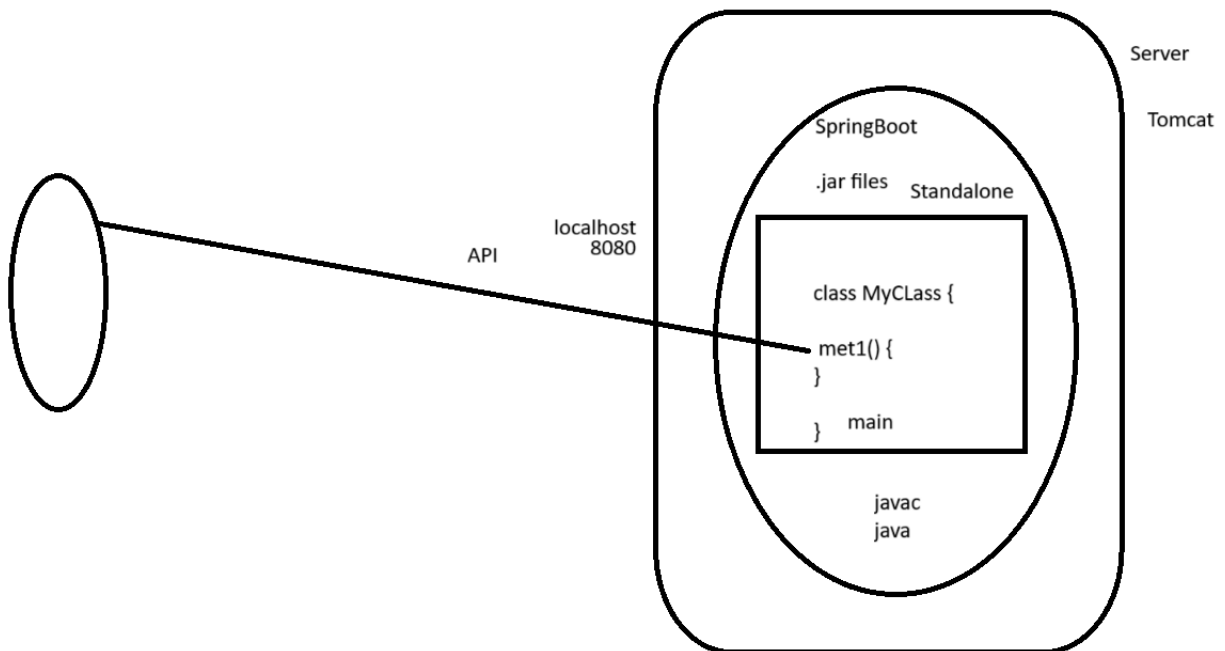
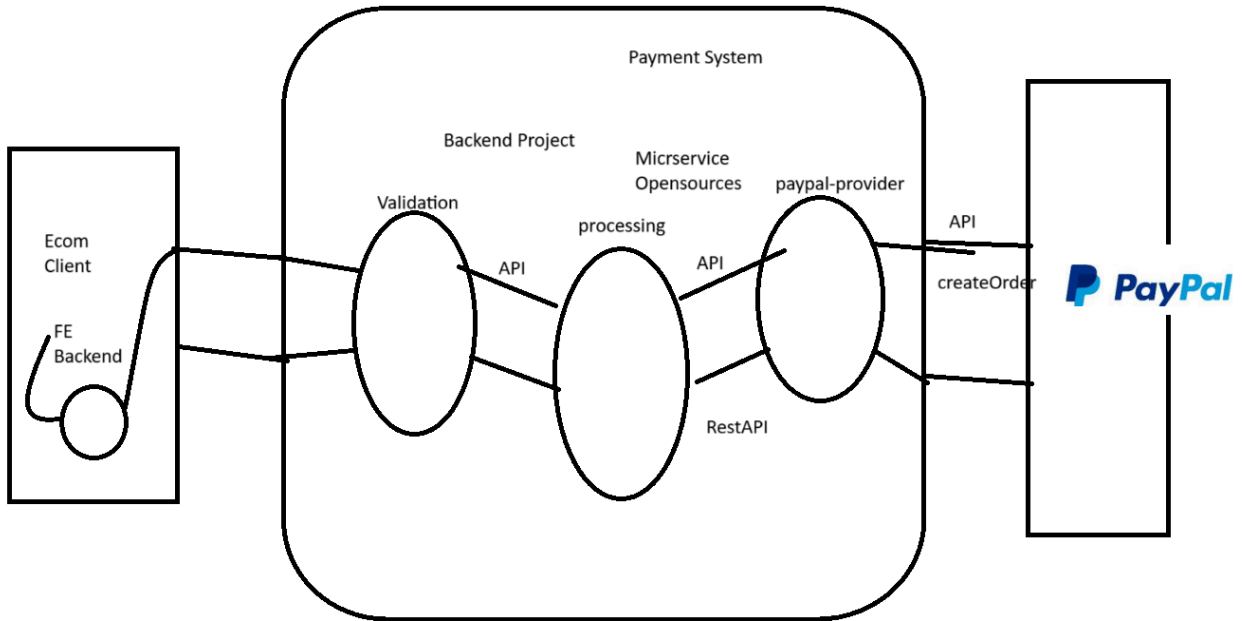


# Diagrams:



# LIVE NOTES

Welcome to Day1 of Sep Java SpringBoot & Microservice internship.

We will start at 10:00AM.

See you See Soon!!!

-----

10AM - 12PM

Overview about Internship

Technical topic

what software setup.

basic of IT tools for regular coding activities..

AWS deployment

10:02

Tausief - 14yrs

2010 - 2011

Java dev - 4yrs

Fullstack - FE/AWS/ScrumMaster - 6yrs

Tech Lead

Manager - 10-15

Senior Manager - 25-30

Head of Technology - 40member

10+ yrs interview candidates...

Entry level

Fresher - 3-6months, project - top companies

small - mid - not able to spend that money or time in training..

College to complete internship

- 3rd/4th

HR: Fresher:

Strong coding knowledge + Good project in resume.. + internship exposure..  
Then ok to hire you.

Fresher: First hire us.. so we can learn & do these..

Many jobseekers..

=====

Internship:

5-10 people

Bulk onboarding for doing internship.. many jobseeker would get exposure to IT ways of working..

1. Strong Realtime Project - working for our Client. IT standards/tools/tech
2. Project & client name in your resume
3. Reflect on linkedin that you are currently working on this internship
4. Internship Offer Letter & Completion
5. Knowledge 1-2yrs of IT exp.

70% of Jobseekers..

2months

working professionals

1-2 yr of IT exp,  
2 months

Complete practical internship

day-to-day live coding, designing, testing, feature understanding.. AGILE modal

You are supposed to code & build application..

Primary, you should be able to solve your own problems!

Project?????

Paypal Payment Service Provider (PSP)

End of session - more details about internship  
Week1-Week2  
2months

Ecom client => Payment System (We) => Paypal  
- Fully backend project.  
RestAPIs  
Java Spring Boot & Microservices

API - 2 parts of system connects to each other using API  
http/https - WebAPI  
How to develop APIs - standard  
Standard for developing APIs, RestAPI standards.

Security - OAuth

<https://www.paypalobjects.com/devdoc/IntegratingCheckoutInIndia.pdf>  
Read the API integration document  
[https://developer.paypal.com/docs/api/orders/v2/#orders\\_create](https://developer.paypal.com/docs/api/orders/v2/#orders_create)

between ecom <=> payment <=> paypal  
multiple API communication happens.. to make every single payment possible..  
double payment should not happen.  
money should not get stuck in between.

How to approach any new integration, like Paypal.

1. We read documentation to understand the over flow.
2. Relate the documentation with API contracts
3. Test these APIs (Postman API testing tool)  
& end-to-end flow
4. Strong understanding of how Paypal works
5. Designing (highlevel )
6. Coding  
(interface, methods, reusable, modular, packing) - low level

Coding - Java SpringBoot & Microservice  
Unittesting - JUnit + Mocking  
Code review

local machine..

dev env

qa env

uat env

prod env

7. Testing

8. UAT

9. PROD launch

how to access API: API Contracts

<https://api-m.sandbox.paypal.com/v2/checkout/orders> (endpoint / url of how to call this API)

Request

Response

Working on any project is a Journey..

Multiple step by step.

Standard for working on project is requirement..

SDLC - Agile approach of software development.

Scrum, Kanban, ScrumBan,...

Tools & Process & Technology (Core, adv - framework)

Design, code, test

----Core-----

Core Java - JDBC - Web

java class

method

for

Fresher

roadmap.sh

Core + jdbc + web

Java - How to work in project

1. Core
2. Build tool - Maven
3. How to talk to DB: ORM or JDBC
4. Logging framework: SLF4j
5. Unit Testing + Mocking
6. Basics of Internet
7. Version Control System - Git / Bitbucket
8. Relation DB: MySQL
9. RestAPI + Security

Daily Attendance..

Task tracking

Demos

Internship Completion Letter

Build 1st WebApp using Spring Boot.

1. You code in IDE - Eclipse STS  
<https://spring.io/tools>
2. Build tool - Maven
3. APIs

We

Your standalone java code, you need to convert to web application, so others can access it across internet

- write an API to call met1, so others can call it.

1. How to convert standalone app into web app

Use a web framework. SpringBoot  
will give lot of read-made code  
This code converts your java app into a web application.  
This readmade code would be available as .jar files.  
Download this & setup in your classpath & use it.  
Your application would become a web application

JDBC - ojdbc.jar, MySQL.jar, ..

Build tool - Maven

It will manage all these jar dependencies for you.

2. How to expose method as APIs so others can invoke it.

====

Building your 1st SpringBoot application (Webapp)

1. Use <https://start.spring.io/> to create your 1st SpringBoot application. => zip file which has your 1st application.

2. IDE - Eclipse

3. Organize all your local projects inside a workspace folder

D:\ctws\sep24ct

4. extract zip

5. Load the zip into IDE

Import Project => Existing maven project

6. All maven configurations are available in pom.xml

All configs which we did in <https://start.spring.io/> is available inside pom.xml

7. How to run webapplication

Server is required. Tomcat is default embedded server

8. Maven commands to run these application

clean

install - download your dependent jar files

package

spring-boot:run

rtcli => runas => maven build... => goal: clean

service, it runs on a port

Spring Boot app runs on tomcat server, which is on default port 8080

```
http://<where>:<port>  
http://localhost:8080  
    default  
    static/index.html  
Goal  
clean spring-boot:run
```

Group: which company developed the project.

mycomp.com

com.mycomp

Tomorrow 10Am, functional business RestAPI  
Postman, AWS, testing in AWS



