**PreAuthorizeAuthorizationManager и PostAuthorizeAuthorizationManager** позволяют добавлять проверку прав доступа (авторизация) с помощью аннотаций **@PreAuthorize и @PostAuthorize.**

**@PreAuthorize – проверяет права доступа ДО.**

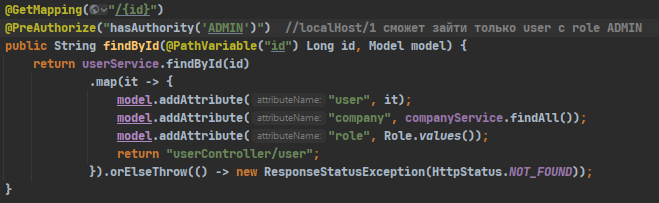
**@PostAuthorize – проверяет права доступа ПОСЛЕ. @PostAuthorize("returnObject")**

Отличие @PreAuthorize от @PostAuthorize в том, что в случае @PostAuthorize у нас есть доступ к возвращаемому значение.

SecurityExpressionRoot – класс, который предоставляет ряд методов для работы с авторизацией. К примеру, SecurityExpressionRoot содержит метод getPrincipal (), который вернет нам пользователя. Или getAuthoritySet (), который вернет все роли.

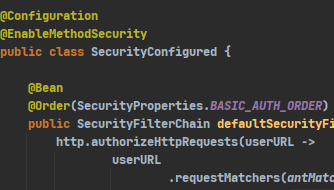
Для SecurityExpressionRoot реализации нам нужно поставить аннотацию **@PreAuthorize("hasAuthority('ADMIN')")** при этом, в этой аннотации мы можем писать и логические операции **@PreAuthorize("hasAuthority('ADMIN') and ").**

**Реализация:**

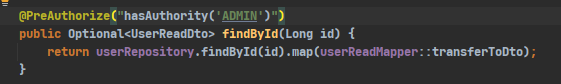
****

****

**2. Поставить аннотация @EnableMethodSecurity над классом securityConfigure**

****

**Так же, мы можем ставить аннотацию и в нашем слое service**

****