**Why this problem?**

This problem represents typical and generic scenario in enterprise level applications across various domains. Although on the face of it, it looks like straightforward problem, but it has tricky scenarios like time overlap, dealing with dates, database polling as a part of solution. I choose this to showcase my abilities for API design, writing production ready, maintainable, extensible and readable code.

**Design and Approach**

Web Oriented architecture is used to provide basic CRUD operations to the claim user. For database polling operation, solution has been implemented using spring integration. Application has been designed keeping SOLID principles in mind.

**Webservice Architecture**

**H2 in memory db**

**Persistence layer (JPA)**

Business Layer

Spring service

**REST API**

Browser

**Data Integration architecture**

DB

Inbound channel

**Patterns used**

* Designed by using WOA (web oriented architecture)
* DAO Pattern (data access object) is used to separate low level data accessing API or operations from high level business services (spring data repositories)
* DTO Pattern (Data transfer object) this has been used to decouple presentation layer object and entity object. We can then control what and how much data we want to transfer and expose to clients.
* DI (dependency injection ) spring beans