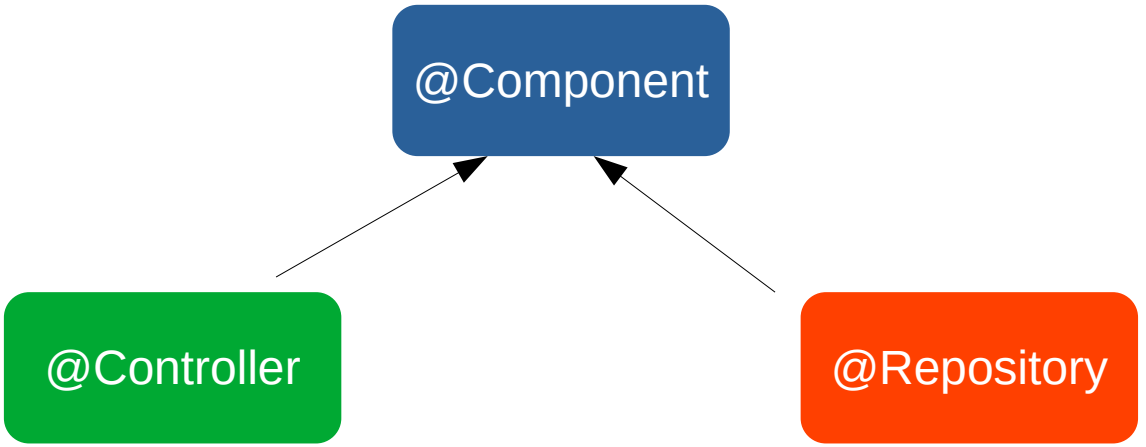
@Repository

Spring dao annotation

- Spring fournit d'autres annotations , en particulier pour les Controllers et les implémentations de DAO



Une annotation spécialisée pour une implémentation de DAO

- @Repository s'applique sur une implémentation de DAO
- Spring enregistre automatiquement l'implémentation dans le contexte grâce au component scaning
- Spring fournira la traduction de toutes les exceptions liées à JDBC (checked exception) traduites en unchecked exceptions.

```java
@Repository
public class UserDaoImpl implements UserDao{

    @Autowired
    private SessionFactory sessionFactory;

    @Override
    @Transactional
    public List<User> getUsers() {
        // récupérer une session hibernate
        Session session = sessionFactory.getCurrentSession();

        // créer une requête
        Query<User> query= session.createQuery("from User",    User.class);

        //executer la requête , récupérer son résultat
        List<User> users = query.getResultList();

        return users;
    }

}
```

2. Réaliser la couche DAO

A présent vous pouvez incorporer la couche Dao

- Créer le package,
- Créer l'interface
- Créer l'implémentation (@Repository)
  - Injecter la sessionFactory (@Autowired)
  - Créer la méthode getUsers,

    (@Transational, session.createQuery,return list...)

JUnit 5

- A ce stade on devrait réaliser des tests unitaires
- Mais ce n'est pas dans le plan de ce cours
- Je vous recommande de les réaliser à chaque développement sans attendre d'avoir trop de composants dans votre application à tester
- Cela permet de fiabiliser vos developpements au fur et à mesure