

Java 8

Classical : Imperative

How

Object mutability

Java 8 : Declarative Style

What we want

Object immutability

Interface:

default method (definition)

static method (definition)

collection api

interface (10 functionalities + 2)
stream

Object Oriented approach : interface

Functional Interface :

Contain only one abstract method

might have static , default method in any count

Lambda:

anonymous function

no method param type, return type

not be encapsulated in any class

can be assigned to a variable of functional interface

```
() -> <single line>
```

```
()->{  
    // multi line  
}
```

```
(msg)->{}
```

```
msg -> {}
```

```
(a,b) -> {}
```

```
(a,b)->{  
    int sum = a+b;  
    return sum;  
}
```

```
(a,b)-> a+b; // by default associated with return statement
```

the method signature of the only abstract method of Functional interface must match with method signature of lambda expression

java.util.function

functional interface containing some very common prototype method

4 categories

Consumer

Predicate

Function

Supplier

Consumer :

void accept(<T>)

Predicate

boolean test(<T>)

Function

<R> apply(<T>);

Supplier

<T> get()

Variants

Consumer : BiConsumer (Generic)

void accept(<T>,<M>)

Primitive type implementation

IntConsumer()

Predicate:

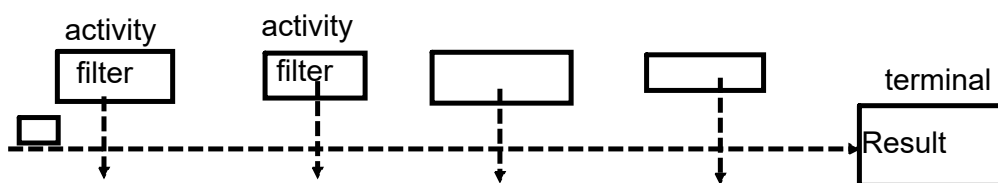
BiPredicate, Primitive type implementation

Function : BiFunction

Functional programming remove overhead of creating objects and loading class files

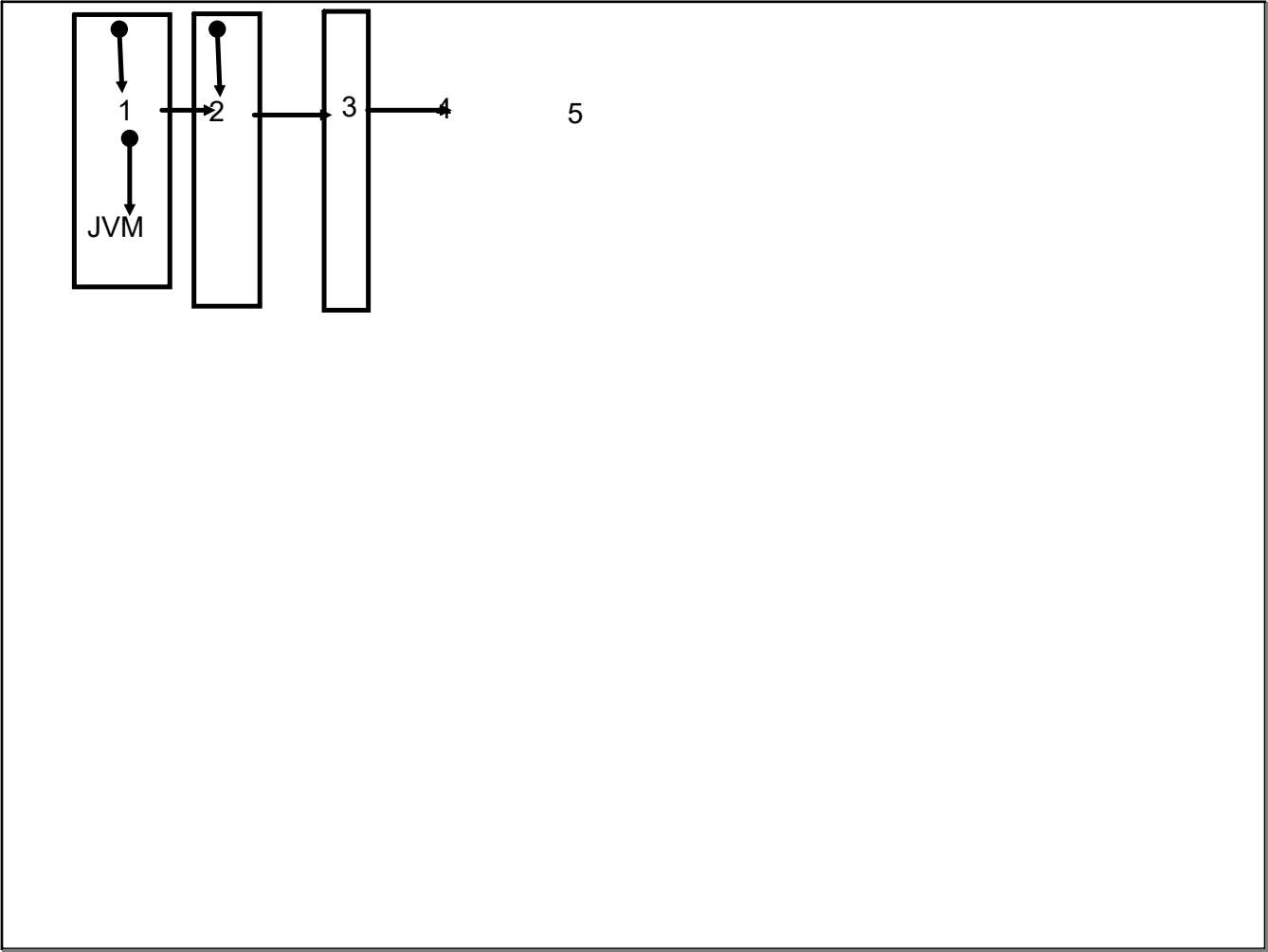


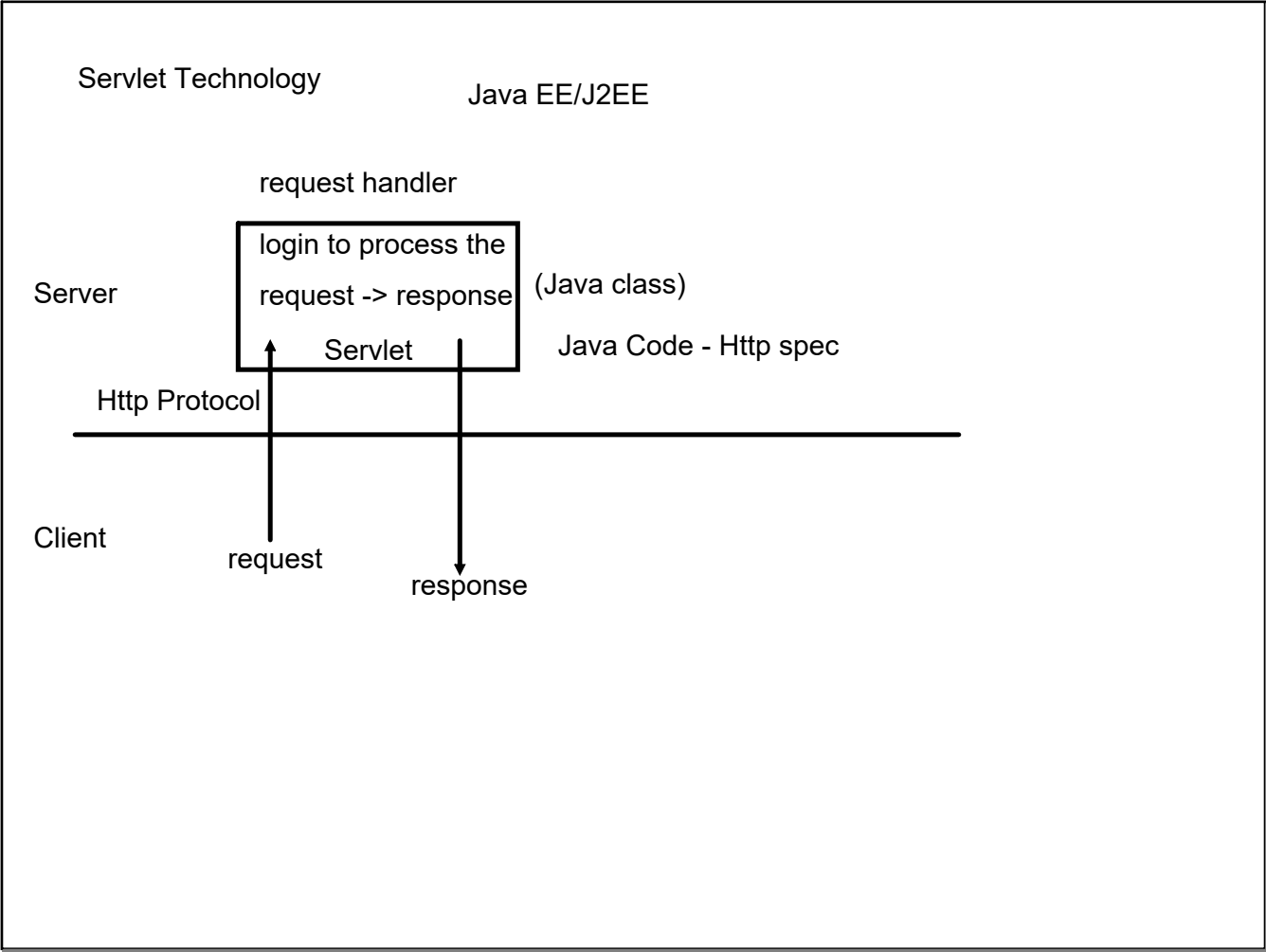
Conveyer belt



Parallel Processing not to be preferred

1. when an external mutable object is involved
2. when the stream activities involve some inherent complexity





Java EE

Servlet

Java Class

```
class MyServ extends GenericServlet/HttpServlet{
```

```
}
```

GenericServlet : Support only generic Http Verb (Form verbs) get/post

HttpServlet : identifies HTTP Verbs (get,post,put,delete)

identifies intention of http verb

```
class MyServ extends GenericServlet/HttpServlet{  
    <---- lifecycle ----->  
    init (load)  
    service(GenericServlet) (doPost,doGet,doPut,doDelete: HttpServlet) : request/response  
    destroy (unloaded)  
}  
  
service/(doPost,doGet,doPut,doDelete) (HttpServletRequest request,  
                                         HttpServletResponse response){  
  
}
```

Spring

Spring Core

Spring MVC (maven)

Spring Boot

Spring Framework : Servlet technology

CORE

Bean Factory

IoC

DI

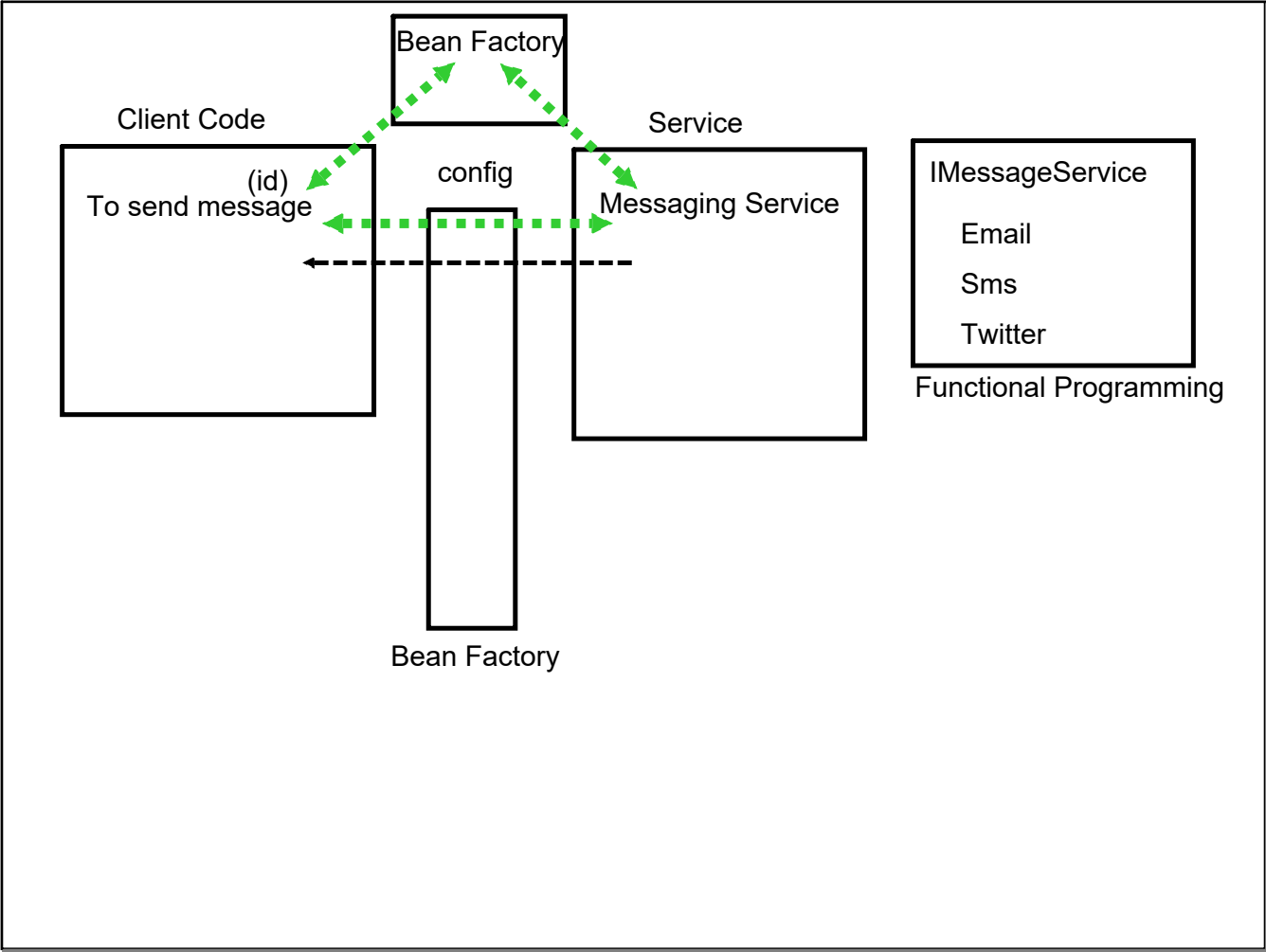
AOP

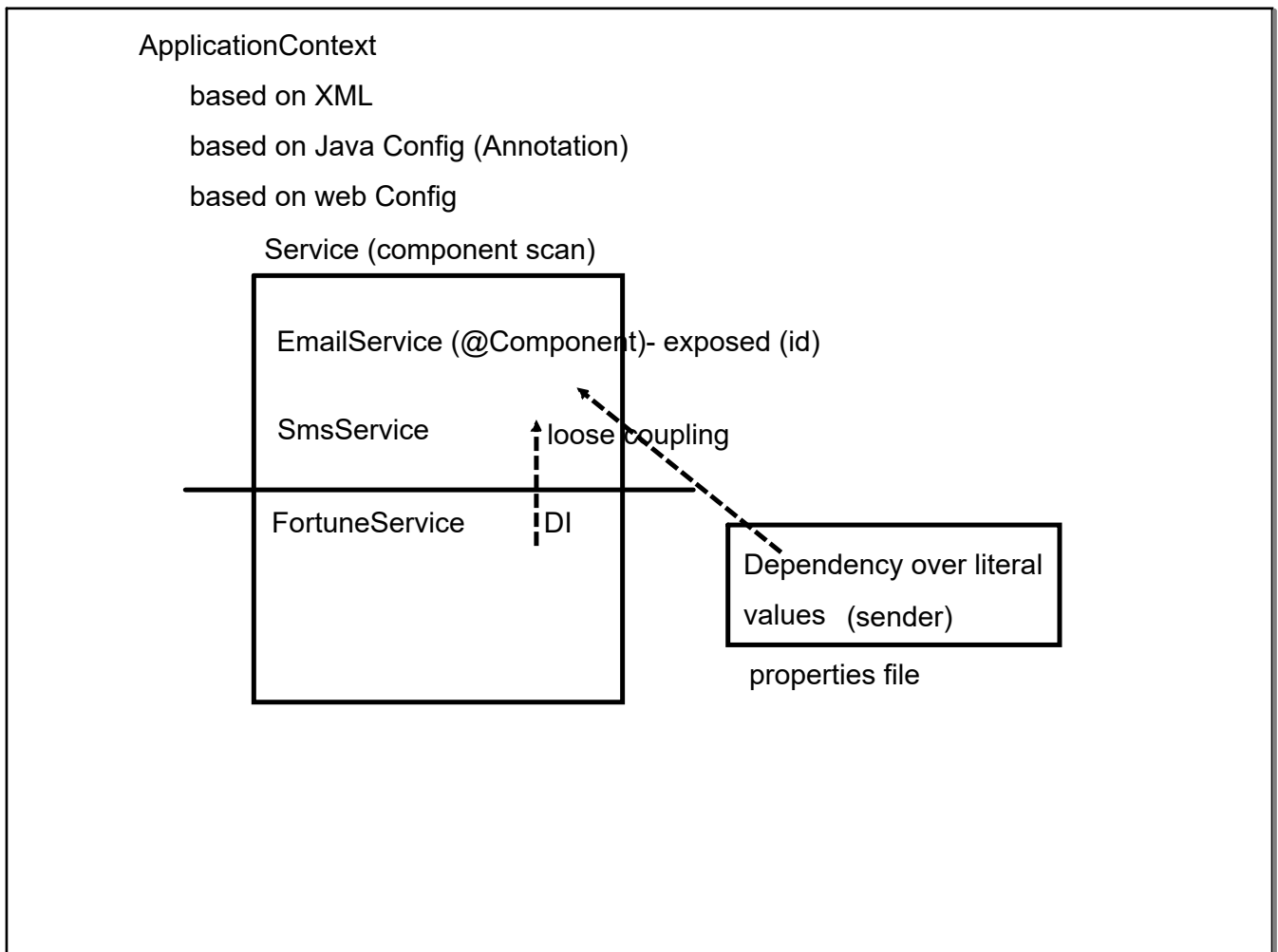
IoC : Outsourcing the creation and management of object

Bean : Java Object managed by container

AOP : Aspect Oriented Programming (Proxy)

Clean, Loosly Coupled, reusable JAva Code



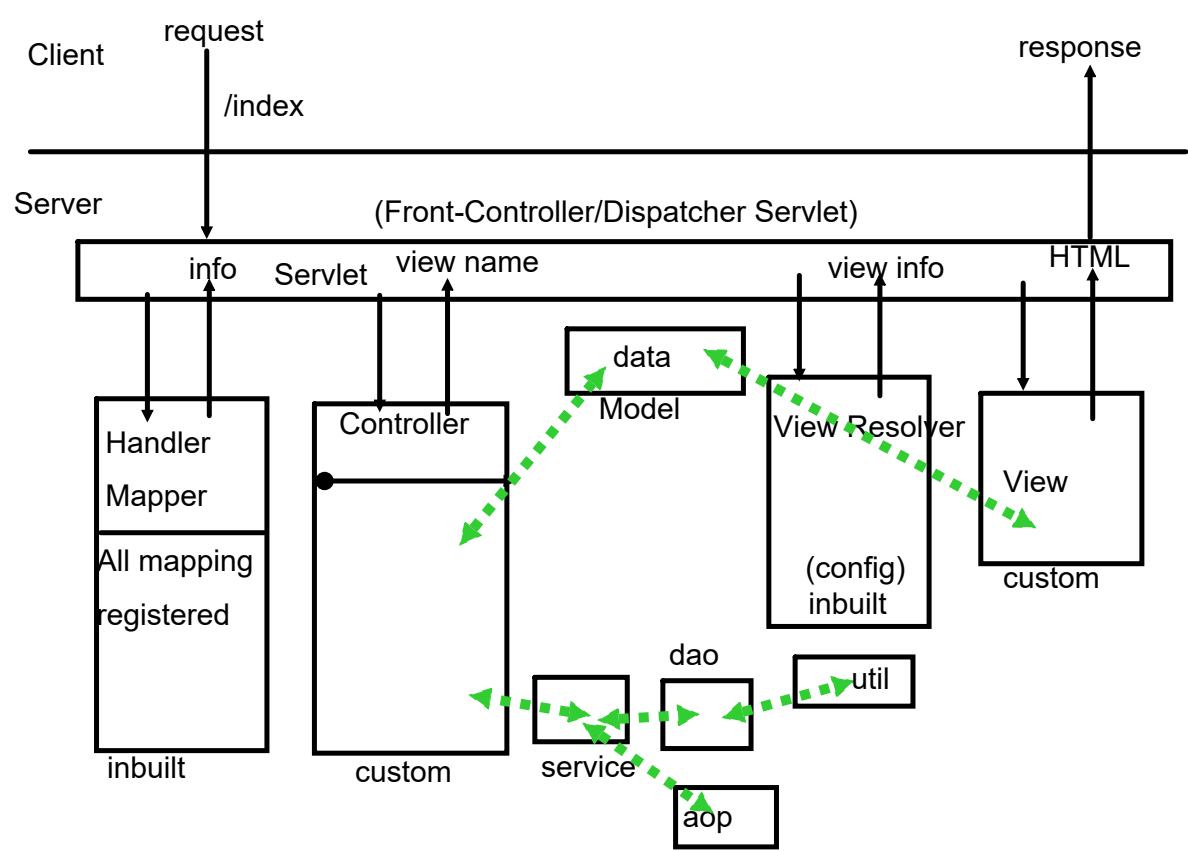


Scope : Singleton (Default)
Prototype

request : single request-response cycle
session : all request-response cycle for a particular user
global : all request-response cycle for all user(web context)

Spring Context does not maintains complete lifecycle of Prototype bean

Servlet Technology being used in Spring MVC framework



Maven:

Dependency Management

Standard folder/file system

build

test

documentation

pom : project object model

all config related to maven activity

web.xml : a must file for servlet config