Spring 101

Java Frameworks

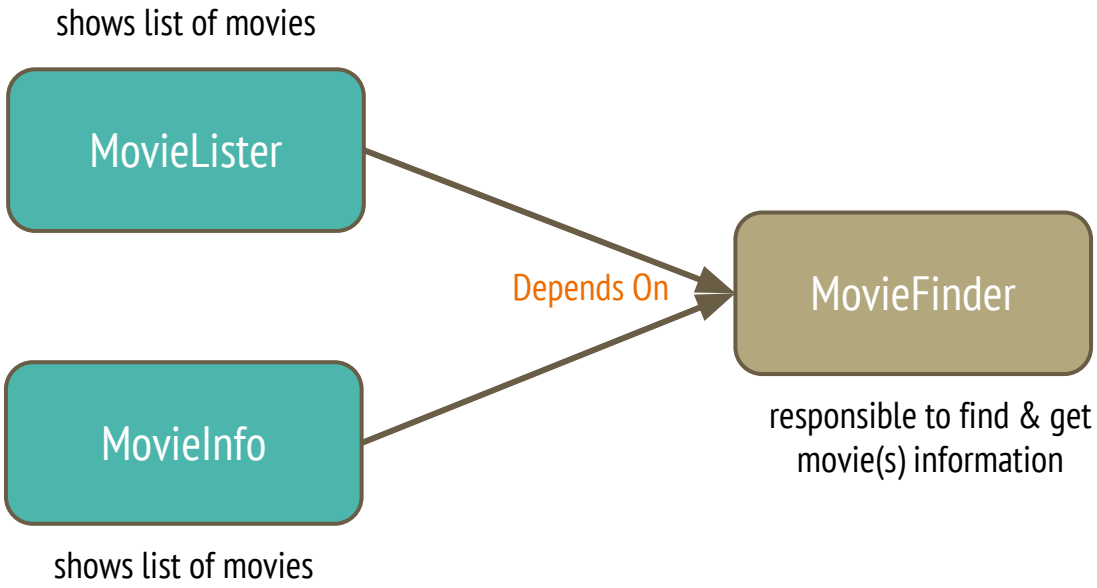
Developed as part of Morning School Training Program @ Citicorp Services India Ltd

Dependency Injection

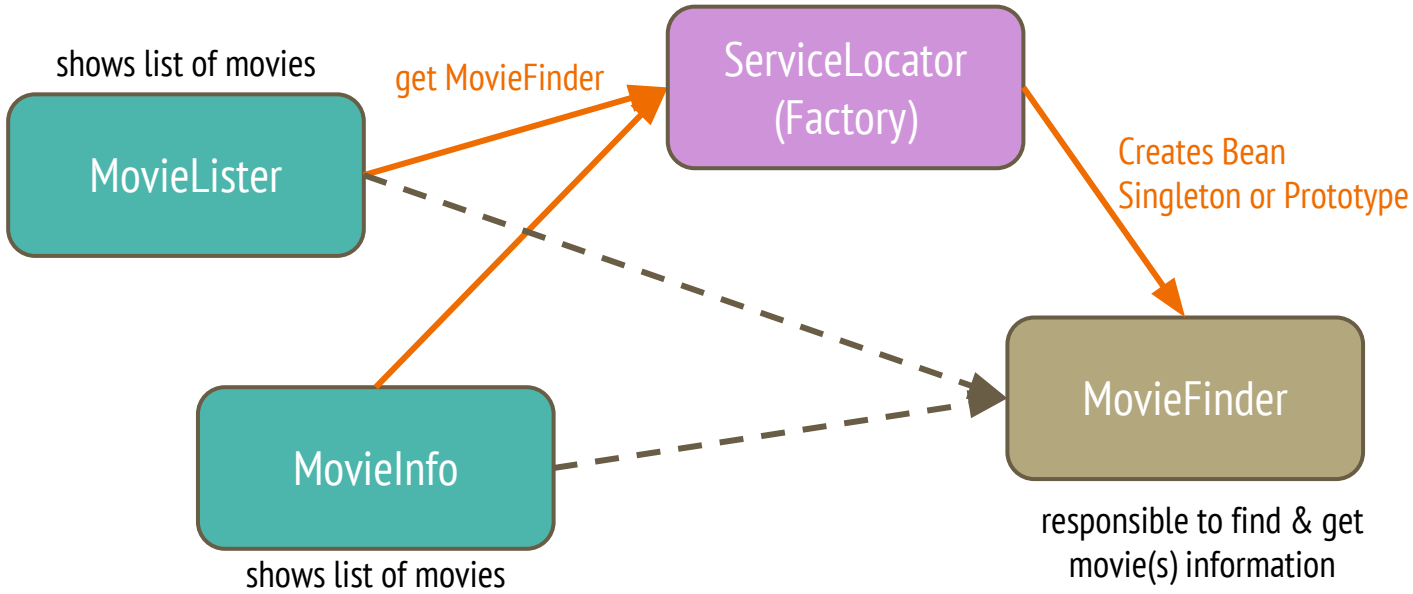
a.k.a Inversion of Control

http://martinfowler.com/articles/injection.html

What is dependency?

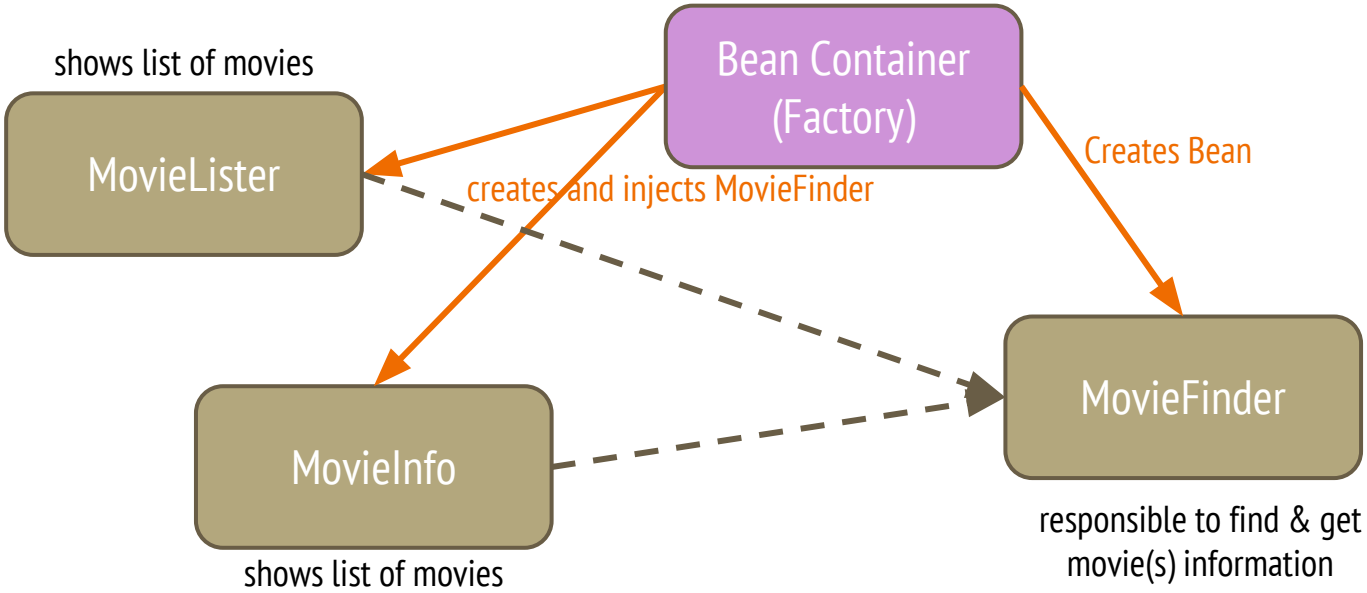


Service Locator / Factory - Design Pattern

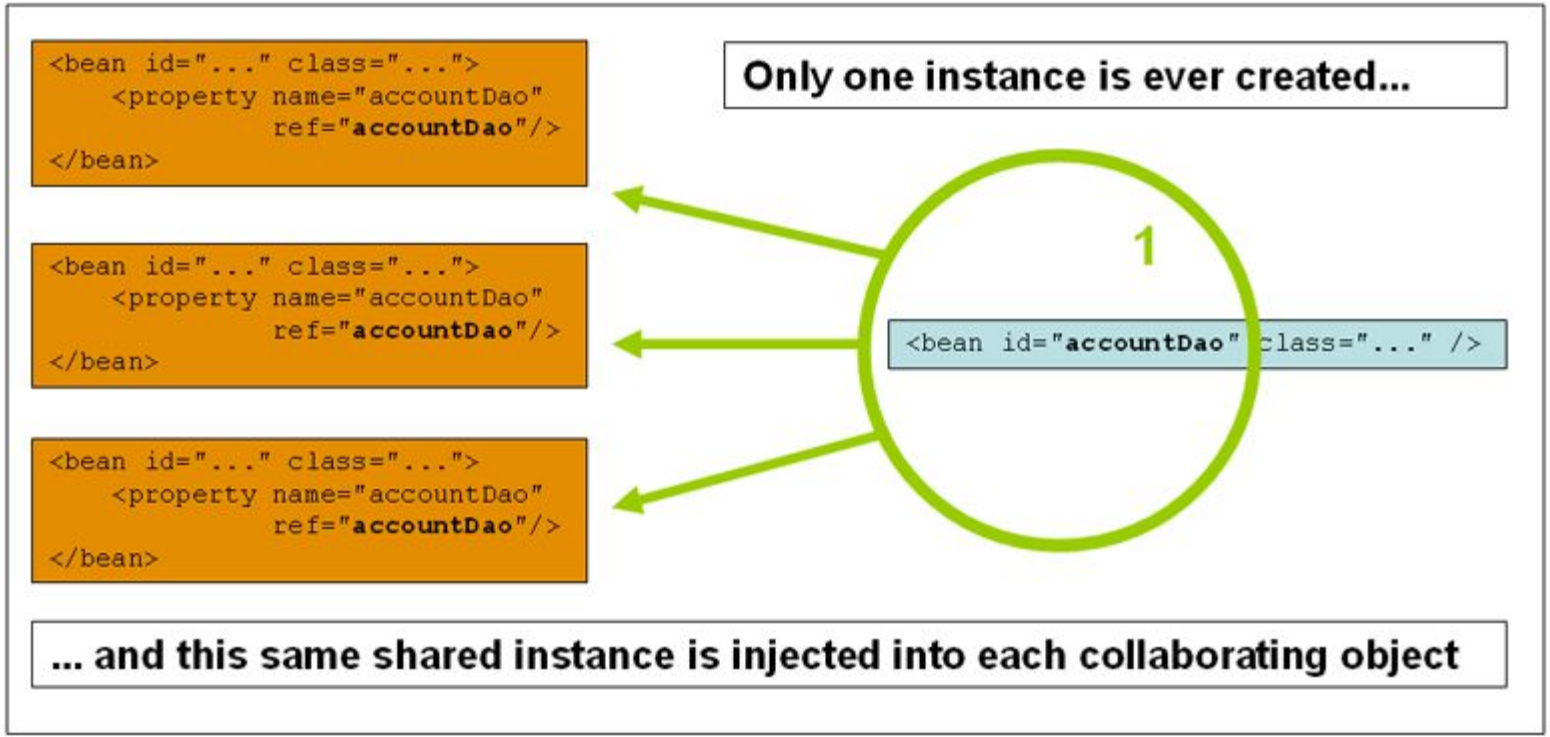


Dependency Injection - Design Pattern

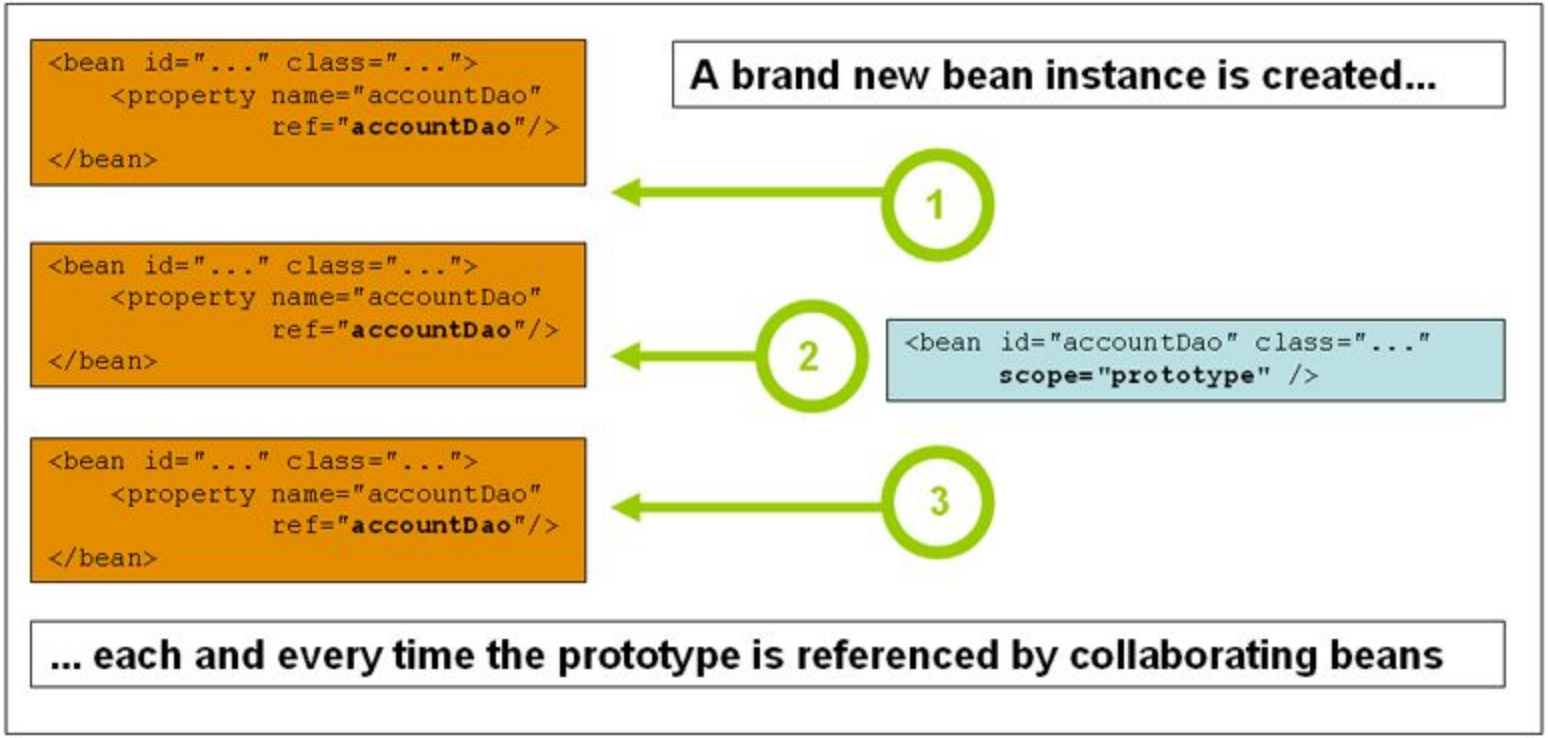
Since approach is inverted from previous, this design pattern was also called initially as inversion of control (IoC).



Singleton - DI



Prototype - DI



Bean Scope

- Singleton
- Prototype
- Request
- Session
- Global Session
- Application

http://docs.spring.io/spring/docs/current/spring-framework-reference/htmlsingle/#beans-factory-scopes

Spring Bean

- Spring as bean container
- Spring Lifecycle @PostConstruct & @PreDestroy
- Spring Bean Scope Singleton & Prototype, Request Scope, Session Scope

XML vs Annotations

- Provides Type Safety
- Bean configuration near to code
- Brings more readability to code

Basic Annotations

@Bean

@AutoWired, @Inject & @Resource

@Component & @Service

@Repository

@Configuration

https://dzone.com/refcardz/spring-annotations

@Bean

```
@Configuration
public class MovieConfiguration {

    @Bean
    @Primary
    public MovieCatalog firstMovieCatalog() { ... }

    @Bean
    public MovieCatalog secondMovieCatalog() { ... }
}
```

@AutoWired

http://www.javabeat.net/difference-resource-autowired-inject-spring-injection/

```
public class MovieRecommender {
    @Autowired  // field based injection
    private MovieCatalog movieCatalog;

    @Autowired // all beans of type MovieCatalog
    private MovieCatalog[] movieCatalogs;

    private MovieCatalog movieCatalog;
    @Autowired  // constructor based injection
    public MovieRecommender(MovieCatalog movieCatalog) {
        this.movieCatalog = movieCatalog;
    }
}
```

@Service, @Repository & @Configuration, @ComponentScan

```java
@Service
public class SimpleMovieLister {

    private MovieFinder movieFinder;

    @Autowired
    public SimpleMovieLister(MovieFinder movieFinder) {
        this.movieFinder = movieFinder;
    }
}
```

```java
@Repository
public class JpaMovieFinder implements MovieFinder {
    // ...
}
```

```java
@Configuration
@ComponentScan(basePackages = "org.example")
public class AppConfig  {
    ...
}
```

Future Reading

- Java Brains: Learning Spring Core
  - http://javabrains.koushik.org/courses/spring_core
- http://docs.spring.io/spring/docs/current/spring-framework-reference/htmlsingle/
- http://www.javabeat.net/difference-resource-autowired-inject-spring-injection/
-

Thanks!

Like to reach us,

Sunit Parekh
parekh.sunit@gmail.com

Sarthak Makhija
sarthak.makhija@gmail.com