**Más Stereotypes**

MÁS DE UNA IMPLEMENTACIÓN DE UNA INTERFACE

**@Qualifer**: Algunas veces se necesita tener más de una implementación de una interface.

* Para el ejemplo, se va a colocar dos implementaciones para el repositorio (PersonaRepoImpl1, PersonaRepoImpl2).
* En este momento van a haber dos clases que implementan la interface, por lo que cuando se cree la instancia para la inyección de dependencias, el no sabrá cual de las dos implementaciones pertenece a esta. Y Saldrá este error

Consider marking one of the beans as @Primary, updating the consumer to accept **multiple beans**, or using @Qualifier to identify the bean that should be consumed

* Para solucionar esto, lo que se hace es utilizar el stereotype @Qualifer(“nombre instancia”), donde en los paréntesis se engancha el mismo nombre en la clase donde se crea la instancia y la clase a la que quiere que implemente la interface. Para este caso se coloca una en PersonaServiceImpl y el mismo stereotype incluyendo el enganche en la clase que se implementa, es decir por ejemplo rn PersonaRepoImpl1:

@Service

**public** **class** PersonaServiceImpl **implements** IPersonaService{

**private** **static** Logger *LOG*= LoggerFactory.*getLogger*(PersonaServiceImpl.**class**);

@Autowired

@Qualifier("persona1")

**private** IPersonaRepo repo;

@Override

**public** **void** registrar(String nombre) {

repo.registrar(nombre);

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_-

@Repository

@Qualifier("persona1")

**public** **class** PersonaRepoImpl1 **implements** IPersonaRepo{

**private** **static** Logger *LOG*= LoggerFactory.*getLogger*(PersonaRepoImpl1.**class**);

@Override

**public** **void** registrar(String nombre) {

// **TODO** Auto-generated method stub

*LOG*.info("Se registró la persona: PersonaRepoImpl"+nombre);

}

}

* De esta manera la aplicación sabrá a cual interface pertenece la implementación, y a cual implementación pertenece la instancia de la inyección de dependencias.
* Para cambiar de implementación, entonces iríamos a donde se esta creando la instancia de la inyección y llamaríamos en el qualifier a la otra implementación por mdeio del enganche

@Service

**public** **class** PersonaServiceImpl **implements** IPersonaService{

**private** **static** Logger *LOG*= LoggerFactory.*getLogger*(PersonaServiceImpl.**class**);

@Autowired

@Qualifier("persona2")

**private** IPersonaRepo repo;

@Override

**public** **void** registrar(String nombre) {

repo.registrar(nombre);

}

}

* Esto se podría aplicar, por ejemplo si se requiriera cambiar de framework, se realiza la implementación nueva y se le indica ya a la instancia de la inyección de dependencias que se enganche a esa implementación, de resto quedaría normal, es decir no se cambiarían el resto de clases, evitando así escribir y eliminar código.