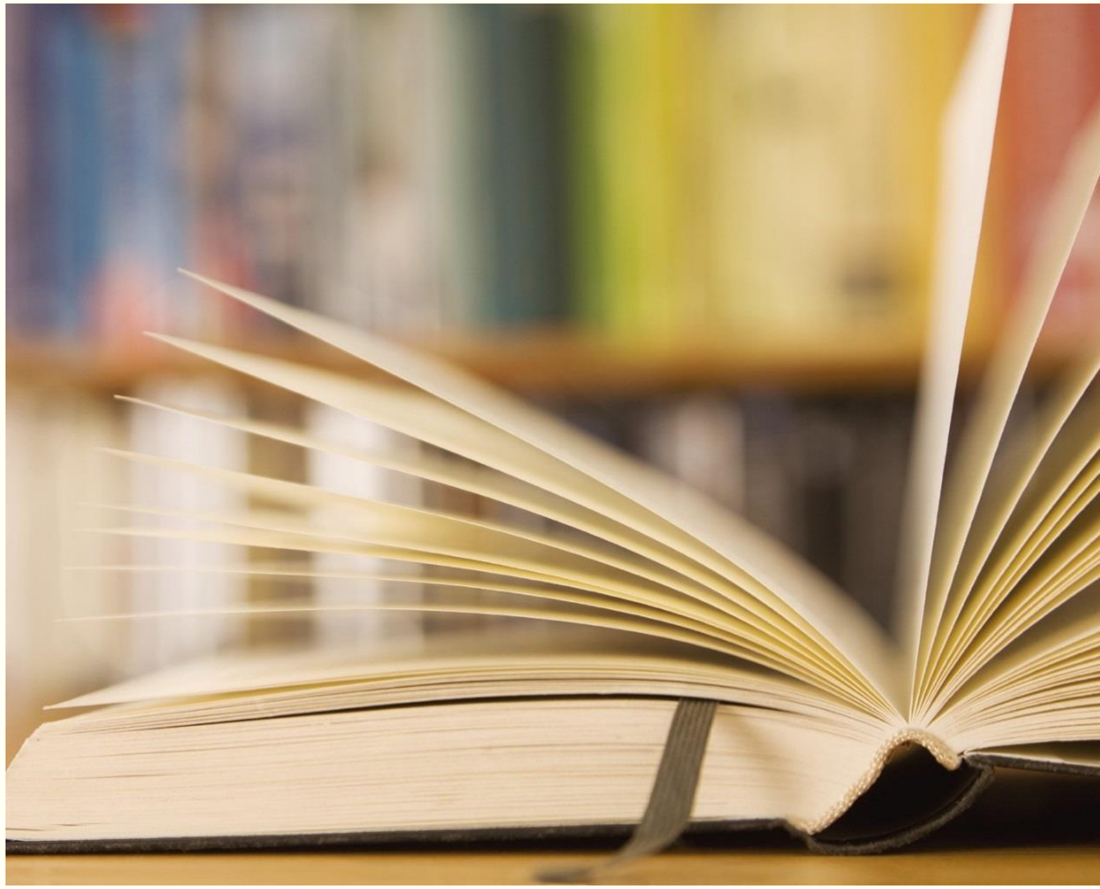
ENTERPRISE ARCHITECTURE

Najeeb Najeeb, PhD

Version 2.1 ©2021



INVERSION OF CONTROL

Inversion of Control (IoC)

Library

- The application controls the flow.
- The library is used to perform tasks.
- You write code to do everything, but if you need help with something you invoke the library.
- Your code has full control.

Framework

- The framework controls the flow.
- The framework may perform tasks or invoke libraries to help.
- At some points you may attach your code to tweak things or do something.
- You relinquish control to the framework. It may allow you some roles to play sometimes.

Dependency Inversion Principle

- Relay on abstractions rather than concrete implementations.
  - P2I
- Why?
  - Flexibility.
  - Testability.
  - Extendibility.
  - Open-Closed principle.
- DI is an example of IoC.

EVENT CALLBACKS

Events

- Persist
  - Post & Pre
  - persist(), merge()
  - Cascading?
  - Order not guaranteed.
  - Post does not guarantee success.
- Update
  - Post & Pre
  - Pre some time before the DB update.
  - Pre not specified in the spec.
  - Post after the DB update.
  - Post does not guarantee success.
- Remove
  - Post & Pre
  - remove()
  - SQL DELETE
  - Post does not guarantee success.
- Load
  - Post
  - When reading entity.
  - When lazy loading.
  - When refresh()
  - Order not guaranteed.

Callback Methods

- Linked to an event.
- Special signature
  - void method();
- Cannot be final or static.
- No checked exceptions.
- Can have runtime exceptions.
- Method can be for several events.
- Singel method per event per entity.
- Should not use em in callback method.
- Should not run queries in method.
- Example:
  - Keep track of last time an entity was synchronized with DB.
  - Log DB updates.
  - …

Entity Listener

- Non-entity class listening to events.
- Entity listener is stateless.
- Signature
  - void method(Object o);
  - void method(Entity e);
  - void method(EntityParent e);
  - void method(EntityInterface e);

```
@Entity
@EntityListeners({EntityListener.class})
public class Student {
    @Id private int id;
    private String name;
    @Transient private long lastSync;
}

public class EntityListener {
    @PrePersist
    public void doThings(Student student
       {
              Log.write("Persist student with id :
" + student.getId();
       }
}
```

Main Point

- IoC relinquishes control from the application code to the framework. The use of callback methods enable application code to still be able to perform activity in between framework operations.