**There are two ways to call the API**

1. **HttpClient**

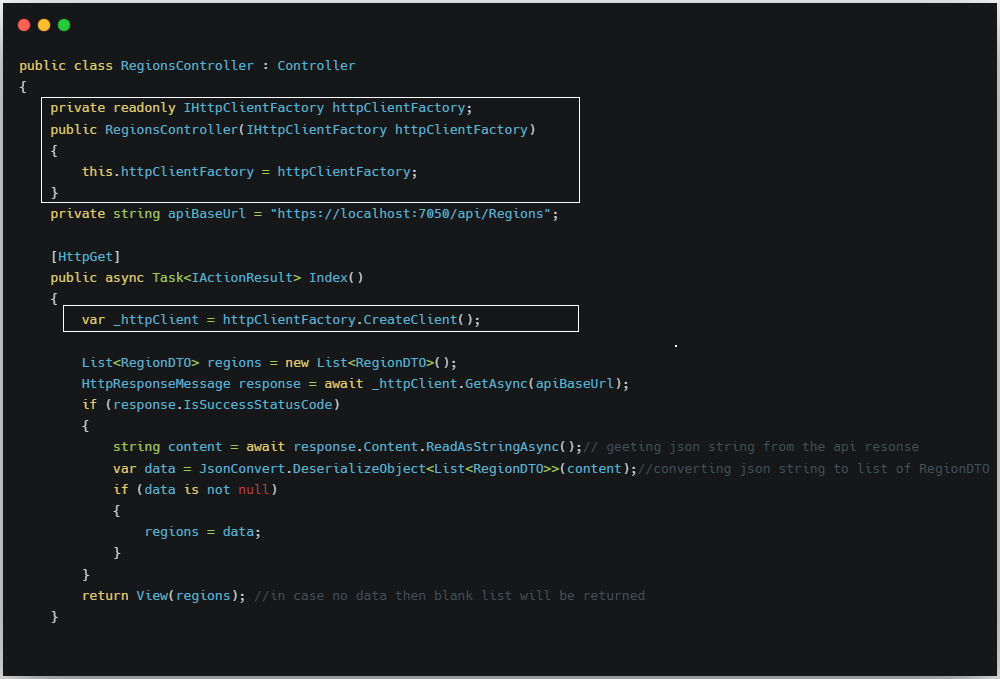
private HttpClient \_httpClient = new HttpClient();

**2- http client factory**

Insert the HTTP client factory inside the program file before the var app = builder line.Build();

// Registering the HttpClient for making API calls

builder.Services.AddHttpClient();



**Methods**

1. **Get Methods**

ReadFromJsonAsync<T>() **and** ReadAsStringAsync()

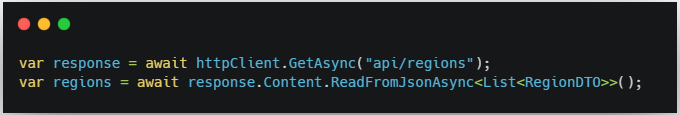
Agar **aapko JSON response ko directly object me convert karna hai**, to use:

👉 ReadFromJsonAsync<T>() (easy & clean)

Agar aap **manually string parse karna chahte ho**, to use:

👉 ReadAsStringAsync() (flexible but more work)

📌 ReadFromJsonAsync<T>() – Easy and Clean



☑️ Auto JSON deserialization  
☑️ Less code  
☑️ Safe & readable

📌 ReadAsStringAsync() – Manual Parsing



☑️ More control (e.g., logging raw JSON)  
☑️ Useful if the API doesn't return clean JSON

**Library for serializing and deserializing**

Newtonsoft.Json

1. **Post Method**

PostAsJsonAsync, PostAsync, and SendAsync

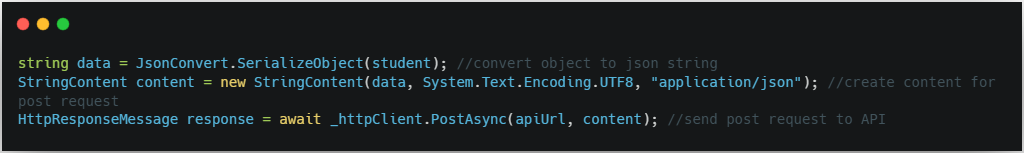
### **🔹 SendAsync**

➡️ **Flexible method** hai, jisme aap GET, POST, PUT, DELETE — **kisi bhi type ki request** bhej sakte ho.



### **🔹 PostAsync**

➡️ Ek **shortcut method** hai sirf POST requests ke liye.



### 🔁 Summary Table:

| **Feature** | **PostAsync()** | **SendAsync()** |
| --- | --- | --- |
| Request Type | Only POST | Any: GET, POST, PUT, DELETE etc. |
| Simplicity | Simple and quick | Flexible and powerful |
| Custom Headers | ❌ Not directly | ✅ Fully supported |
| Body Control | Basic only | Full control over HttpRequestMessage |

Agar aap POST, PUT, DELETE ko custom headers ke sath bhejna chahte hain (e.g. bearer token), to **SendAsync best** rahega.

### **PostAsJsonAsync()**

### **Concept:** Directly object ko JSON bana ke bhejta hai.

