@Inject

In Spring, the `@Inject` annotation is used to inject dependencies into Spring-managed beans. It is similar to the `@Autowired` annotation and is part of the Java Dependency Injection (DI) API.

Here's how `@Inject` works in Spring:

1. **Dependency Injection**: You can use `@Inject` to inject dependencies into beans, just like `@Autowired`. For example:

```java
import javax.inject.Inject;
import org.springframework.stereotype.Component;

@Component
public class MyComponent {

    private MyDependency dependency;

    @Inject
    public MyComponent(MyDependency dependency) {
        this.dependency = dependency;
    }

    // Other methods using dependency
}
```

2. **Constructor Injection**: `@Inject` can be applied to constructors to indicate that Spring should use constructor injection to instantiate beans. In the example above, the constructor is annotated with `@Inject`, indicating that Spring should inject the `MyDependency` bean into the constructor.

3. **Field Injection**: You can also use `@Inject` on fields to perform field injection:

```java
import javax.inject.Inject;
import org.springframework.stereotype.Component;

@Component
public class MyComponent {

    @Inject
    private MyDependency dependency;
```

```
    // Other methods using dependency
}
```

4. **Setter Injection**: Similarly, you can use `@Inject` on setter methods to perform setter injection.

```java
import javax.inject.Inject;
import org.springframework.stereotype.Component;

@Component
public class MyComponent {

    private MyDependency dependency;

    @Inject
    public void setDependency(MyDependency dependency) {
        this.dependency = dependency;
    }

    // Other methods using dependency
}
```

5. **Bean Resolution**: Spring resolves dependencies based on the type of the dependency. If there is only one bean of a particular type in the application context, Spring will inject it. If there are multiple beans of the same type, you can use `@Qualifier` along with `@Inject` to specify which bean to inject.

Overall, `@Inject` provides a standardized way to perform dependency injection in Spring applications, aligning with the Java EE specifications. However, in most Spring applications, developers commonly use `@Autowired` for dependency injection due to its native support within the Spring framework.