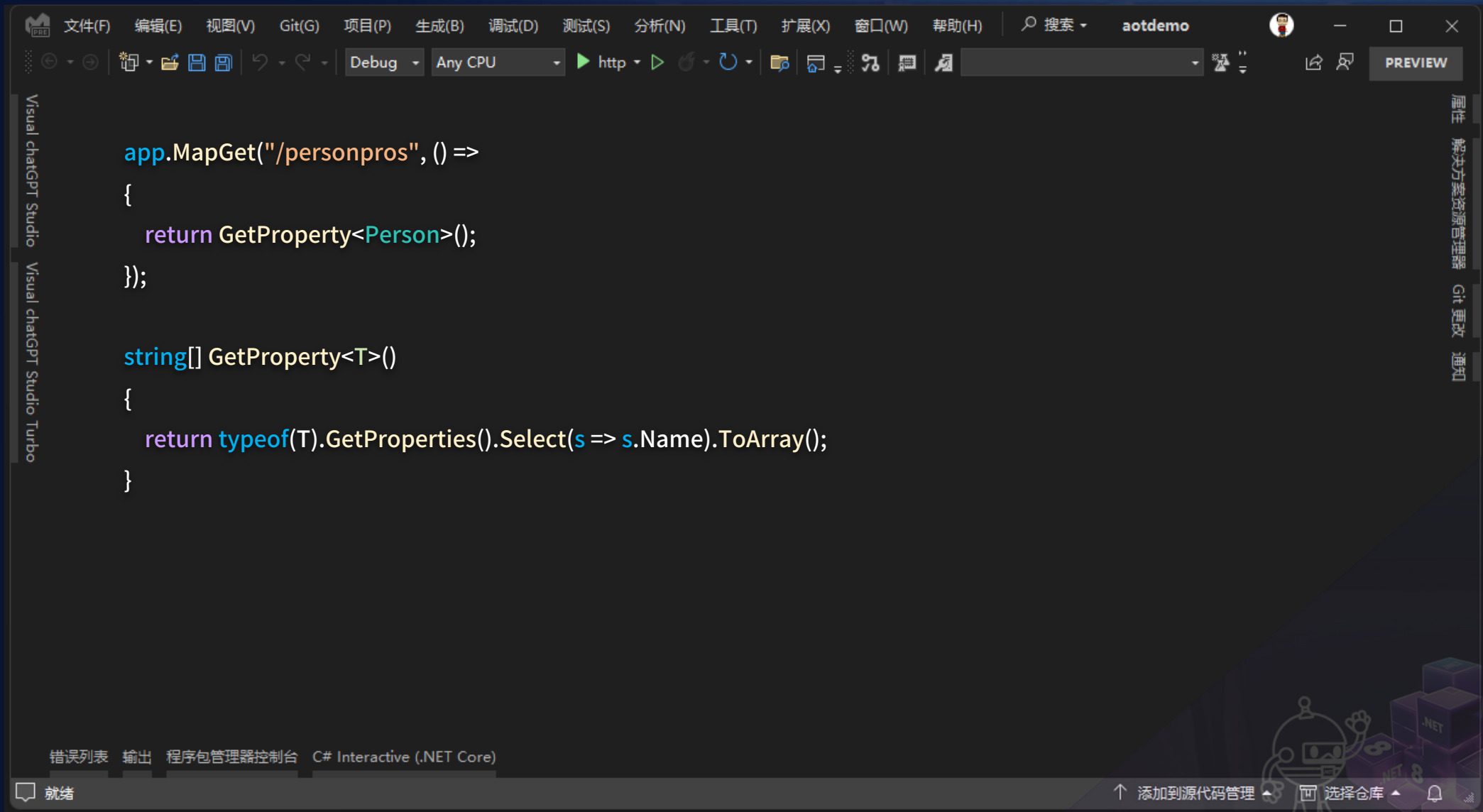




The screenshot shows the Visual Studio Code editor interface. The top menu bar includes options like 文件(F), 编辑(E), 视图(V), Git(G), 项目(P), 生成(B), 调试(D), 测试(S), 分析(N), 工具(T), 扩展(X), 窗口(W), and 帮助(H). The search bar contains the text "aotdemo". The toolbar shows various icons for file operations, debugging, and testing. The main editor area displays the following C# code:

```
app.MapGet("/personpros", () =>
{
    return typeof(Person).GetProperties().Select(s => s.Name).ToArray();
});
```

The bottom status bar shows "就绪" (Ready) on the left and "添加到源代码管理" (Add to Source Control) and "选择仓库" (Select Repository) on the right. The bottom right corner features a decorative graphic with a robot character and .NET logo blocks.



The screenshot shows the Visual Studio Code editor interface. The top menu bar includes options like 文件(F), 编辑(E), 视图(V), Git(G), 项目(P), 生成(B), 调试(D), 测试(S), 分析(N), 工具(T), 扩展(X), 窗口(W), and 帮助(H). The search bar contains the text "aotdemo". The toolbar shows the Debug mode, Any CPU architecture, and a green play button for running the application. The main editor area displays the following C# code:

```
app.MapGet("/personpros", () =>
{
    return GetProperty<Person>();
});

string[] GetProperty<T>()
{
    return typeof(T).GetProperties().Select(s => s.Name).ToArray();
}
```

The bottom status bar shows "就绪" (Ready) and "C# Interactive (.NET Core)". The right sidebar contains a "PREVIEW" button and a vertical list of icons for "属性" (Properties), "解决方案资源管理器" (Solution Explorer), "Git 更改" (Git Changes), and "通知" (Notifications).

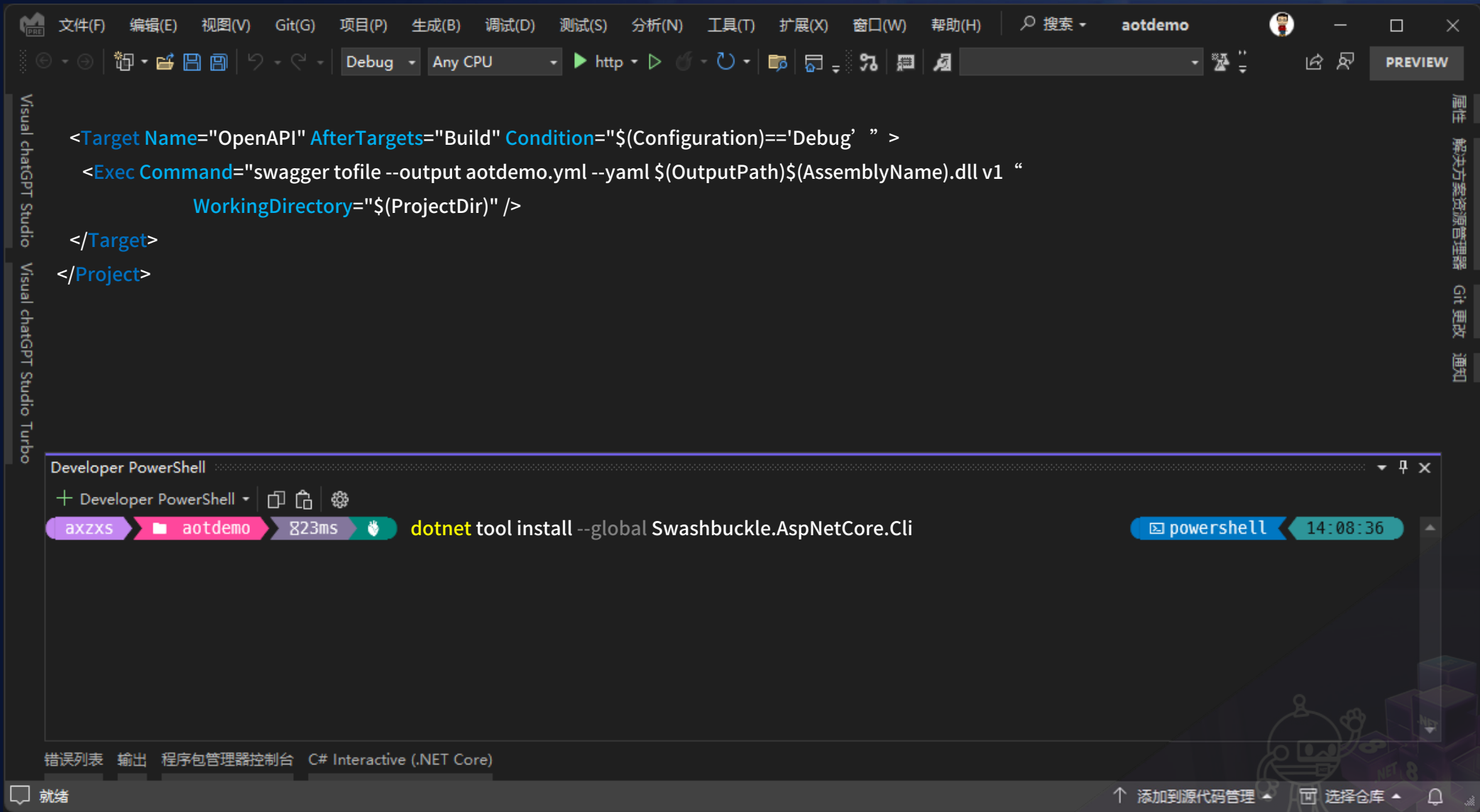




The screenshot shows the Visual Studio Code interface with a C# code file open. The code defines a `Person` class with properties `Name`, `Age`, and `Sex`, and an overridden `GetHashCode()` method. The IDE includes a menu bar, a toolbar with various icons, and a sidebar on the right with a 'PREVIEW' button. The status bar at the bottom shows '就绪' (Ready) and 'C# Interactive (.NET Core)'. In the bottom right corner, there are icons for adding to source code management and selecting a repository.

```
public class Person
{
    public string? Name { get; set; }
    public int Age { get; set; }
    public Sex Sex { get; set; }
    public override int GetHashCode()
    {
        typeof(Person).GetMembers();
        return base.GetHashCode();
    }
}
```





The image shows a screenshot of the Visual Studio Code editor interface. The main editor area displays XML code for a target named "OpenAPI". The code is as follows:

```
<Target Name="OpenAPI" AfterTargets="Build" Condition="$ (Configuration)=='Debug' " >
  <Exec Command="swagger tofile --output aotdemo.yml --yaml $(OutputPath)$(AssemblyName).dll v1 "
    WorkingDirectory="$(ProjectDir)" />
</Target>
</Project>
```

Below the code, a Developer PowerShell terminal window is open, showing the execution of the following command:

```
dotnet tool install --global Swashbuckle.AspNetCore.Cli
```

The terminal window also shows the file path "aotdemo" and the duration "823ms". The status bar at the bottom of the terminal indicates "C# Interactive (.NET Core)".





The screenshot shows the Visual Studio Code interface with a C# code file open. The code defines an enum named 'Sex' with three values: '未知', '男', and '女'. The code is wrapped in a lambda expression: `[JsonConverter(typeof(JsonStringEnumConverter))] <Sex>]]`. The interface includes a menu bar at the top with options like '文件(F)', '编辑(E)', '视图(V)', 'Git(G)', '项目(P)', '生成(B)', '调试(D)', '测试(S)', '分析(N)', '工具(T)', '扩展(X)', '窗口(W)', and '帮助(H)'. A search bar and the text 'aotdemo' are visible on the right side of the menu bar. The bottom status bar shows '就绪' (Ready) and 'C# Interactive (.NET Core)'. On the right side of the editor, there are vertical tabs for '属性', '解决方案资源管理器', 'Git 更改', and '通知'. At the bottom right, there are icons for '添加到源代码管理' and '选择仓库'.

```
[JsonConverter(typeof(JsonStringEnumConverter))] <Sex>]]  
public enum Sex  
{  
    未知,  
    男,  
    女  
}
```



AOT发布
优化技巧



```
<PropertyGroup>  
  <OptimizationPreferen  
    Size  
  </OptimizationPreferen  
</PropertyGroup>
```



模式: Size

尺寸: 33881KB

TPS: 1189.2 RT: 3 P99: 19



模式: null

尺寸: 34674 KB

TPS: 1192.9 RT: 3 P99: 19



模式: Speed

尺寸: 36918KB

TPS: 1163.9 RT: 4 P99: 20




```
FROM mcr.microsoft.com/dotnet/sdk:8.0 AS build
WORKDIR /app
COPY aotdemo.csproj .
RUN dotnet restore --runtime linux-x64 aotdemo.csproj

RUN apt-get update \
  && apt-get install -y --no-install-recommends \
  clang zlib1g-dev

COPY . .
RUN dotnet publish -c Release -r linux-x64 -o out aotdemo.csproj

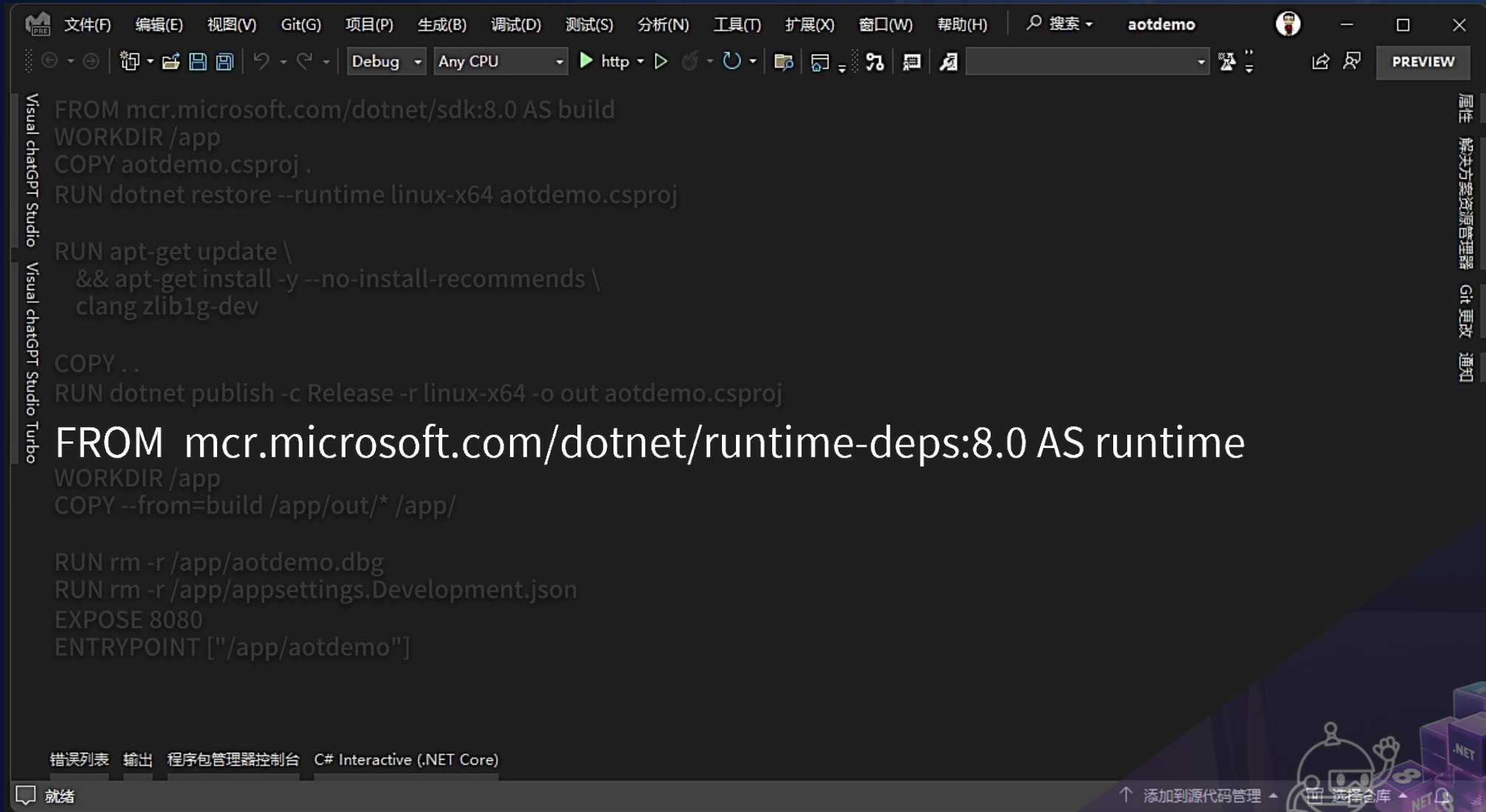
FROM mcr.microsoft.com/dotnet/runtime-deps:8.0 AS runtime
WORKDIR /app
COPY --from=build /app/out/* /app/

RUN rm -r /app/aotdemo.dbg
RUN rm -r /app/appsettings.Development.json
EXPOSE 8080
ENTRYPOINT ["/app/aotdemo"]
```

错误列表 输出 程序包管理器控制台 C# Interactive (.NET Core)

就绪





```
FROM mcr.microsoft.com/dotnet/sdk:8.0 AS build
WORKDIR /app
COPY aotdemo.csproj .
RUN dotnet restore --runtime linux-x64 aotdemo.csproj

RUN apt-get update \
  && apt-get install -y --no-install-recommends \
  clang zlib1g-dev

COPY . .
RUN dotnet publish -c Release -r linux-x64 -o out aotdemo.csproj

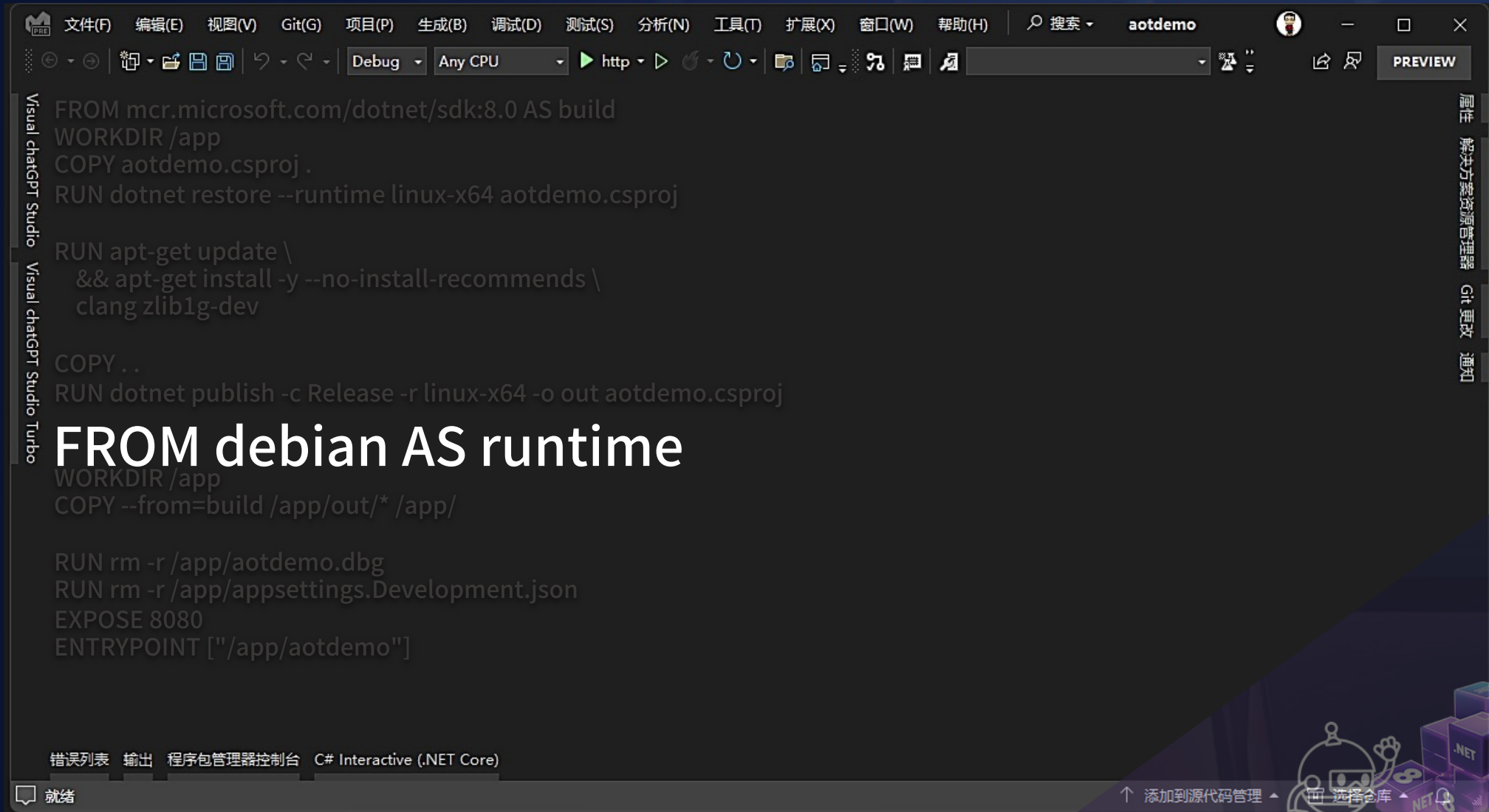
FROM mcr.microsoft.com/dotnet/runtime-deps:8.0 AS runtime
WORKDIR /app
COPY --from=build /app/out/* /app/

RUN rm -r /app/aotdemo.dbg
RUN rm -r /app/appsettings.Development.json
EXPOSE 8080
ENTRYPOINT ["/app/aotdemo"]
```

错误列表 输出 程序包管理器控制台 C# Interactive (.NET Core)

就绪





```
FROM mcr.microsoft.com/dotnet/sdk:8.0 AS build
WORKDIR /app
COPY aotdemo.csproj .
RUN dotnet restore --runtime linux-x64 aotdemo.csproj

RUN apt-get update \
  && apt-get install -y --no-install-recommends \
  clang zlib1g-dev

COPY . .
RUN dotnet publish -c Release -r linux-x64 -o out aotdemo.csproj
```

FROM debian AS runtime

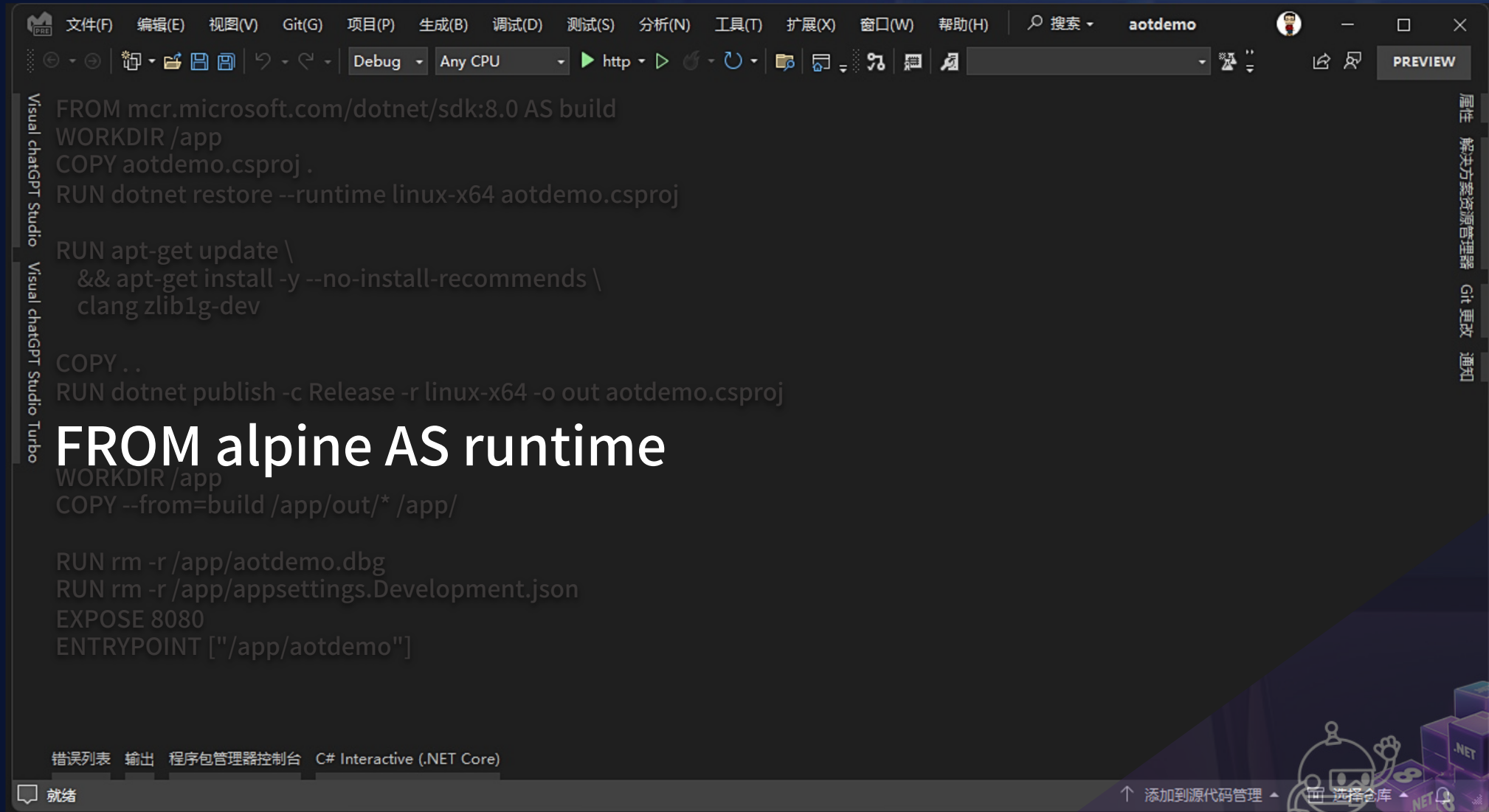
```
WORKDIR /app
COPY --from=build /app/out/* /app/

RUN rm -r /app/aotdemo.dbg
RUN rm -r /app/appsettings.Development.json
EXPOSE 8080
ENTRYPOINT ["/app/aotdemo"]
```

错误列表 输出 程序包管理器控制台 C# Interactive (.NET Core)

就绪





```
FROM mcr.microsoft.com/dotnet/sdk:8.0 AS build
WORKDIR /app
COPY aotdemo.csproj .
RUN dotnet restore --runtime linux-x64 aotdemo.csproj

RUN apt-get update \
  && apt-get install -y --no-install-recommends \
  clang zlib1g-dev

COPY . .
RUN dotnet publish -c Release -r linux-x64 -o out aotdemo.csproj

FROM alpine AS runtime
WORKDIR /app
COPY --from=build /app/out/* /app/

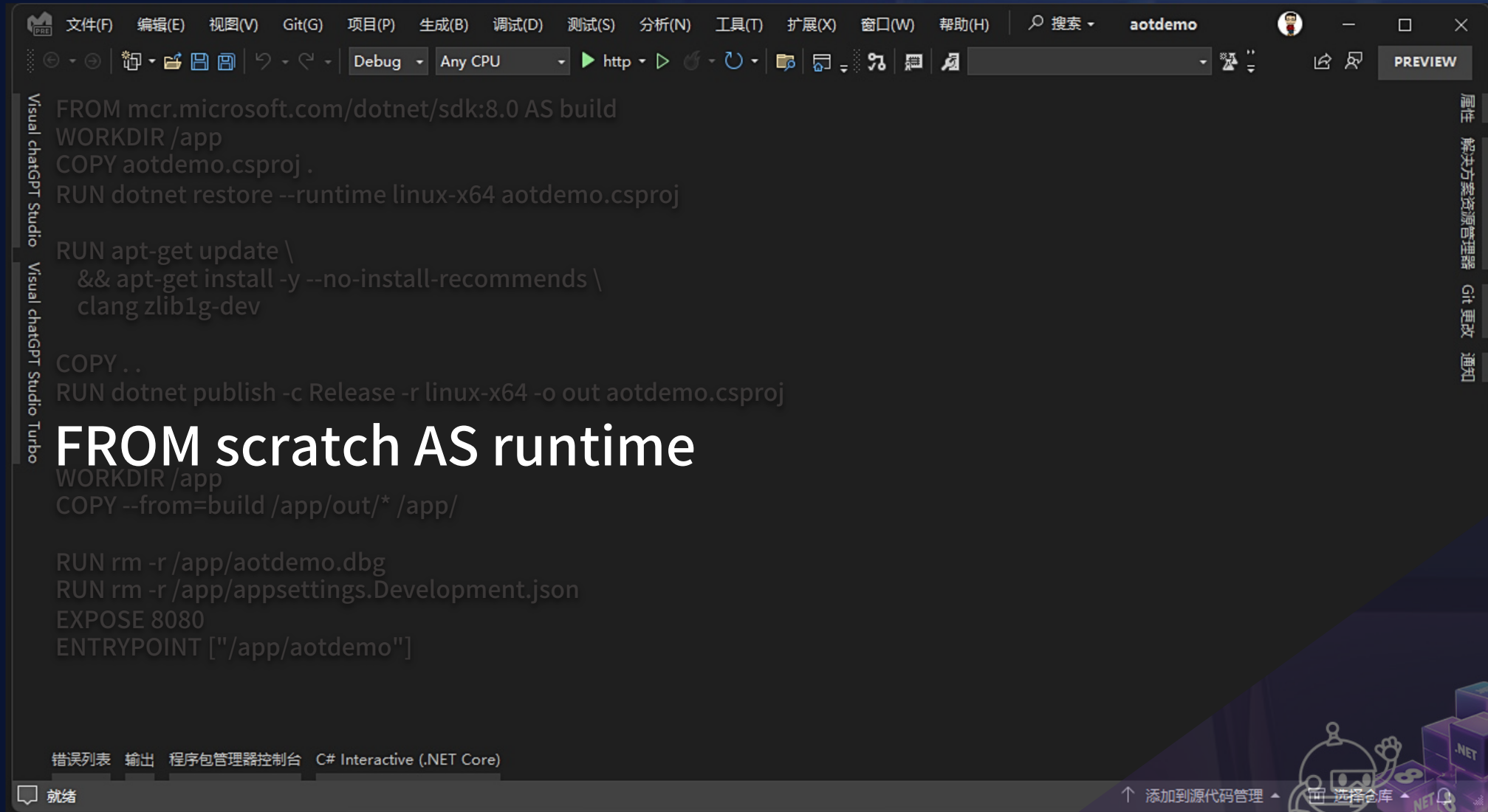
RUN rm -r /app/aotdemo.dbg
RUN rm -r /app/appsettings.Development.json
EXPOSE 8080
ENTRYPOINT ["/app/aotdemo"]
```

错误列表 输出 程序包管理器控制台 C# Interactive (.NET Core)

就绪

添加到源代码管理 选择仓库





```
FROM mcr.microsoft.com/dotnet/sdk:8.0 AS build
WORKDIR /app
COPY aotdemo.csproj .
RUN dotnet restore --runtime linux-x64 aotdemo.csproj

RUN apt-get update \
  && apt-get install -y --no-install-recommends \
  clang zlib1g-dev

COPY . .
RUN dotnet publish -c Release -r linux-x64 -o out aotdemo.csproj

FROM scratch AS runtime
WORKDIR /app
COPY --from=build /app/out/* /app/

RUN rm -r /app/aotdemo.dbg
RUN rm -r /app/appsettings.Development.json
EXPOSE 8080
ENTRYPOINT ["/app/aotdemo"]
```

错误列表 输出 程序包管理器控制台 C# Interactive (.NET Core)

就绪

添加到源代码管理 选择仓库



FROM scratch

ADD alpine-minirootfs-3.18.5-x86_64.tar.gz /

CMD ["/bin/sh"]

https://github.com/alpinelinux/docker-alpine/blob/a83fe3481a89201ac736f735062360f5953be635/x86_64/Dockerfile



```
powershell in ~  
axzxs 82ms docker inspect 1335 powershell 14:53:39  
{  
  "Id": "13359764b034197c44b11a4fb715b43e21db22bd00af048d9cdcdf5460e88269",  
  "Created": "2023-12-06T05:31:47.328555916Z",  
  "Path": "/app/aotdemo",  
  "Args": [],  
  "State": {  
    "Status": "running",  
    "Running": true,  
    "Paused": false,  
    "Restarting": false,  
    "OOMKilled": false,  
    "Dead": false,  
    "Pid": 1188,  
    "ExitCode": 0,  
    "Error": "",  
    "StartedAt": "2023-12-06T05:31:47.974109055Z",  
    "FinishedAt": "0001-01-01T00:00:00Z"  
  },  
  .....  
}
```



```
powershell in ~  
axzxs 82ms docker inspect 1335 14:53:39  
{"Id": "13359764b034197c44b11a4fb715b43e21db22bd00af048d9cdcdf5460e88269",  
  "Created": "2023-12-06T05:31:47.328555916Z",  
  "Args": [],  
  "State": {  
    "Status": "running",  
    "Running": true,  
    "Paused": false,  
    "Restarting": false,  
    "OOMKilled": false,  
    "Dead": false,  
    "Pid": 1188,  
    "ExitCode": 0,  
    "Error": ""  
  }  
}
```




```
powershell in ~  
axzxs 82ms docker inspect 1335 14:53:39  
"Id": "13359764b034197c44b11a4fb715b43e21db22bd00af048d9cdcdf5460e88269",  
"Path": "/app/aotdemo",  
"Args": [],  
"StartedAt": "2023-12-06T05:31:47.974109055Z",  
"Status": "running",  
"Running": true,  
"Paused": false,  
"Restarting": false,  
"Created": "2023-12-06T05:31:47.328555916Z",  
"Dead": false,  
"Pid": 1188,  
"ExitCode": 0,  
"Error": "",  
"FinishedAt": "0001-01-01T00:00:00Z"
```

||
645ms



Thank
YOU



我是桂素伟
欢迎关注『桂迹』公众号

