


```
@Effect()
login$ = this.actions$
  .ofType(`[Login Page] Login`)
  .pipe(
    exhaustMap(auth =>
      this.authService
        .login(auth)
        .pipe(
          map(user => new LoginSuccess({ user })),
          catchError(error => of(new LoginFailure(error)))
        ),
    ),
  );
```





```
@Effect() logoutConfirmation$ = this.actions$
  .ofType(`[Auth] Confirm Logout`)
  .pipe(
    exhaustMap(() =>
      this.dialogService.open(LoginPromptComponent)
        .afterClosed()
        .pipe(
          map(confirmed => {
            if (confirmed) {
              return new LoginConfirmed();
            } else {
              return new LoginCancelled();
            }
          })
        ),
    ),
  );
```

```
@Effect()
login$ = this.actions$
  .ofType(`[Login Page] Login`)
  .pipe(
    exhaustMap(auth =>
      this.authService
        .login(auth)
        .pipe(
          map(user => new LoginSuccess({ user })),
          catchError(error => of(new LoginFailure(error)))
        )
    ),
  );
```