Hibernate & Spring Data JPA

Beginner to Guru

Hibernate Interceptors and Listeners

Hibernate Interceptors and Listeners

- Interceptors and Listeners can be used together
    - Significant overlap in functionality
- Both will be applied to all entities
- Listeners should be considered stateless
    - Shared between requests, and should NOT save state

SPRING FRAMEWORK GURU
2022

Hibernate Interceptors

- Hibernate Interceptors are callbacks registered in Hibernate’s internal operations
  - Can be session scoped or session factory scoped
  - Can be used for security, auditing, or to alter data
  - Implemented via:
    - org.hibernate.Interceptor interface
    - org.hibernate.EmptyInterceptor class
      - Deprecated in Hibernate 6.x - migrating to default interface methods

Hibernate Listeners

- Hibernate has an event system
- Hibernate event listeners subscribe to Hibernate Events
- Hibernate Events are defined by org.hibernate.event.spi.EventType
    - Roughly 36 Event types defined - SAVE, PRE_INSERT, POST_INSERT, etc
- Each type defines a default listener, which can be extended to define a custom listener
- Once created, the customer Listener must be registered with the Hibernate SessionFactory

JPA Callbacks

- JPA defines 7 Callbacks via annotations
- Callbacks are entity specific and can be:
  - On an entity method
  - Or on a Entity Listener Class
    - Entity Listeners must:
      - Be stateless and have no-arg constructor
      - Methods must return void
      - Do not use Entity Manager

SPRING FRAMEWORK GURU 2022

JPA Callbacks

- **@PrePersist** - Before persist operation
- **@PreRemove** - Before remove operation
- **@PostPersist** - After persist operation is completed
- **@PostRemove** - After remove operation is completed
- **@PreUpdate** - Before update operation
- **@PostUpdate** - After update operation
- **@PostLoad** - After entity is loaded or refreshed

SPRING FRAMEWORK GURU 2022

SPRING FRAMEWORK

GURU