

Server Programming Auditing

Juha Hinkula



Auditing

- Database auditing means logging the persistent entities. In the business applications we are often interested in who created the record and when.
- JPA auditing allows you to record who, what and when for each entity object.
- To enable JPA auditing in existing Spring Boot project (with JPA + Spring Security) you need following steps



1.) Add following annotations to entity classes

`@Entity`

`@EntityListeners(AuditingEntityListener.class)`

`public class Todo {`

`@Id`

`@GeneratedValue(strategy=GenerationType.AUT)`

`private long id;`

`private String description, status;`

`@CreateDate`

`private Date createDate;`

`@CreatedBy`

`private String createdBy;`

`@LastModifiedDate`

`private Date lastModifiedDate;`

`@LastModifiedBy`

`private String lastModifiedBy;`



Auditing

- `@CreatedBy`, `@CreatedDate`, `@ModifiedBy` and `@ModifiedDate` are used to annotate entity attributes that records auditing data.
- Specify `AuditingEntityListener` with `@EntityListeners` annotation



Haaga-Helia

1.) Create auditing configuration class

```
@Configuration
@EnableJpaAuditing
class AuditConfig {
    @Bean
    public AuditorAware<String> createAuditorProvider() {
        return new SecurityAuditor();
    }

    @Bean
    public AuditingEntityListener createAuditingListener() {
        return new AuditingEntityListener();
    }

    public static class SecurityAuditor implements AuditorAware<String> {
        @Override
        public String getCurrentAuditor() {
            Authentication auth = SecurityContextHolder.getContext().getAuthentication();
            String username = auth.getName();
            return username;
        }
    }
}
```

