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

1. **HttpClient**

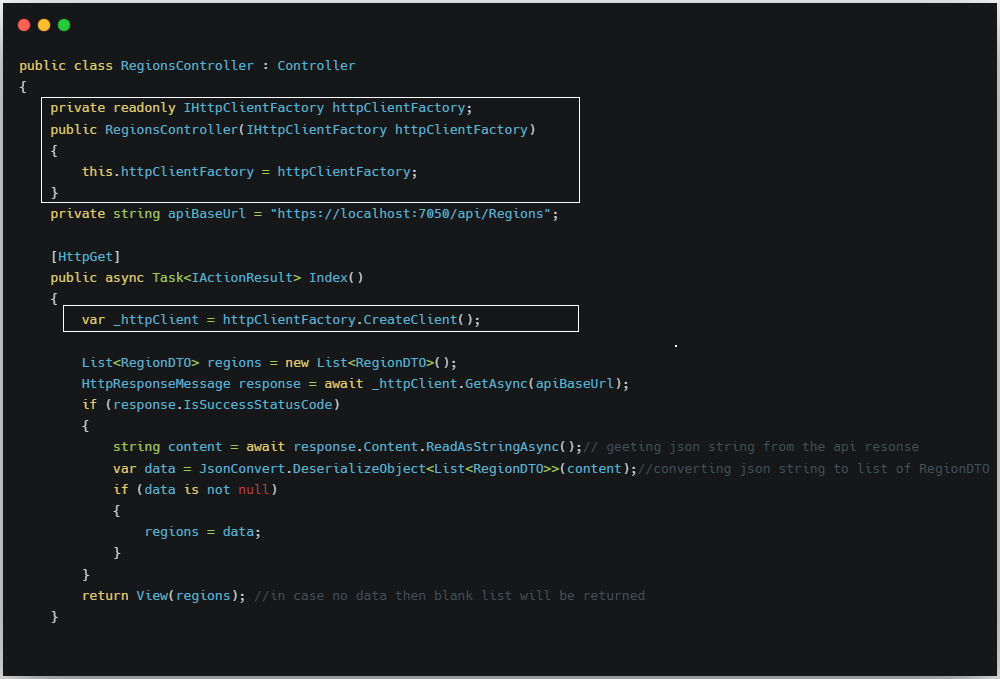
private HttpClient \_httpClient = new HttpClient();

**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. **GetMethod()**

GetAsync() vs GetFromJsonAsync() **in ASP.NET Core**

### **✅ 1. HttpClient.GetAsync()**

📌 **Low-level method**  
Sirf API se response lata hai (HttpResponseMessage deta hai) — parsing aapko manually karni padti hai.

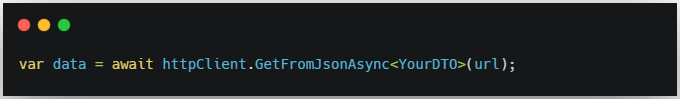


🧠 **Use When:**

* Aapko headers, status code ya raw JSON inspect karna ho.
* Advanced error handling chahiye.

### **✅ 2. HttpClient.GetFromJsonAsync<T>()**

📌 **High-level shortcut**  
Ye GetAsync + Deserialize dono ka combined version hai. Direct object return karta hai.



🧠 **Use When:**

* Simple GET call hai.
* Sirf JSON object chahiye, status code ya headers important nahi.

### 🎯 Final Recommendation:

| **Situation** | **Recommendation** |
| --- | --- |
| Simple GET + JSON | ✅ Use GetFromJsonAsync() |
| Need headers/status/logging | ✅ Use GetAsync() |

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

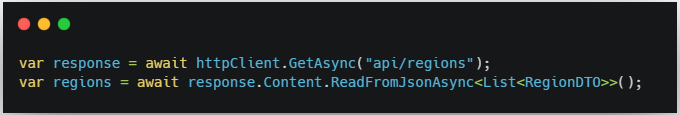
Agar **aapko JSON response ko directly object me convert karna hai**, to use:

👉 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

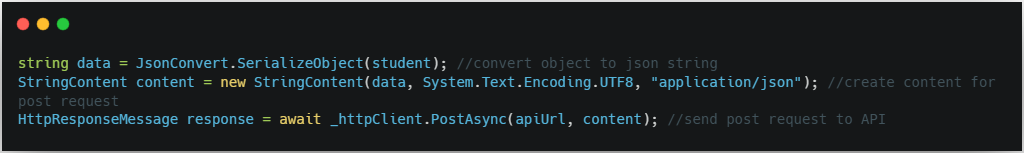
### **🔹 SendAsync –** required content as per format

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



### **🔹 PostAsync -** required url\_with\_Id and json\_content

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



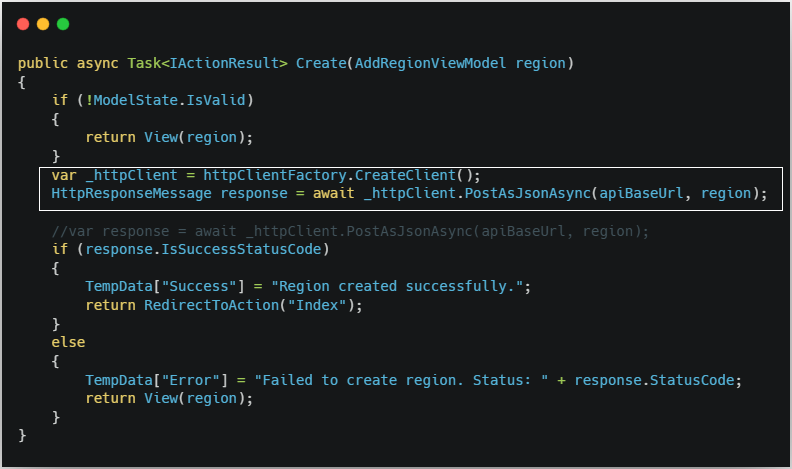
### 🔁 Summary Table:

| **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()** – required

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



1. **Put Method –** required url\_with\_Id and json\_content and id(optional)

SendAsync(), PutAsync()

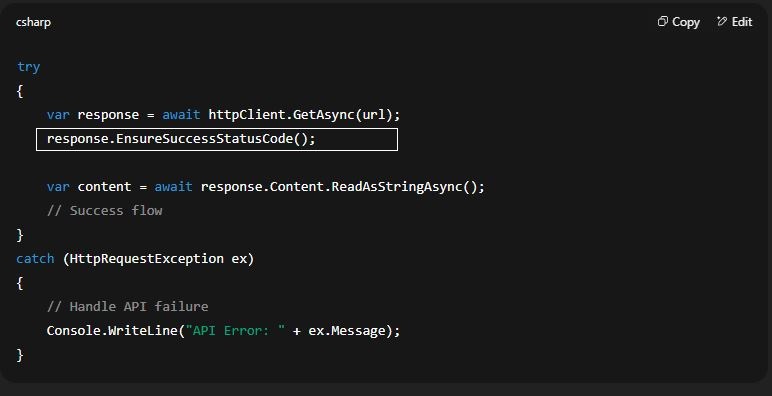
SendAsync()- Nzwalk

PutAsync()- programentor

1. **Delete Method –** required only url\_with\_Id

**Random**

**response.EnsureSuccessStatusCode()n n**



**response.EnsureSuccessStatusCode()**

Ye method check karta hai ki **HTTP response status code 200–299** ke range me hai ya nahi. Agar **successful** hai, to aage ka code chalega. Agar **error (400, 500, etc.)** aaya, to ye **exception throw** karega (HttpRequestException).

**Summary:**

* EnsureSuccessStatusCode() = 2xx check karta hai
* Agar error aaya to HttpRequestException throw karega
* Isse aapko alag se if (response.IsSuccessStatusCode) likhne ki zarurat nahi