1. Create the data : post request
2. To fetch the data : get request
3. To delete the data : delete request
4. To update the data : Put request

After setting the angular application for http then only we can perform the http methods operation

a. Import or configure the `HttpClientModule` into the `app.module.ts`
b. creating a new service with the help of angular-cli command `ng generate service service-name`
c. Inject the `HttpClient` in the service created in the previous step. Then you can see code as below:

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

@Injectable({
providedIn:  'root'
})
export class HttpService {

constructor(private http: HttpClient) { }

}
```

d. Also create one dummy API for the backend operation

   Now the setting part is completed. Now we can perform the crud operation

1. **Get method**: reading data from server
- Here I have created the service name of "users using the command "ng g s users"

Code related to the get methods

a. In users.service.ts file

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

@Injectable({
  providedIn: 'root'
})
export class UsersService {
  private url="https://64b11a36062767bc4825ae5a.mockapi.io/testing";

  constructor( private http:HttpClient) { }
//fetch the data from the server with the help of HTTP GET request
  getData(){
    return this.http.get(this.url);
  }
}
```

b. In app.component.ts file

```
import { Component } from '@angular/core';
import { UsersService } from './users.service';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'api-testing';
  employeelist: any;
  constructor(private user: UsersService) {

    // Getting/fetching data from server
    this.user.getData().subscribe((data: any) => {
      this.employeelist = data;
      console.log(data);
    })
  }
}
```

c. In app.component.html file

```
<h1>Reading Data</h1>

<!-- <div *ngFor="let employee of employeelist ;let i = index">
    <div class="m-5">{{employee.name}} , {{employee.score}} ,{{employee.email}} ,
{{employee.id}}</div>
</div> -->

<ul>
    <li *ngFor='let employee of employeelist'>{{employee.name}} , {{employee.score}}
,{{employee.email}} , {{employee.id}} </li>
</ul>
```

2. Post method: Sending Data to APIs

Here I have created one new form for the sending data to api’s

a. In app.component.html file

```
<div class="main-area">
  <div class="content-area">
    <div class="header">
      <h1>Manage Products</h1>
      <hr>
    </div>
    <div class="container">
      <!--Add product form-->
      <div class="form-area">
        <h3>Create Product</h3>
        <form #productsForm="ngForm" (ngSubmit)="onProductcreate(productsForm.value)">
          <label>Procuct Name</label>
          <input type="text" name="pName" ngModel>

          <label>Procuct Description</label>
          <input type="text" name="desc" ngModel>

          <label>Procuct Price</label>
          <input type="text" name="price" ngModel>


          <input type="submit" value="Add Product">
        </form>
      </div>

      <!--Display product area-->
      <div class="product-display-area">
        <h3>All Products</h3>
        <table id="products">
          <tr>
            <th>#</th>
            <th>Name</th>
            <th>Description</th>
            <th>Price</th>
            <th></th>
          </tr>
          <tr>
            <td>1</td>
            <td>iPhone</td>
            <td>iPhone Pro 11</td>
            <td>$1299</td>
            <td><button class="btn-delete">Delete</button></td>
          </tr>
        </table>
        <hr>
        <div class="action-btn-container">
          <button class="btn-fetch">Fetch Product</button>
          <button class="btn-clear">Clear Product</button>
```

```
        </div>
      </div>
    </div>
  </div>
</div>
```

b. In app.component.ts file

```
import { HttpClient, HttpHeaders } from '@angular/common/http';
import { Component } from '@angular/core';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent{
  title = 'AngularHttpRequest';
  constructor(private http:HttpClient){}

  onProductcreate(products:{pName:string, desc:string, price:string})
  {
    console.log(products);
    const headers=new HttpHeaders({'myHeader':'proacademy'})
    // post(url:string, body:any, options:{headers?:...............})
    this.http.post('https://angularbysuraj-default-rtdb.firebaseio.com/products.json',
    products,{headers:headers})
    .subscribe((res) =>{
      console.log(res);
    })
  };
}
```

Note: instead of console window , we will be checking network window and API page for the output

To convert the simple form into template driven form
Steps

1. Add <name="" ngModel> in each input element
2. To use ngModal , import FormsModule in app.module.ts file
3. Add template variable and (ngSubmit) function in the form element
4. Then declare the same function in the typescript file I.e. .ts file.

Example:

```
<form #productsForm="ngForm" (ngSubmit)="onProductcreate(productsForm.value)">
          <label>Procuct Name</label>
          <input type="text" name="pName" ngModel>

          <label>Procuct Description</label>
          <input type="text" name="desc" ngModel>

          <label>Procuct Price</label>
          <input type="text" name="price" ngModel>

          <input type="submit" value="Add Product">
        </form>
```

In ts file

```
onProductcreate(products:{pName:string, desc:string, price:string})
  {

  };
```