# Effettuare chiamate

Chi effettua le chiamate è l’ HttpClient, questo deve essere importato nell’app.module.ts tramite la seguente riga :

import { HttpClientModule } from '@angular/common/http';

Sempre nello stesso fine deve essere indicato tra gli imports dell’ @NgModule :

@NgModule({

    imports: [

        …

        HttpClientModule,

Le chiamate, sia di tipo Get che Post, sono inserite in un service, tenendo sempre presente che è il verbo subscribe che esegue la chiamata.

Chiamata di tipo Get nel service :

  getUserById(id: number ): Observable<userModel> {

    return this.\_http.get( this.domainApiPrefix + '?Id=' + id ).pipe

    (map( response => {

      let r = response as userModel;

      return r;

    } ))

  }

Nell’esempio viene richiamato, tramite il suo, un oggetto di tipo userModel, tramite il pipe e il map della risposta sull’oggetto.

L’equivale di questa chiamata è, ad esempio:

<http://localhost:29993/api/Profile?Id=1>

La chiamata nel component :

    this.\_profileSVC.getUserById(this.user.id).subscribe(res => {

        this.user = res;

        this.userForm.patchValue(this.user);

    });

La chiamata viene eseguita perché c’è il subscribe e i valori restituiti aggiornano la form.

Chiamata di tipo Post

Nel service:

  saveUser(user: userModel) {

    this.\_http.post(this.domainApiPrefix, user).subscribe(res => {

      let r = res as userModel;

      this.\_toastSVC.success("Notifica", "Scrittura eseguita correttamente")

      return r;

    },

    error => {

      let messagesOut: string[] = [];

      error.error.forEach(element => {

        messagesOut.push(element.ErrorDescription);

      });

      this.\_toastSVC.error("Errore", messagesOut.toString());

    })

  }

Questa chiama passa, nell’Header l’oggetto user e poi in base alla risposta chiama un service che notifica attraverso un messaggio toast se la stessa ha avuto un esito positivo o negativo.

Nel component :

    public Save() {

        // Aggiorno i dati dell'oggetto con i dati inseriti/modificati nel form

        Object.assign(this.user, this.userForm.value);

        // Eseguo l'aggiornamento dei dati dell'utente

        this.\_profileSVC.saveUser(this.user);

    }