**Sequence Diagram**

participant User

participant LoginComponent

participant AuthStore (NgRx)

participant AuthEffects

participant Router

participant HomeComponent

participant ProductStore (NgRx)

participant ProductEffects

participant ProductService

participant FakeStoreAPI

User ->> LoginComponent: Fill login form and submit

LoginComponent ->> AuthStore: Dispatch login action

AuthStore ->> AuthEffects: Handle login action

AuthService -->> AuthEffects: Return user/token (mocked)

AuthEffects ->> localStorage: Save token

AuthEffects ->> AuthStore: Dispatch loginSuccess

AuthEffects ->> Router: Navigate to /home

Router ->> AuthGuard: Check auth token in ngrx

AuthGuard -->> Router: Allow navigation

Router ->> HomeComponent: Load HomeComponent

HomeComponent ->> ProductStore: Dispatch loadProducts

ProductStore ->> ProductEffects: Handle loadProducts

ProductEffects ->> ProductService: Call fetchProducts()

ProductService ->> FakeStoreAPI: GET /products

FakeStoreAPI -->> ProductService: Return product list

ProductService -->> ProductEffects: Pass product data

ProductEffects ->> ProductStore: Dispatch loadProductsSuccess

ProductStore -->> HomeComponent: Products loaded via selector

HomeComponent ->> User: Display products

### **Summary of Flow:**

1. **User logs in** through the LoginComponent.
2. **NgRx dispatches login**, handled by AuthEffects,
3. On success:  
   * User data is stored.
   * loginSuccess is dispatched.
   * Redirect to /home.
4. **Route Guard** checks if token exists.
5. **HomeComponent** dispatches loadProducts.
6. ProductEffects calls the fake store API.
7. On success, product data is stored and shown to the user.