

Un poco de Java y +

Otra forma de hablar de nuestro día a día...

JAVA, SPRING

Auditoría de Entidades JPA con Spring Boot

Date: 8 enero 2018 **Author:** Luis Miguel Gracia ☐ 0 Comentarios

Spring Data incluye un conjunto de anotaciones que me permiten auditar las inserciones y modificaciones sobre mis entidades JPA.

Así anotando la clase con **@EntityListeners** y sobre un atributo de tipo Date o long con **@CreatedDate** o **@LastModifiedDate**

```
@MappedSuperclass
@JsonIgnoreProperties(value = {"createdAt", "updatedAt"},
    allowGetters = true)
@EntityListeners(AuditingEntityListener.class)
@ToString
public abstract class AuditableEntity implements Serializable {

    private static final long serialVersionUID = 1L;

    @Column(nullable = false, updatable = false)
    @Temporal(TemporalType.TIMESTAMP)
    @CreatedDate
    @Getter @Setter private Date createdAt;

    @Column(nullable = false)
    @Temporal(TemporalType.TIMESTAMP)
    @LastModifiedDate
    @Getter @Setter private Date updatedAt;
```

Automáticamente mi PersistenceManager incluirá la fecha de creación del registro o de la última modificación.

Si además tengo integrado Spring Security en mi aplicación también podré auditar el usuario que creó o modificó el registro con las anotaciones **@CreatedBy** y **@LastModifiedBy**

```
@Entity
@EntityListeners(AuditingEntityListener.class)
public class Bar {

    //...

    @Column(name = "created_by")
    @CreatedBy
    private String createdBy;
```

Si mi persistencia está en un proyecto Spring Boot lo único que me queda es anotar mi punto de entrada a la aplicación Spring Boot (mi `SpringBootApplication`), con la anotación **`@EnableJpaAuditing`**:

```
@SpringBootApplication
@EnableJpaAuditing
public class ConfigDbEntitiesApplication {

    public static void main(String[] args) {
        SpringApplication.run(ConfigDbEntitiesApplication.class, args);
    }
}
```

© 2018 UN POCO DE JAVA Y +

