Authentication Guards

ตอนนี้เราสามารถเข้าได้ทุกเพจได้ทุกเวลา ซึ่งนี้ไม่ใช่สิ่งที่ถูกต้องเสมอไป

- ผู้ใช้อาจจะไม่ถูกอนุญาตให้เข้าดู Component นี้
- ผู้ใช้อาจจะต้อง Login ก่อน
- จำเป็นต้องมีข้อมูลก่อนถึงจะแสดงผล Component นี้
- อาจจะต้องบันทึกการเปลี่ยนก่อนออกจาก Component นี้ (Auto-save)
- หรือแม้กระทั่งมีปุ่มให้ยืนยันก่อนออกจาก Component นี้ (พิมพ์ข้อความในเฟสแต่ยังไม่ได้กดโพสต์แล้วพยายามจะคลิกไปหน้าอื่น เฟสก็จะมีข้อความแจ้งเตือน)

ในสถานการณ์เหล่านี้เราจำเป็นต้องใช้ Guard ร่วมกับ Route ใน Angular โดยค่าที่ส่งออกมาจาก Guard จะส่งผลกระทบให้กับ Route ดังนี้

- ถ้า Return true ผู้ใช้สามารถเข้าถึง Component ได้
- ถ้า Return false การเข้าถึง Component นั้นจะถูกหยุด และไม่ทำอะไร (ยกเว้นว่าเราจะสั่งให้ Route ส่งผู้ใช้ไปเพจอื่นแทน)
- ถ้า Return UrlTree การเข้าถึงจะถูกหยุดแล้วไปใช้ UrlTree แทน

Route สามารถเรียกใช้งาน Interface ใน Guard ได้ดังนี้

- CanActivate ใช้ตัดสินใจว่า Route นั้นสามารถเข้าถึงได้หรือไม่
- CanActivateChild ใช้ตัดสินใจว่า Route Child นั้นสามารถเข้าถึงได้หรือไม่
- CanDeactivate ใช้ตัดสินใจว่า Route นั้นควรปิดหรือไม่

สำหรับในแลปนี้เราสนใจแค่ CanActivate ที่มักจะทำมาใช้ร่วมกับ Authentication โดยเราควบคุมสิทธิการเข้าถึงไม่ว่าผู้ใช้จะ Login หรือมีสถานะที่กำหนดไว้ เริ่มต้นด้วยการสร้าง Guard

1. ใช้ Angular Generator สร้าง Guard ตั้งชื่อว่า auth
2. ตั้งค่าตัวแปร isLoggedIn ประเภท Boolean ขึ้นมา

3. ในฟังก์ชั่น canActivate ให้เปลี่ยน return true; มาเป็น return this.isLoggedIn
4. Import AngularFireAuth จาก @angular/fire/auth
5. สร้างฟังก์ชั่น constructor แล้ว inject ตัวแปรประเภท AngularFireAuth
6. ในฟังก์ชั่น constructor ให้ผูกค่า this.isLoggedIn ไว้กับข้อมูล Auth จาก AngularFireAuth ผ่านตัวแปร authState ด้วยการ Subscribe

```
this.afAuth.authState.subscribe((user) => {
  if(user) {
    this.isLoggedIn = true;
  } else {
    this.isLoggedIn = false;
  }
});
```

หลังจากที่เราสร้าง Auth Guard เรียบร้อยแล้ว เราจะทำการเพิ่ม canActivate: [AuthGuard] ไปยัง path ที่เกี่ยวข้อง ในที่นี้คือ ‘tweet’

1. ไปยัง app.module.ts แล้ว Import AuthGuard จาก ./auth.guard (หรือ path ไปยัง auth.guard.ts ที่สร้างขึ้นมา)
2. แก้ไข path ‘tweet’ เป็น {path:'tweet',component:AddTweetComponent,canActivate:[AuthGuard]}

ทีนี้เราลองเข้าไปยังหน้าเพจเราแล้วเติม /tweet เราจะพบว่าเมนูโหลดมาแต่ไม่มี Component อะไรโหลดเลย นั่นแสดงว่า AuthGuard ได้บล็อกการเข้าดูของเพจ /tweet สำเร็จ ทีนี้ลอง Login แล้วเข้าในยังเพจ /tweet ด้วยวิธีการพิมพ์ URL เราจะพบว่า Add Tweet Component แสดงผลได้ถูกต้อง แต่การไม่แสดง Component อะไรเลยเมื่อผู้ใช้ไม่ได้ Login อาจจะทำให้เกิดความสับสนว่าเว็บแอพเราไม่สามารถใช้งานได้ ดังนั้นแทนที่จะบล็อกการเข้าถึงอย่างเดียวเราจึงจำเป็นต้อง Route ผู้ใช้ไปยังหน้าที่เหมาะสมด้วย เราสามารถทำได้โดยการแก้ไขการ return ค่าจาก AuthGuard ให้มีการ Route ไปยังเพจแรกได้ดังนี้

1. Import Router จาก @angular/router
2. Inject ตัวแปรประเภท Router ไว้ใน constructor

3. แก้ไขการ Return ตัวแปรของฟังก์ชั่น canActivate มาเป็นการ Return ฟังก์ชั่นใหม่
4. ในฟังก์ชั่นใหม่นั้นให้ Return ค่า true false เหมือนเดิม แต่ก่อนหน้านั้นให้มีการเช็คสถานะ Login ถ้าไม่ได้ Login ให้เรียกฟังก์ชั่น navigate ของตัวแปรที่ประกาศไว้ในข้อ 2.