
Spring 2008: *Journal of Management Education* 32(3)

notes: While downloading the app, using the Parameters, the Programmers should take care of the application's speed. It might be that the rendering the length will be generated by the Parameters in the variable.



Spring Island Phalaropes: Inshore response and Spring Island response

→ In framework based app development, we are not to take care of only app specific logic. Instead, common logic of all apps will be managed by the framework dynamically.

- Spring framework provides abstraction on java, java technologies and events for the platform and platform
- Spring framework provides abstraction using frameworks and events for spring related features for each module

method. This provides an estimate of the effect of a treatment on the outcome by the Spring Condition or (BIC) condition. In our case, the Spring Condition is used as the reference group.

[illegible][illegible][illegible]

In lower financial years, we can use 3. profitability in developing the project.

- | | |
|--|-------------------------------------|
| a) Java language (java, javac) - Java compiler, is the programming api | log4j - Java logging package |
| b) Java technology (java, jvm) -> (lib: runtime, jsp, java/ga, jndi and etc., Java/defined products) | java - Java programming runtime |
| c) Java frameworks (Struts, Spring, Hibernate) | javac - Java compiling tool/runtime |
| JavaEE - Fully featured platform | javaw - Java runtime api |

[illegible]

Using a sampling interval of 100,000, we generated 100,000 observations.

- a) translation is rapid.
 b) translation is not a linear process.
 c) translation is polycistronic.
 d) initiation is the slowest step of translation.

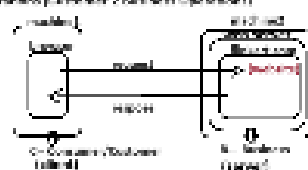
Scout du larve J2c1

o) The `App` that is specified in `mainComponent` and `mainView` only consists of a `Form` to generate the application in the `mainComponentApp`.

Web applications / websites

and likely to affect our view of the evolutionary large numbers that I listed.¹³ Furthermore, however, before we can solidify our understanding of evolution as a useful concept, we have to

- The proposed assignment model is supported by 228 experiments (Table 2). Experiments 2 and 3 are representative of the proposed assignment model. The proposed assignment model is supported by 228 experiments (Table 2). Experiments 2 and 3 are representative of the proposed assignment model.



• **Offload from App** is a possibility for some applications and will be supported by means of a User File. It will be a standard component for any application to be considered for this component.

As the web application developer, will be required by the client to ensure by creating the web application security (OWASP) for the internal operation of the web application and to make security of our data we need to install software that can monitor the operation of the web application and the web site on the network to monitor the web server side.

[illegible]

Downloaded from <http://ajph.org/> by guest on June 11, 2016

The Application: It allows different types of Client Applications locally or remotely to connect and communicate with the DBMS through the Database Client Interface. End User's Access.

- app: HP (Payment), Apple (Payment), Google (pay) and etc. }
 Payment of Customers (App) (Mon., no order, transaction and etc.)
 Payment of Banker (App) (no coin, payment, payment and etc.)
 (C.R. App) (no coin)

Need of Spring Boot

Spring Boot Spring ++

Spring Boot Framework = Spring Framework ++

note:: Framework is a software that is built on the top of one or more technologies to simplify the application development process by generating the common logics of the application..

note:: While developing the Apps using the frameworks the Programmers should take care of the application specific b.logics becoz the remaining the logics will be generated by the Framework dynamically

=>In Technology based app development,we must take care of both common logics and application specific logics

=> In Framework based app development, we we need to take care of only app specific logics becoz the common logics of the app will be generated by the framework dynamically

Spring framewor

Spring boot framework

Spring boot framework is the super set of Spring framework

Spring boot= spring - xml configurations

note:: The java class whose object is created and managed by the Spring Container /IOC container is called Spring Bean

like

AutoConfigurations

(avoid or minimize xml configurations)

(Based on the jar files added to the

=> Spring framework provides abstraction on java, jee technologies and avoids boilerplate code problem

=> Spring boot framework provides abstraction spring frameworks and avoids the spring related boilerplate code problem

class path of the Project, The Spring boot performs lots tommon operations dynamically)

Spring boot f/w IS self-intelligent

i.e it does many common operations

automatically to simplify the

process of the

=> making certain pre-defined class the spring beans

=> Injects certain spring beans with certain other spring beans

=> Gives InMemory DB s/w (This DB s/w will be created only during App's execution)

=> Establishing the connection with DB s/w

=>provides the additional plugins that are required in app's execution

=> Gives Servers as the Embedded Servers like Tomcat server and etc..

=> Takes care of adding more relevant and

Application development and execution

dependent jar files addition to the CLASSPATH based on main jar files that are added

and etc...

Spring f/w is a good worker /employee (Does every thing based on instructions given by the programmer)

Spring boot f/w ilike intelligent employee (Does most of common operations automatically)

(self intelligent)

In Java Domain, we can use 3 platforms to develop the projects

a) Java language (core java) (raw materials of the programming)

b) Java Technologies (adv.java) --> jdbc, servlet,jsp, jms,jta, jndi and etc.. (semi-finished products)

c) Java frameworks (spring-spring boot)

jsp :: java server pages jms :: Java messaging service

jndi :: java naming directory interface jta :: java transaction api

Enterprise Apps

(Best)

(Fully finished products)

=>if we develop Proj1 (Super market) using Java language (core java) we need to write 10,000 LOC (Lines of Code) =>if we develop Proj1 (Super market) using Java Technologies (adv.java) we need to write 5,000 LOC (Lines of Code) =>if we develop Proj1 (Super market) using Spring Framework we need to write 4,000 LOC (Lines of Code) =>if we develop Proj1 (Super market) using Spring Boot Framework we need to write 1,500 LOC (Lines of Code) (Becoz of the AutoConfiguration activity)

Using Spring boot framework, we can develop

a) standalone Apps

b) web applications/Web sites

c) Distributed Applications

is

d) MicroService Architecture Applications (** spring boot known for these applications)

Standalone Apps

=====

=>The App that is specific to one computer and and allows only one user at a time to operate the application is called standalone App

eg: Core Java App (class with main(-)), calculator app, calendar app, desktop game, anti-virus s/w and etc..

Web applications/websites

=====

=> It is a client-server or two-tier App where the Client software browser interacts with software app called website or web application in request -response model supporting C2B operations (Customer 2 Business Operations)

eg:: browser to flipkart.com

eg: browser to amazon.in

eg:: browser to gmail.com

eg:: browser to nareshit.com and etc..

Distributed App / Remoting App

=====

=====

machine1 browser

machine2

request

web server. flipkart.com (website)

respose

C-- Consumer/Customer (client)

B-- Business

(server)

one

=>Standalone App is specific to one computer and will be operated by user at a time So it will be executed manually i.e no automation is required in the execution

=> The web application/website will be operated by the lots of end users by sending simultaneous requests 24/7. For this the manual execution of the web application and its web comps is not possible i.e we need to special software that automates the execution of the web application and its web comps. that software is nothing but web server s/w

=> web Server is a piece of software that automates web application and its web comps execution. It listens to client requests continuously, takes the requests , maps the requests with web comps, executes the web comps and delivers the generated results to the browses (clients) as responses

The Application that allows different types of Client Apps either locally or remotely to connect and consume services in B2B (Business to Business) model

called Distributed App

server

server

server

eg:: UPI Payment Apps (Phone Pe, Google pay and etc..)

Payment Gateway Apps (Visa, master, mastreo and etc..) Payment Broker Apps (razor pay, nimopay, paypal and etc..) ICC App (scoring App)

BSE App NSE App

(Trading Apps)

ICC :: International Cricket Council (scoring) BSE:: Bombay Stock Exchange (Trading) NSE :: National Stock Exchange (Trading)

eg: Tomcat, weblogic, jetty, undertow, wildlfy, glassfish,...

we can consume the services of phone App (sever App) from different types of client Apps

like Mobile Apps, Desktop Apps, web applications, IOT Apps, Embedded System App, another Distributed app and etc..

Spotity Mobile App

Machine 100

Machine 10

Bank App

flipkart.com (web application)

network

Phone Pe server App Distributed app)

(Distributed app)

Desktop App (Bank employee Appt

Swiping machine

(Embedded system)

Alexa device (IOTY

OTT Application (Netflix.com)

server

Browser

flipkart.com

-Phone Pe App

BankApp

web application (c2b)

(e-commerce App) browser -----> amazon.in

----> paypal/razor pay

↓ Distributed App (b2b)

(Payment broker)

Distributed App (b2b app) (Payment Gateway)

-> VISA/Master/Mastero/...

-----> Bank App(SBI/ICICI)

web application (c2b)

(Distribucted App) (b2b) (Distribucted App) (b2b)

(Distribucted App) (b2b)

Payment broker Apps act as bridge b/w e-commerce apps and Payment gateway Apps

eg:: paypal, razorpay and etc..

Payment Gateway Apps provide world wide infrastructure to Operate the credit and debit cards eg: VISA, MASTER and etc..

Enterprise App = Only web application or only Distributed app or combination of both

(web application+ Distributed app)

What is the difference between web application and Distributed Application?

Web application

a) It is always C2B Application (Customer to Business)

b) Allows only browser as the the Client

c) This is browser to software App interaction

d) represents two-tier Apps as the Thin Client - Fat Server App

e) Runs on request -response model

f) we can use servlet,jsp technologies develop

to java based web applications

use

g) we can struts,jsp, spring mvc, spring boot mvc

frameworks to develop the java based

web applications

Distributed Application

a) It is always B2B application (Business to Business)

b) Allows different types of the Client Apps like Mobile App, web application, Desktop App, Embedded System App, IOT apps, Another Distributed App

c) This is application to application interaction

d) This is Fat Client - Fat Server App

e) Runs on Services invocation model (we call the b.methods) (Either through method calls or through request-response model)

f) we can use RMI, EJB technologies to develop the java based distributed Apps

g) we can use WebServices frameworks like jax-ws, jax-R■ apache cxf,apache axis spring rest,

and etc.. are the frameworks to develop the java based distributed Applications

spring boot rest

App App

h) eg: flipkart.com, amazon.in, nareshit.com

eg:: Phone pe, google pe, Paytm, ICC App, BSE App, VISA, Master and etc..

d) MicroService Architecture Applications

to

=>The standalone Apps, web applications, distributed Apps that we have discussed so far are developed based on Monolithic Architecture which say develop different services as the different modules of the Project and pack them single unit either as jar file or as war file

jar file: Java archive file

war file :: Web application archive file

Bank Project (ICICI Bank)

Daily Tx service/module

(Monolithic/ Monolith)

Loan

service/module

BankApp.war/jar

Mutual funds service/module

Locker service /Module

Admin Service/module

=> if other Projects or Client organizations want to use one or another service or module of the Monolithic Architecture Project .. that is not possible because all the services/modules are

The problem in one module/service may affect another module /service because all the modules/services are integrated into single unit

packed into single unit called war file or jar file. Moreover, all the services/ modules of the Project must be developed using same

of

language/technology/framework

To overcome this problem, take the support MicroService architecture in the development of the Project .. which

says develop every service /module as the separate project (war/jar file) and make them interacting with each other by taking the support of third party tools and design Patterns (Best practices)

=> In microservice (MS) arch, every service is going to be a project having its own packing of jar or war file

MS#1

Daily Tx Service

(proj#1) --> .war

DB s/w

MS#4

Locker Service

--> .war

(proj#) --

DB s/w

MS#3

MS#2

Loans Service

Mutual Funds Service

(proj#2) --> .war

(proj#3) --> .war

DB s/w

DB s/w

MS#5

MS#6

Admin Service

Reports Service

(proj#4) --> .war

(proj#5) --> .war

For this communication or interaction

DB s/w

b/w microservices we take the support of third party tools like API Gateway, Eureka server

DB s/w

jar file represents standalone app

war file represents web application

and lots of Design Patterns (best practices)

Basic Advantages of MicroService Architecture

=====

is

=====

Since every Service developed as the independent project in the form of jar or war file they can be used in different projects on pay and use Model

=> The Problem /bug in one Micro Service does not effect another Micro Services becoz all these are independent services/Projects

=> Every MicroService is the extension of WEBService and web services can be developed and consumed in different languages i.e the microservices of the MS architecture can be implemented either in same language

or in different languages

or

to

=> Most of the new Project in the realtime are happening moving MciroServices architecture

Micro Services = webservices ++

What is the difference b/w Webservices and microsevices?

Ans) WebServices is the methodology that allows us to develop interoperable Distributed apps i.e we can develop the server apps having b.methods in any language and their services can be consumed either in same language or different language Client apps

=> MicroServices in the special architecture to develop the Enterprise Apps in which every service will be developed as the separate Project in the form webservice project and they will be integrated using third party tools and design patterns

Micro services arch app is the extension of web services app

The AutoConfiguration feature of Spring boot helps to get common operations automatically like establishing the connection with DB s/wth the Projects/Services of MicroService ARchitecture Application. So we can say

Spring boot is the Best env. to develop MS Architecture Projects

We use following modules spring boot framework to develop different types of apps

=>Spring boot Core:: For standalone apps

=> spring boot web mvc :: for web applications

=> Spring boot Rest :: For Restful web services

=> Spring Boot cloud :: For MicroService architecture based apps

all these modules based apps use additional modules as the supporting modules

a) spring boot data jpa :: To interact with SQL DB s/ws

b) spring boot data mongodb:: To interact with MongoDB(NoSQL)

c) spring boot security :: For SEcuring the apps

d) Spring boot scheduling :: For enabling scheduling on the apps

and etc..

]