

## TAP operator

### Transparently perform actions or side-effects, such as logging

When you transform the response using RXJS operator (like pipe), then if you want to perform any action transparently means **action will not effect in the response**.

### Use of TAP operator:

1. Console log response data
2. Change property value like any Boolean property
3. An emit any value in subject
4. 4. Redirect to any other url

### Real time Examples:

1. When data is fetching from api that time you want to do any actions like button click, redirect to any url, change the value of any subject, any subject/behavior subject will be subscribe/unsubscribe, open any dialog modal, console log of data etc
2. If at login time, you need to check user is prime user/normal user. As per user credential, you need to give access of your page. Then at login time, you have to call the subject by which you need to set the access of modules of your application

### Login Example

```
public logout(): Observable<void> {  
  const logout = this.socialAuthService.signOut();  
  return from(logout)  
    .pipe(  
      take(1),  
      tap(() => this.router.navigate([ '/' ])),  
      map(() => this.socialUser.next(null)),  
    );  
}
```

```
<div class="widget-account">  
  <div class="user" *ngIf="user !== null">  
    <h3 *ngIf="user && user.name">{{ user.name }}</h3>  
    <a (click)="logout()">Logout</a>  
  </div>  
  <img *ngIf="user && user.photoUrl" [src]="user.photoUrl">  
</div>
```

```
logout(): void {  
  this.authService  
    .logout()  
    .subscribe();  
}
```

```
public checkLoggedIn(): Observable<SocialUser> {  
  return this.socialAuthService.authState.pipe(  
    map((currentUser) => this.convertToUser(currentUser)),  
    tap((account) => console.log('Tap after convertToUser => ', account)),  
    map((account) => {  
      this.socialUser.next(account);  
      return account;  
    })  
  );  
}
```

Using logout function I want to redirect to home page on logout button click . So I have user tap operator to redirect tap(() => this.router.navigate([ '/' ]))

Using checkLoggedIn function, I want to check the socialUser value, if I am getting value then user is logged in otherwise not. I am using tap operator to console the account after mapping with convertToUser.

From operator is being used to convert promise value (logout) to observable.

```
const logout = this.socialAuthService.signOut();
```

## Hindi Video Tutorial:

[Tap in Rxjs | Rxjs Tap Operator Angular | Rxjs Tutorial in Hindi \(2020\) \[Ep - #12\]](#)

File Edit Selection View Go Run Terminal Help authentication.service.ts - social-login - Visual Studio Code

EXPLORER

OPEN EDITORS

src > app > social-login > service > TS authentication.service.ts > AuthenticationService > checkLoggedIn > map() callback

25 }  
26 this.signInWithRedirect();  
27 }  
28  
29 public signInWithRedirect(): void {  
30 this.router.navigate(['/home']);  
31 }  
32  
33 public checkLoggedIn(): Observable<SocialUser> {  
34 return this.socialAuthService.authState.pipe(  
35 map((currentUser) => this.convertToUser(currentUser)),  
36 tap((account) => console.log('Tap after convertToUser => ', account)),  
37 map((account) => {  
38 this.socialUser.next(account);  
39 return account;  
40 }));  
41 };  
42 }  
43  
44 public logout(): Observable<void> {  
45 const logout = this.socialAuthService.signOut();  
46 return from(logout)  
47 .pipe(  
48 take(1),  
49 tap(() => this.router.navigate([ '/' ])),  
50 map(() => this.socialUser.next(null)),  
51 );  
52 }  
53  
54 private convertToUser(user: SocialUser): any {  
55 console.log('user => ', user);  
56 if (!user) {  
57 throw new Error('no user');  
58 }  
59  
60 //const uid = user.uid;  
61 const name = user.name;  
62 const email = user.email;  
63 const avatar = user.photoUrl;