Java – full stack developer bootcamp

# Index

# Week 1: Core JAVA

# Week 2: advance Java, GIT-HuB and Mavan Depedency

# Week 3: Rest full API, Spring boot and API testing

# week 4:

# Spring JDBC, hibernet and JPA

# introduction to AWS (amazon Web services- Cloud)

# Week 5: Ui devlopment using AngularJS

# Week

## Getting Started with Java

## Introduction to IDE’s and Creating first java application

**Software Required before the beginning of class**

**Eclipse**

<https://www.eclipse.org/downloads/>?

**JDK 8**

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

How to install java in mac with Java home

How to install Java in windows

## Data Types and Variables

## Operators and Expressions

## Control Flow

## Methods

## Object Oriented Programming

## Objects and Classes

## Using Java Objects

## Inheritance in Java

# week

## Advanced Inheritance and Generics

## Exception handling

## Input/output Streams

## **GITHUB Account**

## <https://github.com/join>

## **Install GIT**

## <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

## Maven Dependency

* Manages Dependencies - Web Layer (Spring MVC), Data Layer (JPA - Hibernate) etc.
* Build a jar or a war or an ear
* Run the application locally - Tomcat or Jetty
* Deploy to a T environment
* Add new dependencies to a project
* Run Unit Tests

## Inductions to Web services

## Web server vs Application server

## Introduction to Frameworks in java

# week

## Spring Boot – REST full API (Application Programming Interface)

* **Introduction to Spring Framework**
  + - The most important concept in Spring. Here we look at what it is, and why you need a Container to support it.

**REST Warm Up**

* We start by reviewing the HTTP Verbs.

**Introducing REST**

* What is REST? Is it a standard? We define REST using "4 core principles". In this chapter we explore the first two of these.

**Representations and URIs**

* We'll start by building some REST representations and assigning them URIs.

**Introduction To Spring Boot**

* What is Spring Boot? What's in the course?

**Getting Started with Spring Boot**

* + - Initializing your first project (with Initializr) and mounting the project in Eclipse.

**REST Full API Testing**

* POSTMAN

<https://www.getpostman.com/>

* SoapUI :

<https://www.soapui.org/>

# Week

## **Spring JDBC and JPA (Hibernate)**

* Step 01 - Setting up a project with JDBC, JPA, H2 and Web Dependencies
* Step 02 - Launching up H2 Console
* Step 03 - Creating a Database Table in H2
* Step 04 - Populate data into Person Table
* Step 05 - Implement findAll persons Spring JDBC Query Method
* Step 06 - Execute the findAll method using CommandLineRunner
* Step 07 - A Quick Review - JDBC vs Spring JDBC
* Step 08 - Whats in the background? Understanding Spring Boot Auto configuration
* Step 09 - Implementing findById Spring JDBC Query Method
* Step 10 - Implementing deleteById Spring JDBC Update Method
* Step 11 - Implementing insert and update Spring JDBC Update Methods
* Step 12 - Creating a custom Spring JDBC RowMapper
* Step 13 - Quick introduction to JPA
* Step 14 - Defining Person Entity
* Step 15 - Implementing findById JPA Repository Method
* Step 16 - Implementing insert and update JPA Repository Methods
* Step 17 - Implementing deleteById JPA Repository Method
* Step 18 - Implementing findAll using JPQL Named Query
* Step 19 - Introduction to Spring Data JPA
* Step 20 - Connecting to Other Databases

# Week

1 Overview of AngularJS

1.1 AngularJS architecture overview

1.2 Using Karma

1.3 Get Setup

1.4 Angular Seed Tour

1.5 Build: Hello World

Bind tweet to input

Disable button, if tweet is empty

Character count binding

2 Data Binding

2.1 Wiring up a controller

2.2 Binding

2.3 Iteration

2.4 Filters

2.5 Forms binding and validation

2.6 Build: A two-way bound form with validation

Post: add tweet to list

Show error validation message (tweet too long)

Search filter by tag/username/text

3 Services and DI

3.1 Overview of the built-in AngularJS services

3.2 Using angular's $http and $resource services

3.3 Promises

3.4 Service registration and injection

3.5 Using services to build a service

3.6 Injecting services

3.7 Build: Create a twitter search service

Submit simple search

Set result size

Create a paging stream (with promises)

Pass tests

4 Templates and Routing

4.1 Linking and images

4.2 The routing API

4.3 PushState, hasbangs and SEO

4.4 Build: Using templates with iteration

Create a template for each tweet in the list

4.5 Build: Using templates and routing for master/details navigation

Create a details page for a tweet

5 Directives

5.1 Simple directives

5.2 Using templates

5.3 Working with controllers

5.4 Transclusion

5.5 Directive scope and isolate scope

5.6 Build: Tweet Directive

Add a timer that updates tweet time continuously

Add a marquee news ticker that can accept data from the incoming tweets

Bonus: Use D3 to create a chart of something

7 Advanced Topics

7.1 How to avoid polluting the global namespace

7.2 Using modules

7.3 Some AngularJS best practices

7.4 Using $watch, $digest, $apply

7.5 AngularJS events

7.5.1 Routing lifecycle events

7.6 Route Resolving