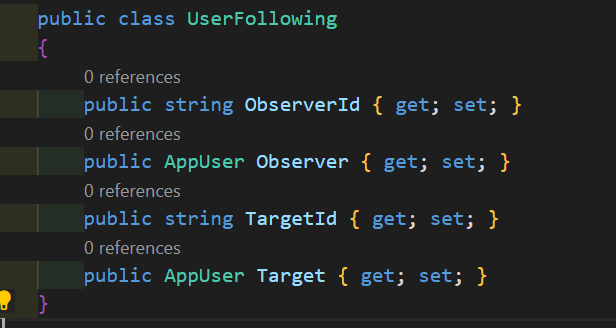
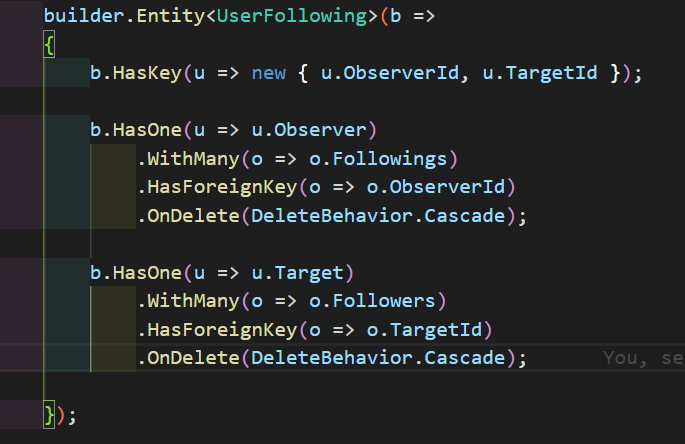
.net Core 5 y EFCore 5 tiene limitaciones al referenciar relaciones muchos a muchos en los mismos elementos.

-Vamos a crear una “Joint Entity”, en este caso UserFollowing para tratar de solventar esto. Usando observers and targets. Esta clase la podemos usar para followings y followers.

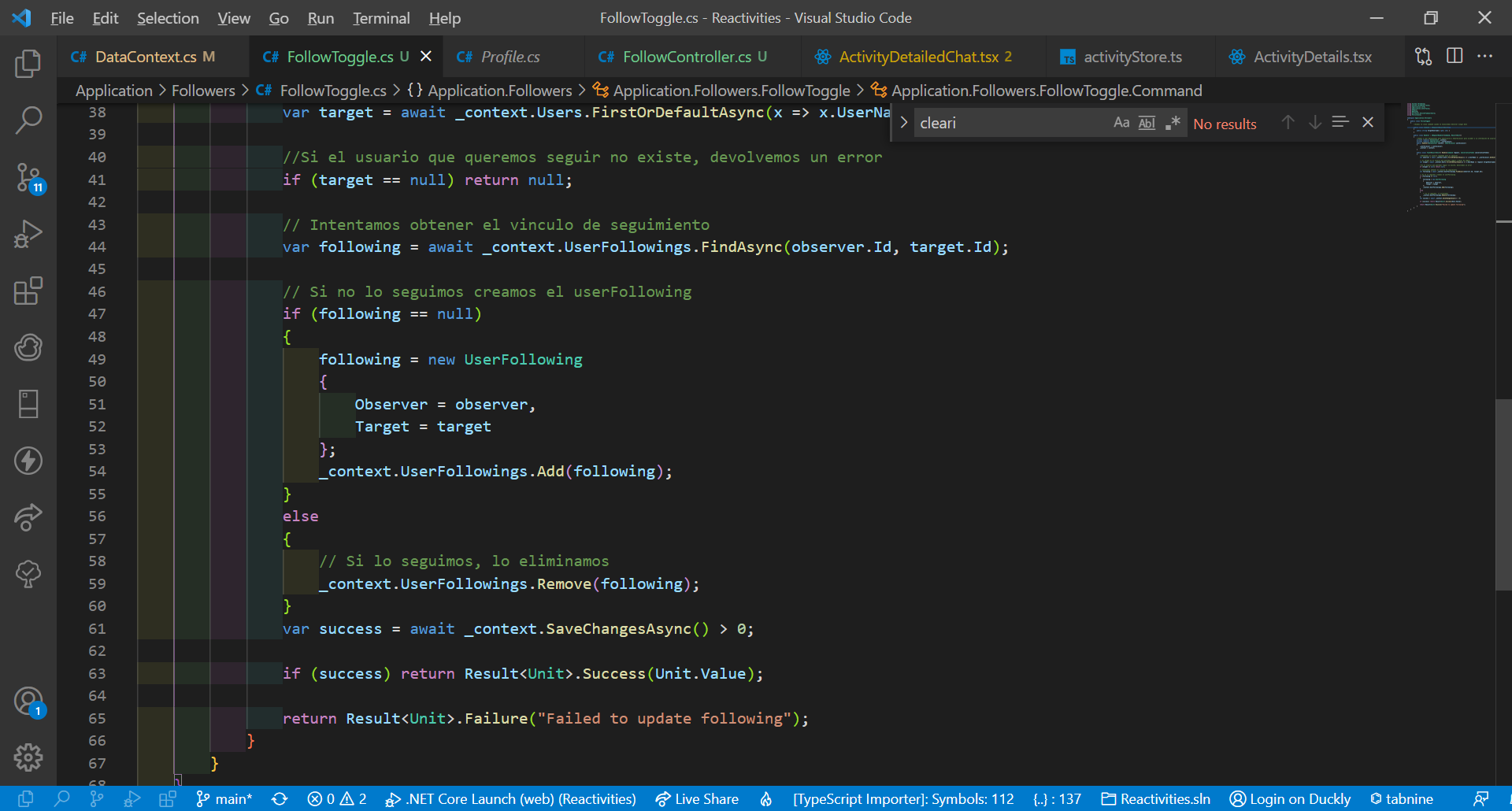


Para hacer la relación en la db en el DataContex tenemos que especificarlo asi:



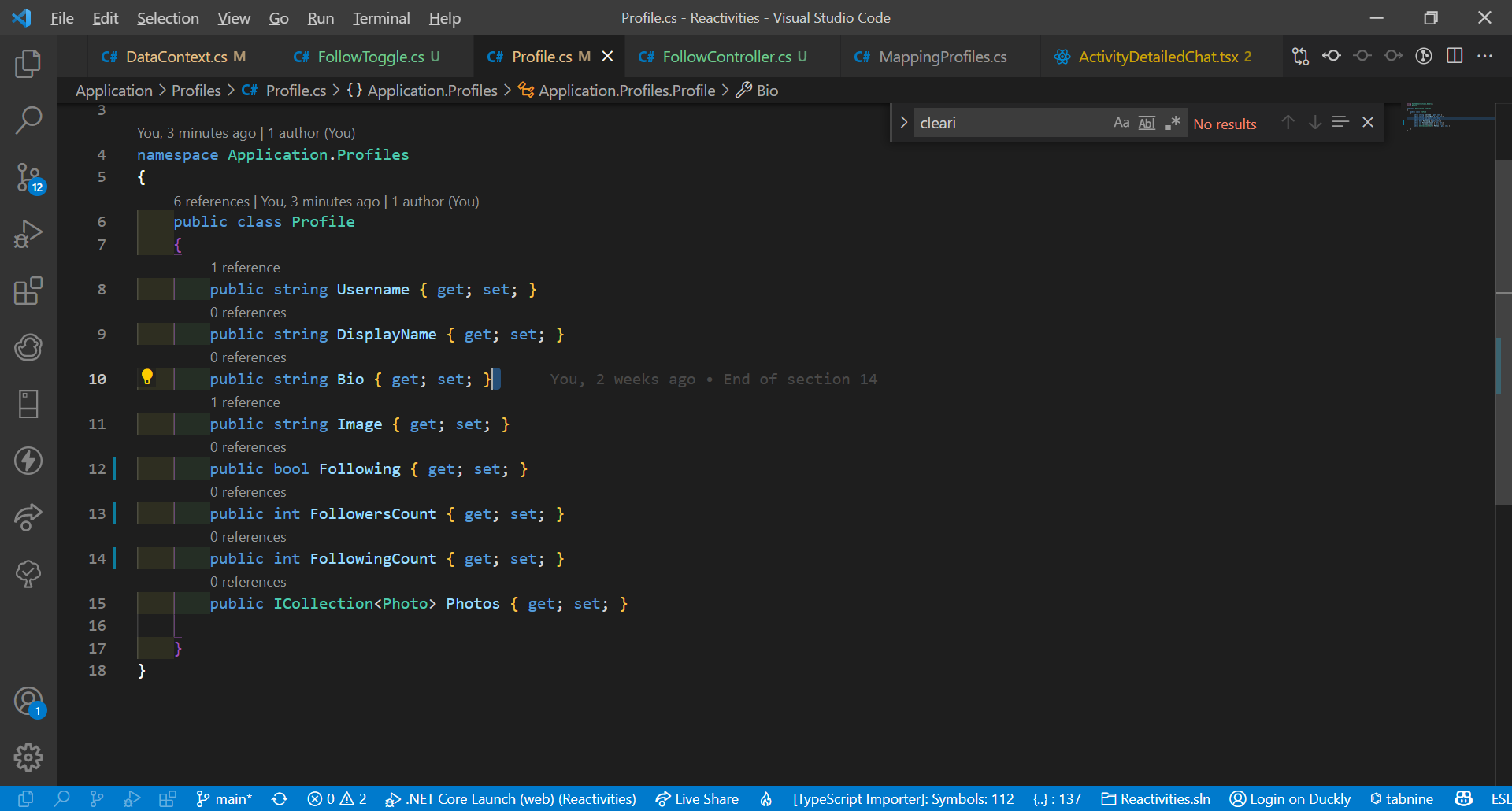
- Creamos en Application/Followers la Clase FollowToggle con el Handler para seguir y dejar de seguir. Como mediante el UserAccesor sabemos quien es el usuario logueado ya tenemos al “observer” y solo necesitamos que nos pasen al target.

- Buscamos en la tabla de relación si existe un seguimiento con esos datos. Si no lo hay lo creamos, si lo hay lo borramos.



-Luego hacemos el controller para consumir este método. (los controllers siempre en API)

-Ya que lo queremos mostrar en el profile tenemos que pasarle las propiedades



-Tenemos que modificar el mapeo para que cuenten los Followings y Followers en AppUser y los pasen al Profile

namespace Application.Core

{

    public class MappingProfiles : Profile

    {

        public MappingProfiles()

        {

            CreateMap<Activity, Activity>();

            CreateMap<Activity, ActivityDto>()

                .ForMember(d => d.HostUsername, o => o.MapFrom(s => s.Attendees

                    .FirstOrDefault(x => x.IsHost).AppUser.UserName));

            CreateMap<ActivityAttendee, AttendeeDto>()

                .ForMember(d => d.DisplayName, o => o.MapFrom(s => s.AppUser.DisplayName))

                .ForMember(d => d.Username, o => o.MapFrom(s => s.AppUser.UserName))

                .ForMember(d => d.Bio, o => o.MapFrom(s => s.AppUser.Bio))

                .ForMember(d => d.Image, o => o.MapFrom(s => s.AppUser.Photos.FirstOrDefault(x => x.IsMain).Url));

            CreateMap<AppUser, Profiles.Profile>()

                .ForMember(d => d.Image, o => o.MapFrom(s => s.Photos.FirstOrDefault(x => x.IsMain).Url))

                //Agregamos el mapeo de las propiedades para llenar el Profile con los followings y followers

                .ForMember(d => d.FollowersCount, o => o.MapFrom(s => s.Followers.Count))

                .ForMember(d => d.FollowingCount, o => o.MapFrom(s => s.Followings.Count));

            CreateMap<Comment,CommentDto>()

            .ForMember(d => d.DisplayName, o => o.MapFrom(s => s.Author.DisplayName))

                .ForMember(d => d.Username, o => o.MapFrom(s => s.Author.UserName))

                .ForMember(d => d.Image, o => o.MapFrom(s => s.Author.Photos.FirstOrDefault(x => x.IsMain).Url));

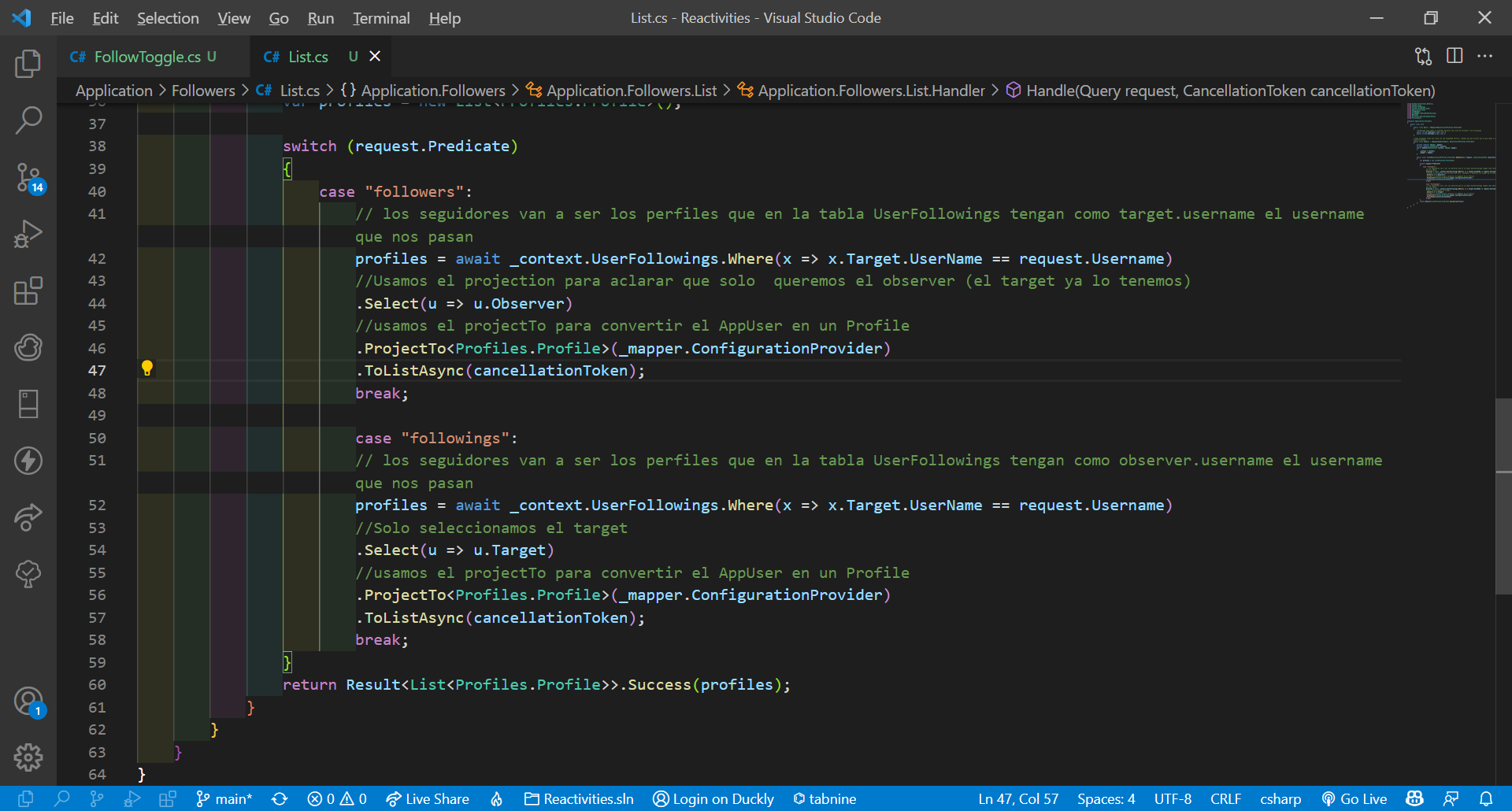
        }

    }

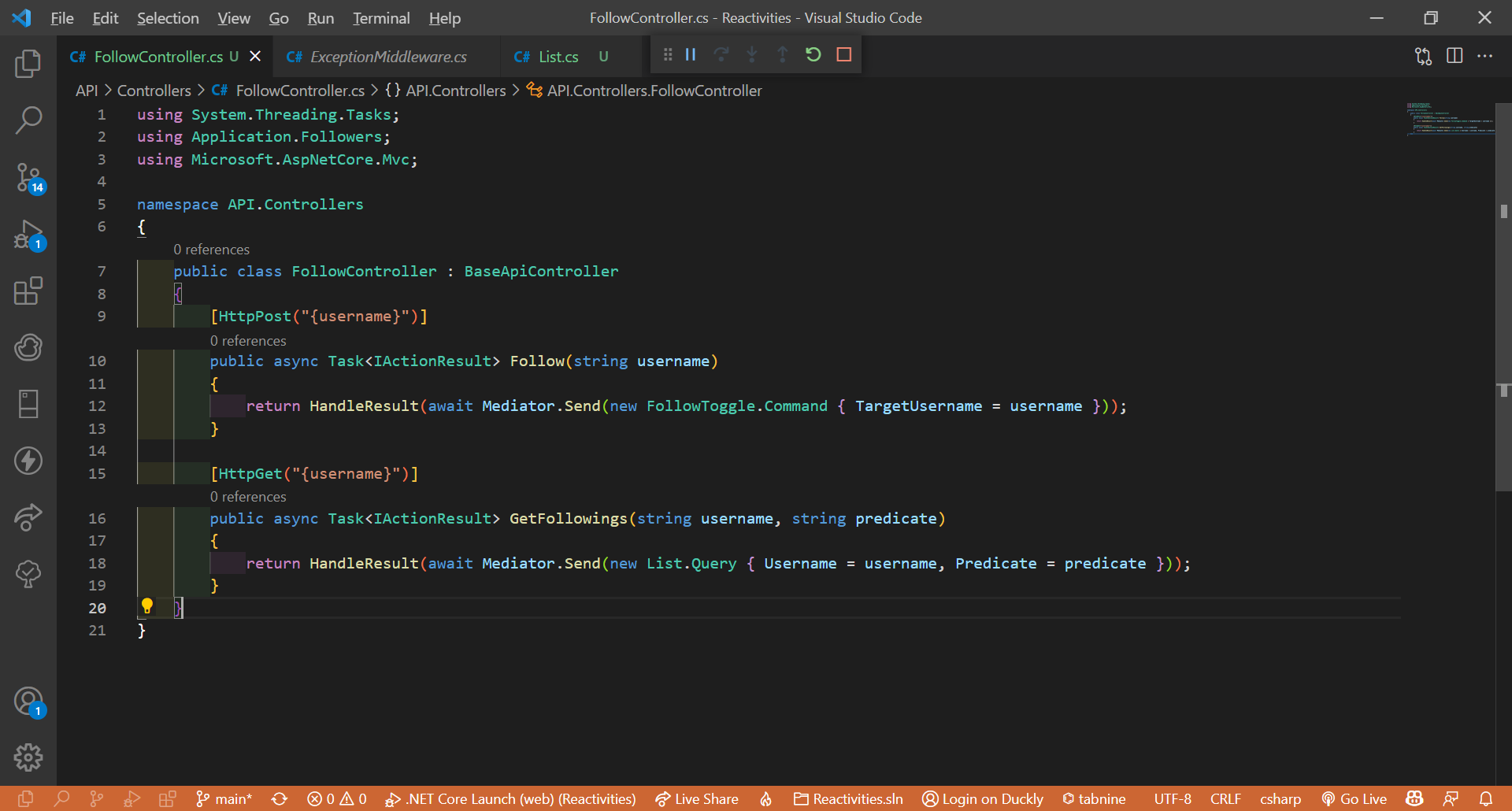
}

-Creamos el handler que devuelva la Lista de usuarios que sigue un usuario en particular o lista de los usuarios que lo siguen.

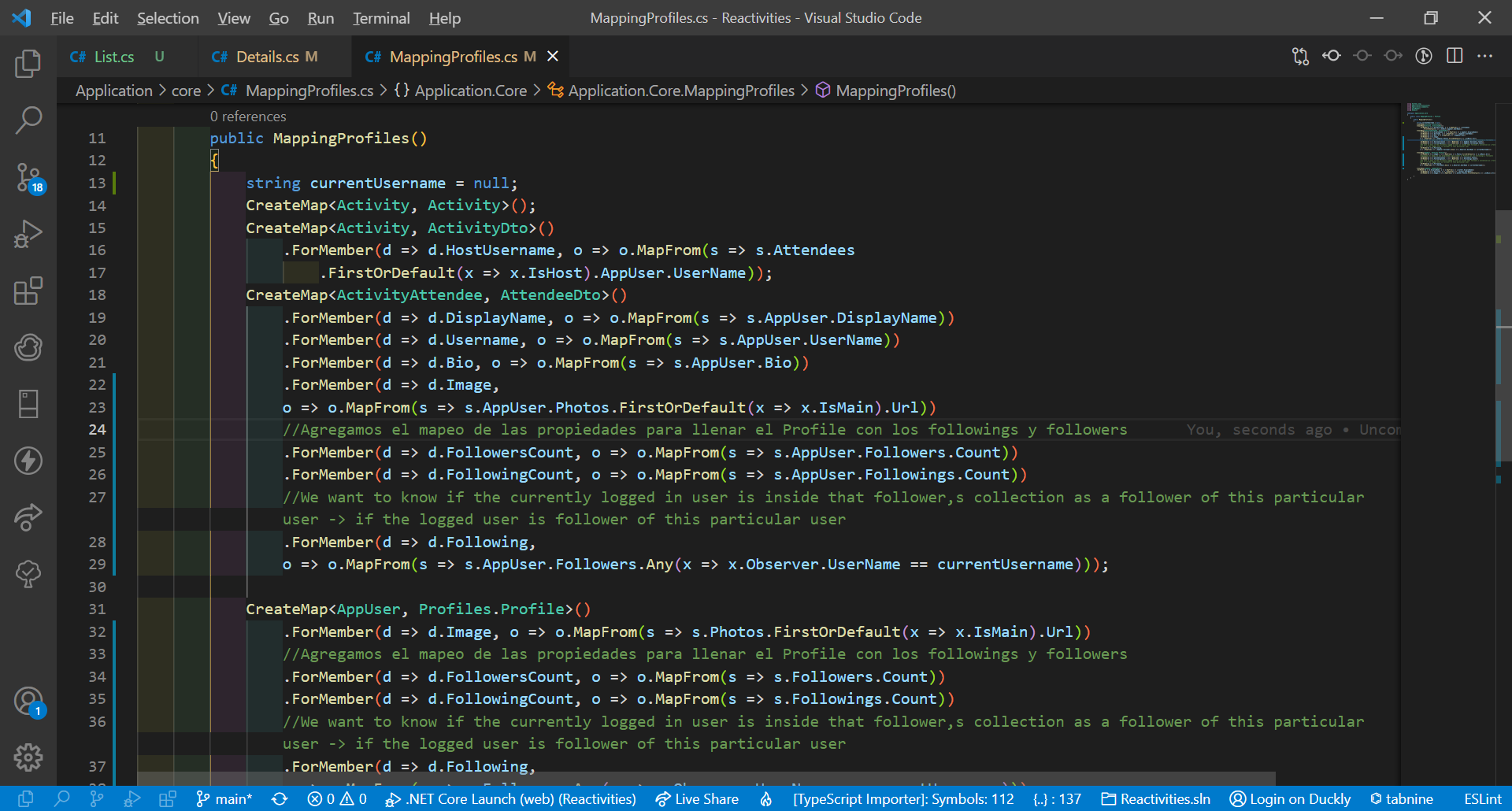
Profiles.List



-Agregamos el método en el Controller para consumir esta List



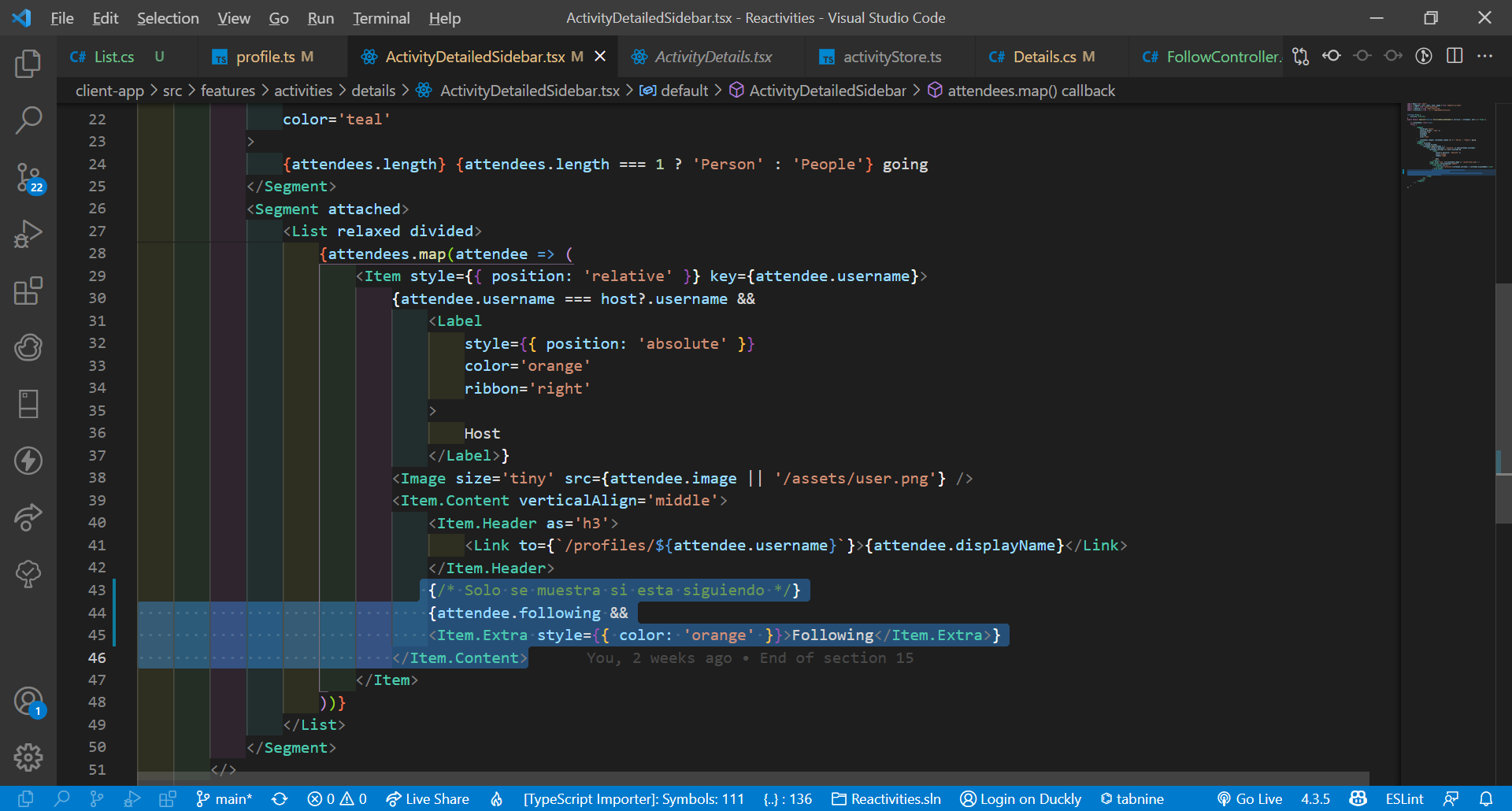
-Mapeamos los followers y followings y también el following. Usamos para esto el userAccesor para pasar el usuario logueado actual (currentUsername) a través del ProjectTo para ver si el usuario logueado sigue a un usuario en particular.

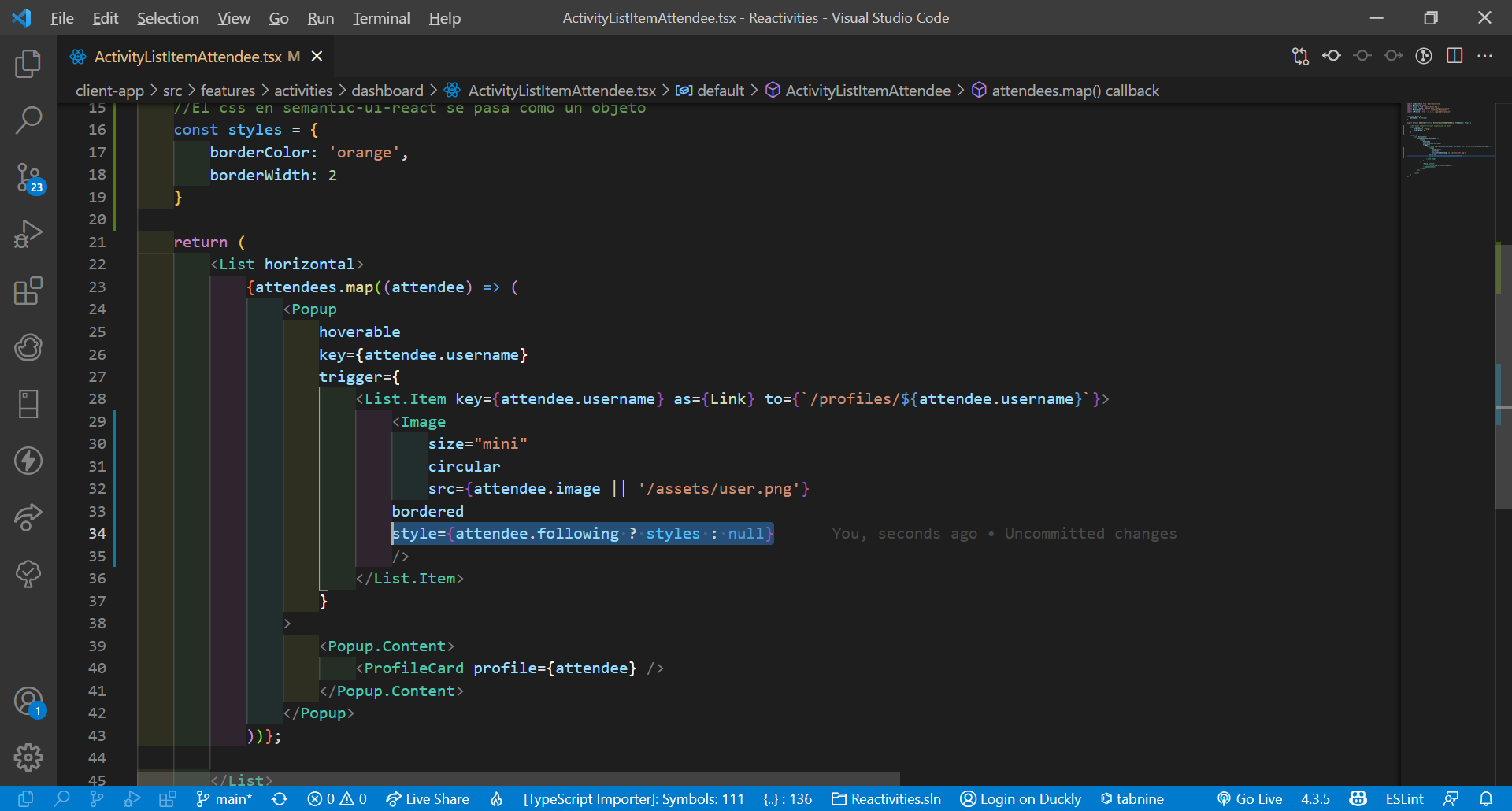


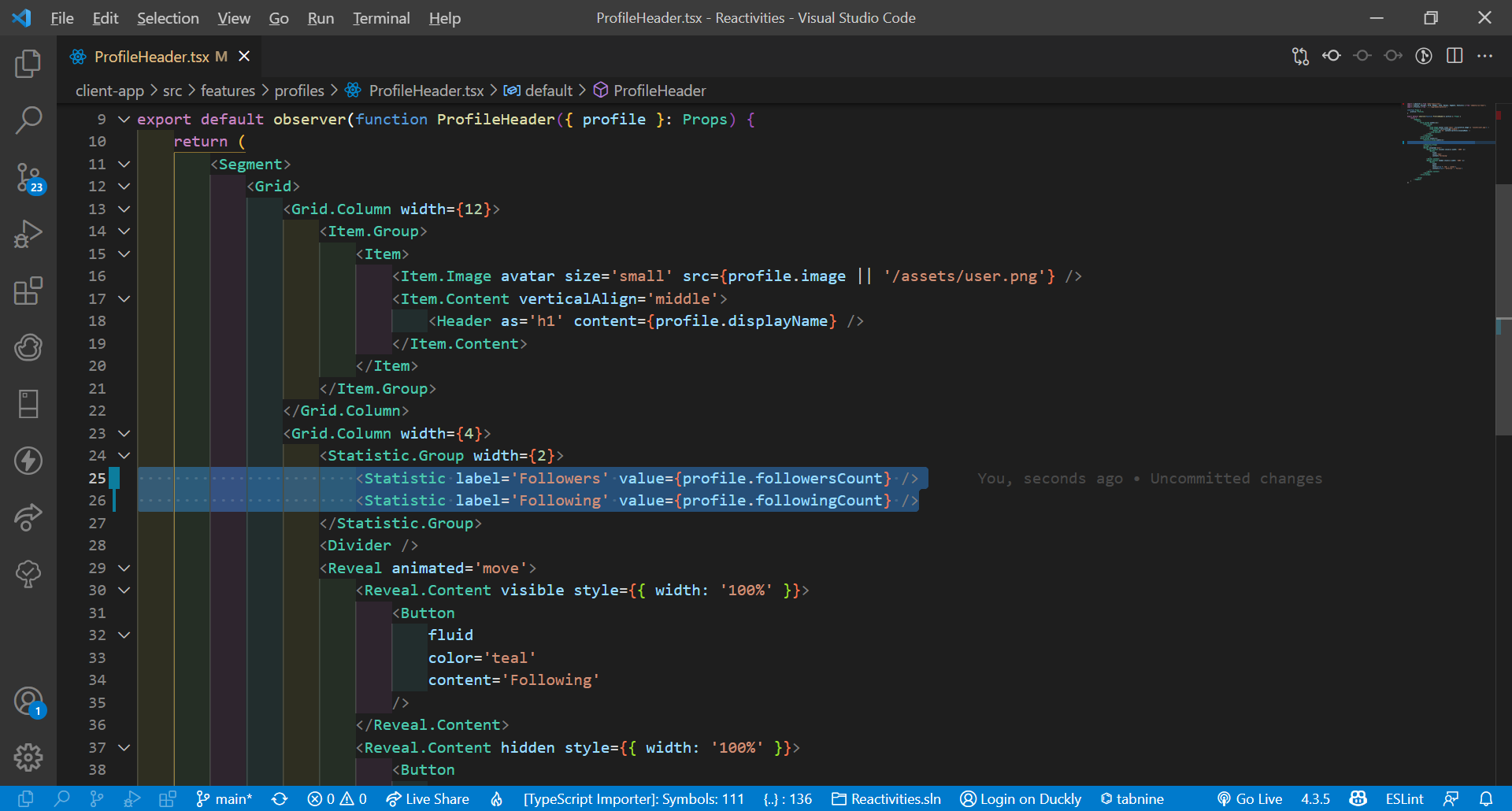


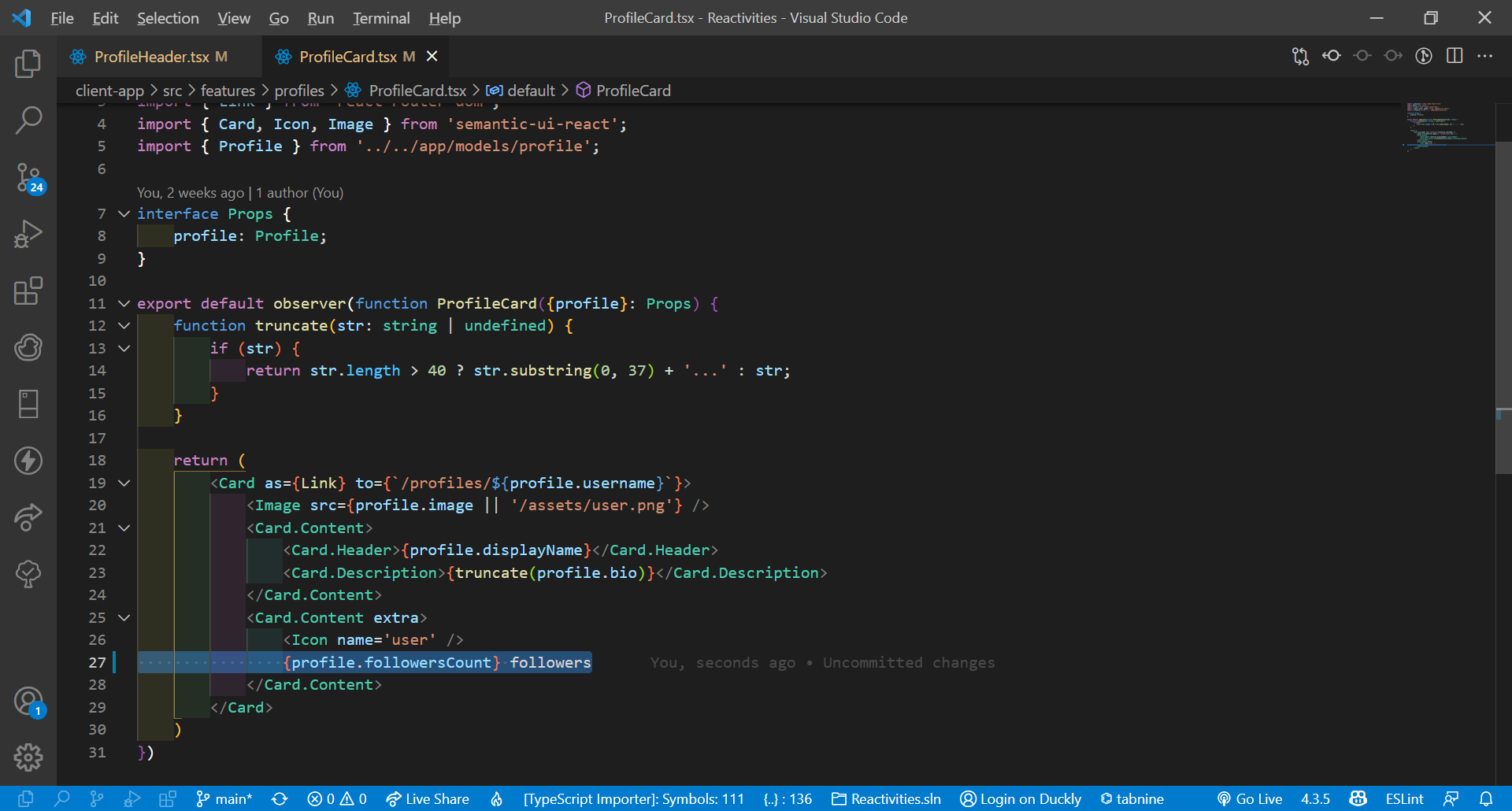
El currentUsername debemos pasarlo en cada método o clase que devuelva un profile con los followers.

Para mostrarlo en el UI tenemos que agregar los campos FollowingCount, FollowersCount y Following en la interfaz de profile.ts (asi figura en los attendees).

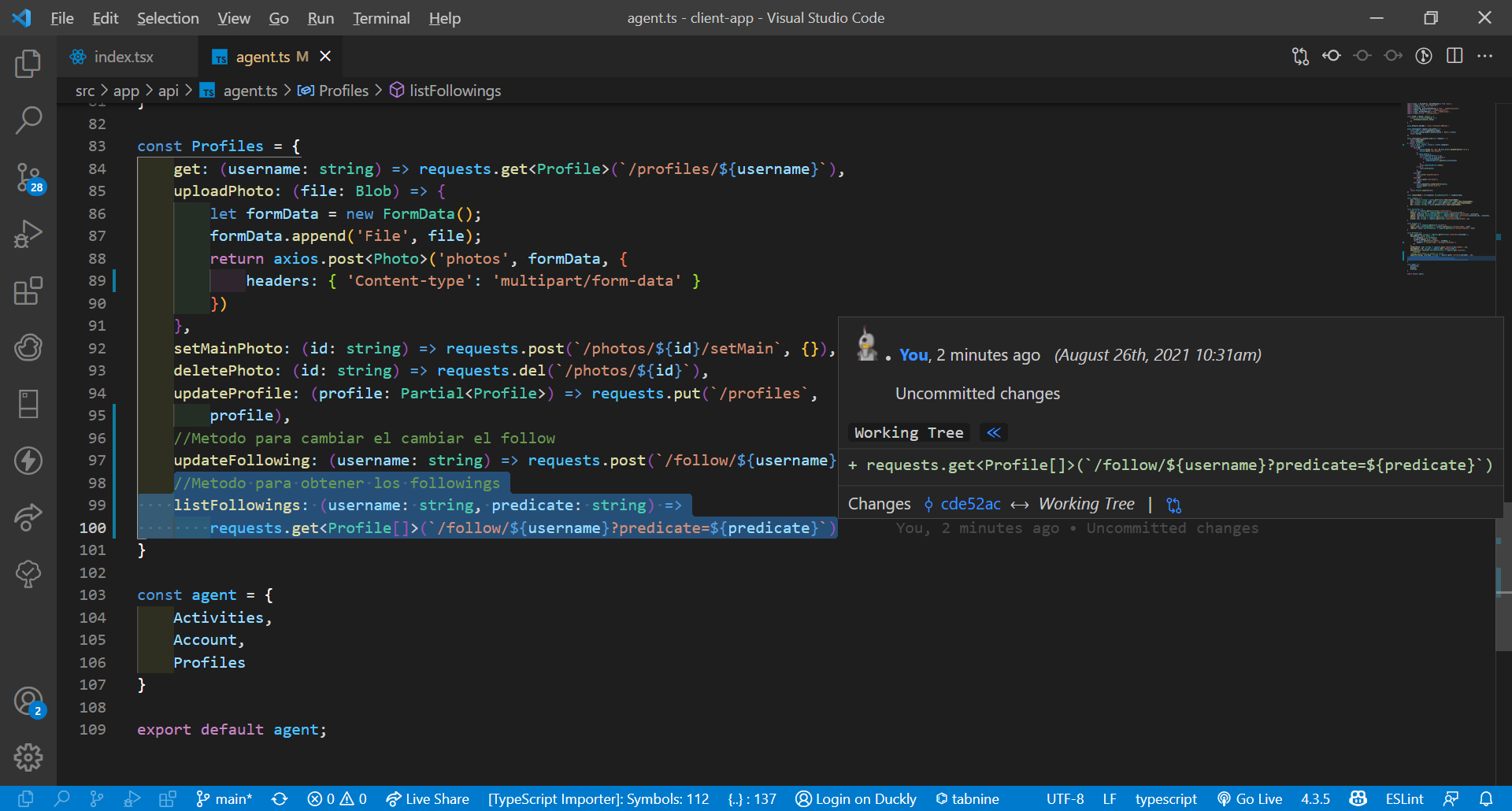




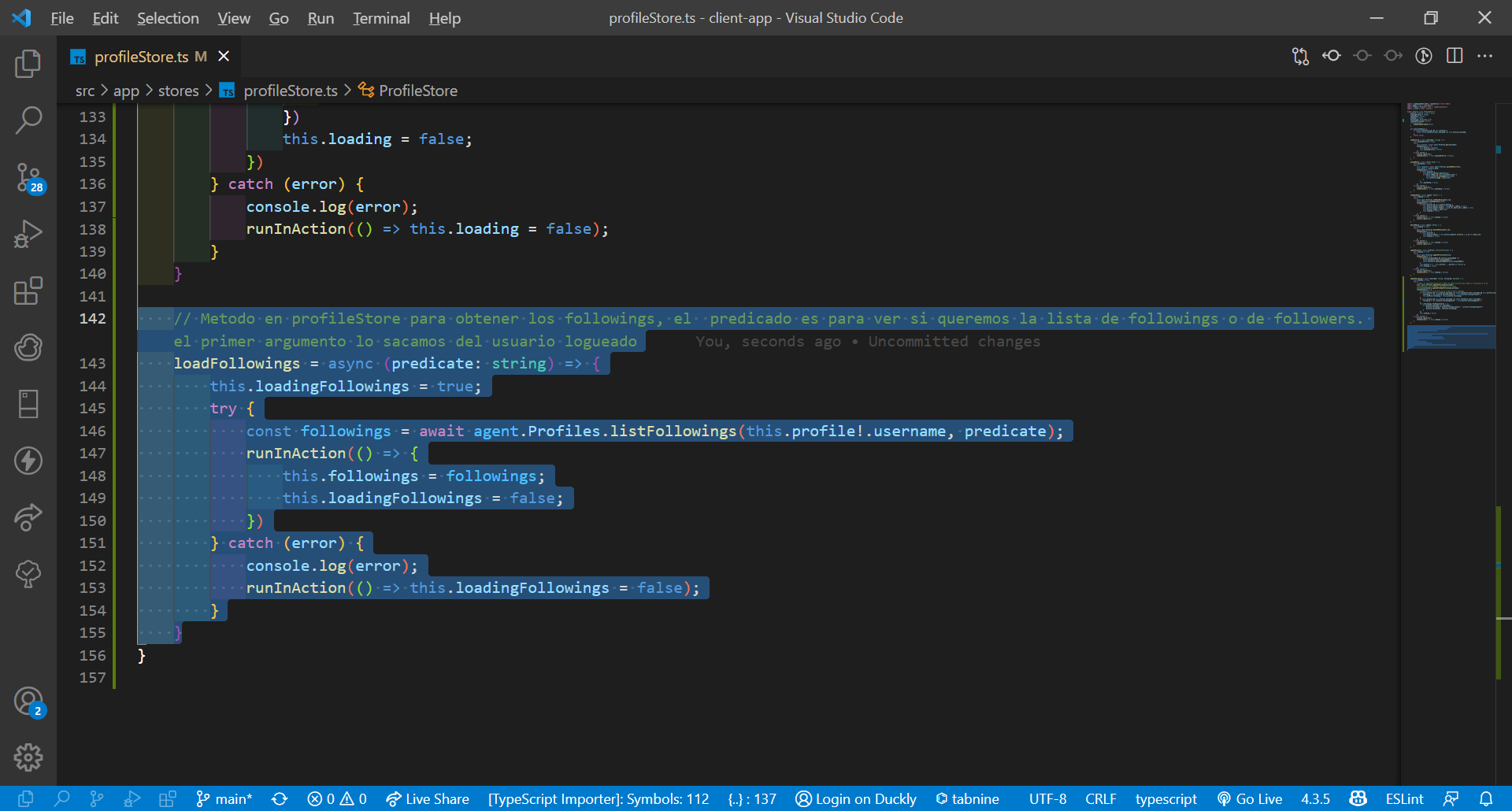


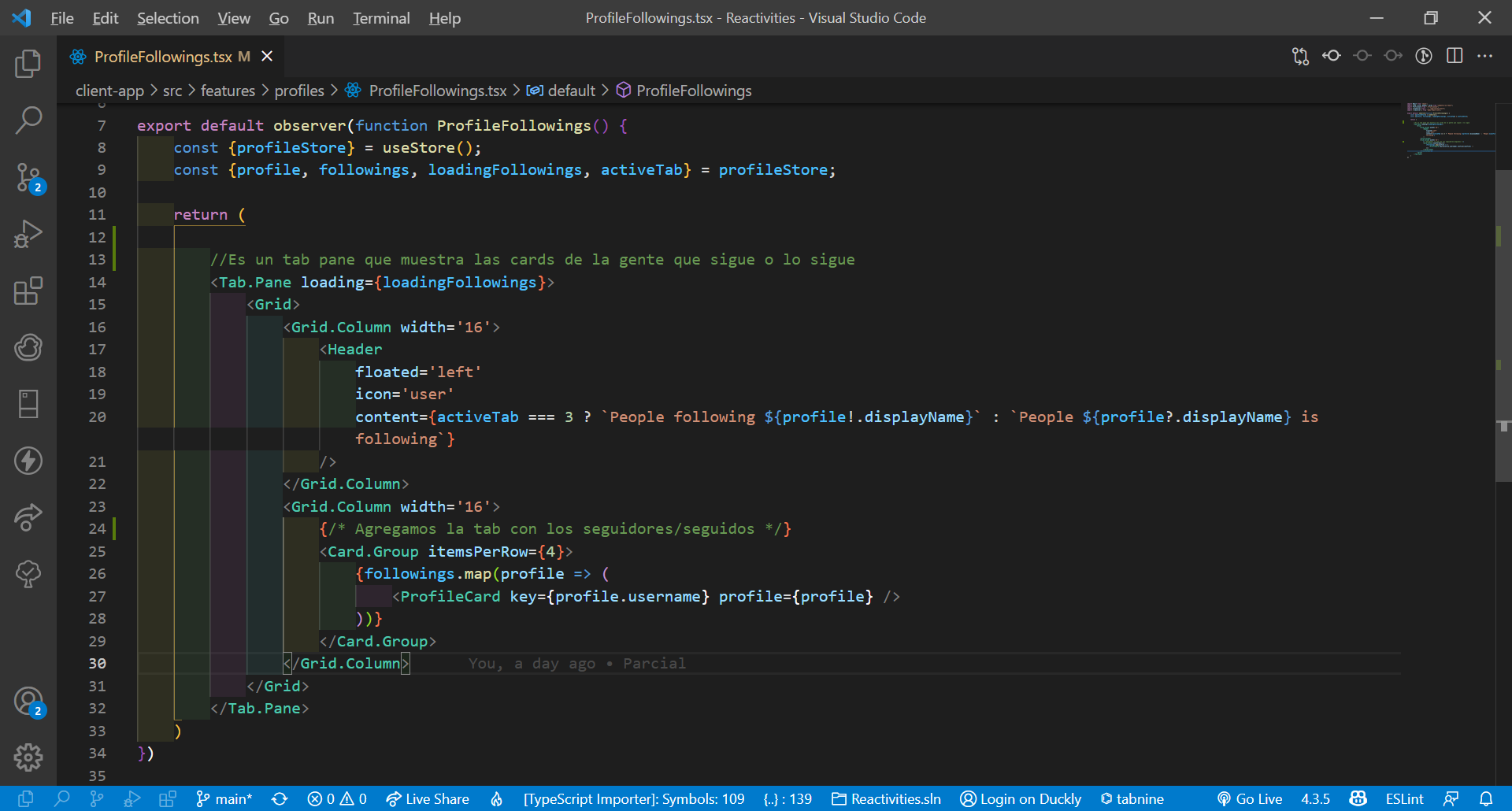


Creamos un metodo en agent.ts -> Profiles para obtener los followings

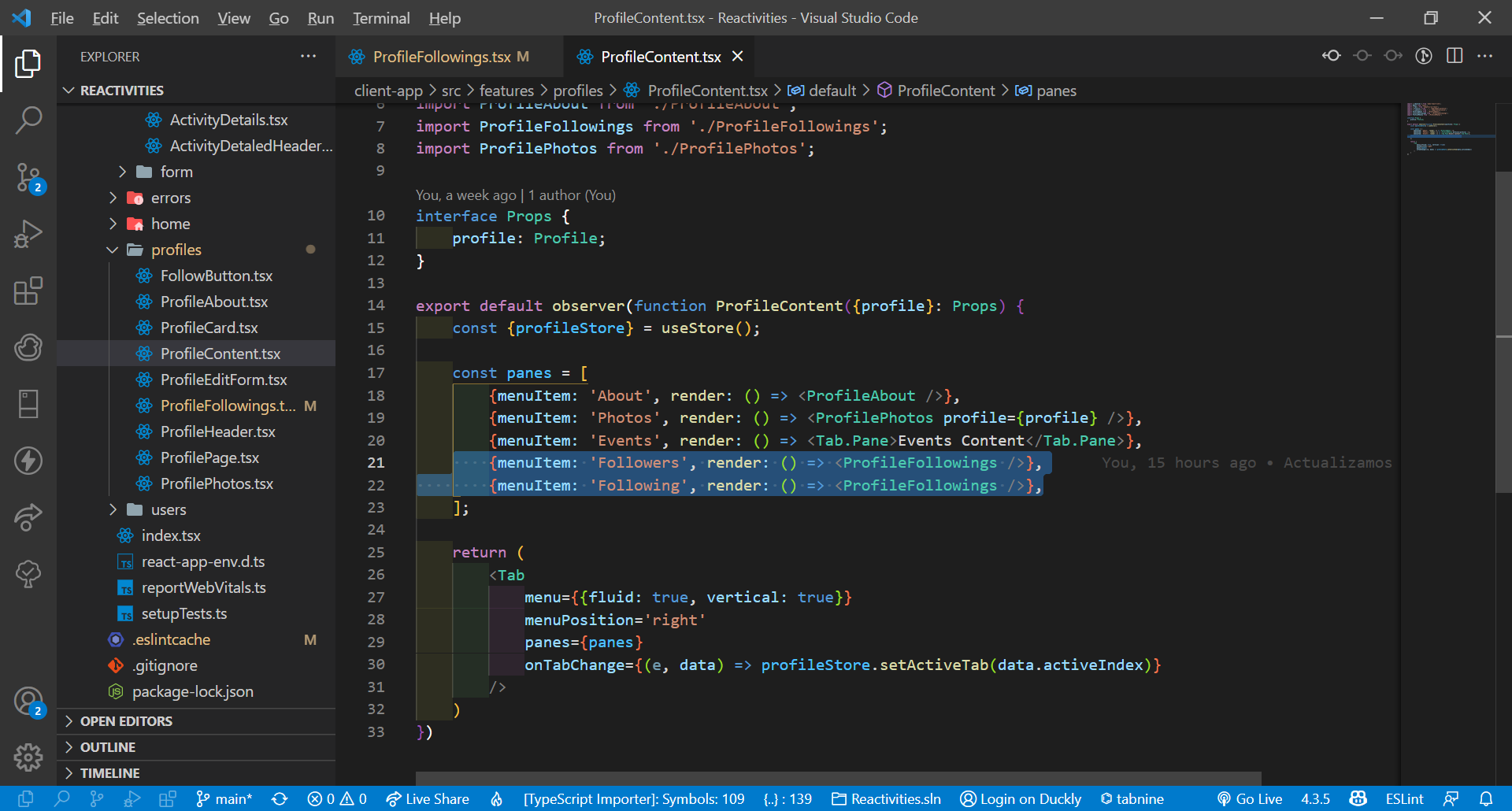


En profileStore creamos el metodo loadFollowings que va a consumir el listFollowings que declaramos en agent.ts y va a llenar los followings





Actualizamos el panel con las tab



Para poder cargar followings o followers vamos a usar mobx reactions que escuche que numero de tab esta seleccionada y según eso te los cargue.

Va a reaccionar al cambio de activeTab mediante una función setActiveTab que vamos a llamar desde profileContent

