

HTPP

HTTP (HyperText Transfer Protocol) é o protocolo usado para **trocar informações entre computadores na web**.

- Ele define **como um navegador ou front-end se comunica com um servidor**
- Ele funciona no modelo **cliente → servidor**
 - O **cliente** pede algo
 - O **servidor** responde

Exemplo simples do mundo real:

- Você abre um site no navegador → navegador manda requisição HTTP para o servidor → servidor responde com a página HTML → navegador mostra a página

O módulo que habilita HTTP no Angular

Para usar HTTP no Angular, você precisa **importar o módulo HttpClient**.

Se você estiver usando Angular **standalone** (sem AppModule), faz assim:

```
import { provideHttpClient } from '@angular/common/http';
import { bootstrapApplication } from '@angular/platform-browser';
import { App } from './app/app';

bootstrapApplication(App, {
  providers: [provideHttpClient()]
});
```

Isso habilita o Angular para fazer requisições HTTP.

Métodos HTTP mais usados

No Angular, o **HttpClient** oferece métodos correspondentes aos principais tipos de requisições:

Método Angular	HTTP	Para quê?
<code>http.get(url)</code>	GET	Buscar dados do servidor
<code>http.post(url, dados)</code>	POST	Enviar dados ao servidor
<code>http.put(url, dados)</code>	PUT	Atualizar dados no servidor
<code>http.delete(url)</code>	DELETE	Remover dados do servidor

- Todos retornam **Observable**, que é como o Angular entrega os dados de forma assíncrona.

Criando um serviço Angular para HTTP

O serviço é **onde centralizamos todas as chamadas HTTP**, para o componente ficar limpo:

```
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Observable } from 'rxjs';

@Injectable({ providedIn: 'root' })
export class UsuarioService {
  private apiUrl = 'http://localhost:8080/usuarios'; // endpoint do servidor

  constructor(private http: HttpClient) {}

  // GET - buscar usuários
  getUsuarios(): Observable<string[]> {
    return this.http.get<string[]>(this.apiUrl);
  }

  // POST - criar um novo usuário
  criarUsuario(usuario: any): Observable<any> {
    return this.http.post(this.apiUrl, usuario);
  }
}
```

Explicação passo a passo:

1. `HttpClient` → faz a comunicação HTTP

2. `getUsuarios()` → faz **GET** para `/usuarios`
 3. `criarUsuario()` → faz **POST** enviando um objeto
 4. `Observable` → permite o componente "**escutar**" quando a resposta chegar
-

Chamando o serviço no componente

```
import { Component, OnInit } from '@angular/core';
import { UsuarioService } from './usuario.service';

@Component({
  selector: 'app-home',
  standalone: true,
  template: `
    <h1>Lista de Usuários</h1>
    <ul>
      <li *ngFor="let u of usuarios">{{ u }}</li>
    </ul>
  `
})

export class Home implements OnInit {
  usuarios: string[] = [];

  constructor(private usuarioService: UsuarioService) {}

  ngOnInit() {
    // Aqui o componente chama o serviço
    this.usuarioService.getUsuarios().subscribe(res => {
      this.usuarios = res; // recebe dados do servidor e atualiza a tela
    });
  }
}
```

Explicação do fluxo:

1. Componente inicia → `ngOnInit()`
2. Chama serviço → `getUsuarios()`

3. Serviço faz HTTP GET → servidor retorna dados
 4. Componente recebe dados via `subscribe()`
 5. Template mostra os dados com `ngFor`
-

A ligação do Angular com HTTP

1. Você cria **um serviço Angular**
 2. Esse serviço **faz a requisição HTTP para o servidor**
 3. O servidor processa e retorna uma resposta
 4. O serviço entrega essa resposta para o componente Angular
 5. O componente mostra os dados na tela
-