The difference between @Inject vs. @Autowire vs. @Resource?

**@Autowired:** spring propriety annotation (as opposed to @Inject and @Resource) that inject a resource by-type, i.e. by the class of by the interface of the annotated field or contractor. In case we have few implementation of an interface or a subclass we can narrow down the selection using the @Qualifier annotation to avoid ambiguity. For a fallback match, the bean name is considered a default qualifier value. Although you can use this convention to refer to specific beans by name, @Autowired is fundamentally about type-driven injection with optional semantic qualifiers.

**@Inject:** Annotation based on JSR-330 (Dependency Injection for Java) identifies injectable constructors, methods, and fields. This annotation is an almost complete drop-in replacement for Spring’s @Autowired annotation. So, instead of using the Spring-specific @Autowired annotation, you might choose to use @Inject. One of the differences between @Autowired and @Inject is that @Inject does not have the required field so in case we fail to find a suitable object to injected it will fail while @Autowired can used required=false and allow null able field (only if required!). Advantage of @Inject annotation is that rather than inject a reference directly, you could ask @Inject to inject a Provider. The Provider interface enables, among other things, lazy injection of bean references and injection of multiple instances of a bean. In case we have few implementation of an interface or a subclass we can narrow down the selection using the @Named annotation to avoid ambiguity. @Named annotation works much like Spring’s @Qualifier

**@Resource:** annotation based on JSR-250. @Resource is quite similar to @Autowired and @Inject, but the main difference is the execution paths taken to find out the required bean to inject. @Resource will narrow down the search first by name then by type and finally by Qualifiers (ignored if match is found by name). @Autowired and @Inject will narrow down the search first by type then by qualifier and finally by the name.