



Introduction to Spring

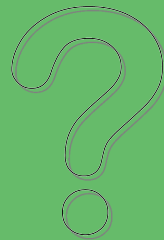


Let's meet



Cristian Iliescu

Senior Software Engineer
Java / Scala Trainer



Who you are?
What's your tech experience?
What's your class expectation?

Agenda



Agenda 1

- Introduction to Spring
 - Java and Web Concepts
 - Dependency Injection
- Spring Core
 - What is a Bean / Type of Beans
 - Spring Life Cycle
 - Stateful Application
 - Intro to MicroServices



Agenda 2

- Spring Batch
 - Introduction
 - Orchestration and Transactionality
- Spring Security
 - Basic Concepts
 - Auth Example
 - Filters



Introduction to WEB



Java Concepts - Check Knowledge

- Class vs Object
- Immutable objects
- Collections
- OOP concepts
- Java 8 features
- Multi-threading



Web Application

- A web application (or web app) is application software that runs on a web server, unlike computer-based software programs that are run locally on the operating system (OS) of the device.
- Web applications are accessed by the user through a web browser with an active network connection.
- These applications are programmed using a client–server modeled structure—the user ("client") is provided services through an off-site server that is hosted by a third-party.
- Examples of commonly-used web applications include: web-mail, online retail sales, online banking, and online auctions.



HYPertext TRAnSFER PRoTOCOL

An application protocol to transfer hypertext. A client sends a request to the server and server replies with a response. Often, the client is a browser.



URI - UNIFORM RESOURCE IDENTIFIER

URI is an unambiguous string that identifies a resource.

There are two types of URI:

- URN - Uniform Resource Name- identifies a resource in a namespace
- URL - Uniform Resource Locator



URL - UNIFORM RESOURCE LOCATOR

Identifies a resource by its location. It consists of:

- Protocol (http/https/ ftp/file/etc.)
- Domain name
- Optional port number (default port for http protocol is 80)
- Path
- Optional query and or a fragment preceded by an hash - additional parameters can be passed



URL EXAMPLE

For url: `http://google.com:80/search?q=abc`

- `https` - protocol
- `google.com` - domain name
- `:80` - port
- `/search` - path
- `q=abc` - query with a parameter named `q` with value `abc`.



IP ADDRESS

Web interfaces are identified in a network by their IP addresses. If an IP address locates an interface in a local network we call it private address and if it specifies global address we call it public address.



IP ADDRESS

- IPV4 - Fourth version of IP address is 32 bits long. It is commonly written as: a.b.c.d where a, b, c and d are values in range from 0 to 255.
 - Example IPv4 address: 216.58.215.110
- IPV6 - extend IPV4 pool sixth version of the IP protocol has been introduced (the first one did not get popular). IPv6 address is 128 bits long. It is commonly written as: a:b:c:d:e:f:g:h where a, b, c, d, e, f, g and h are values in range from 0000 to ffff (hexadecimal system).

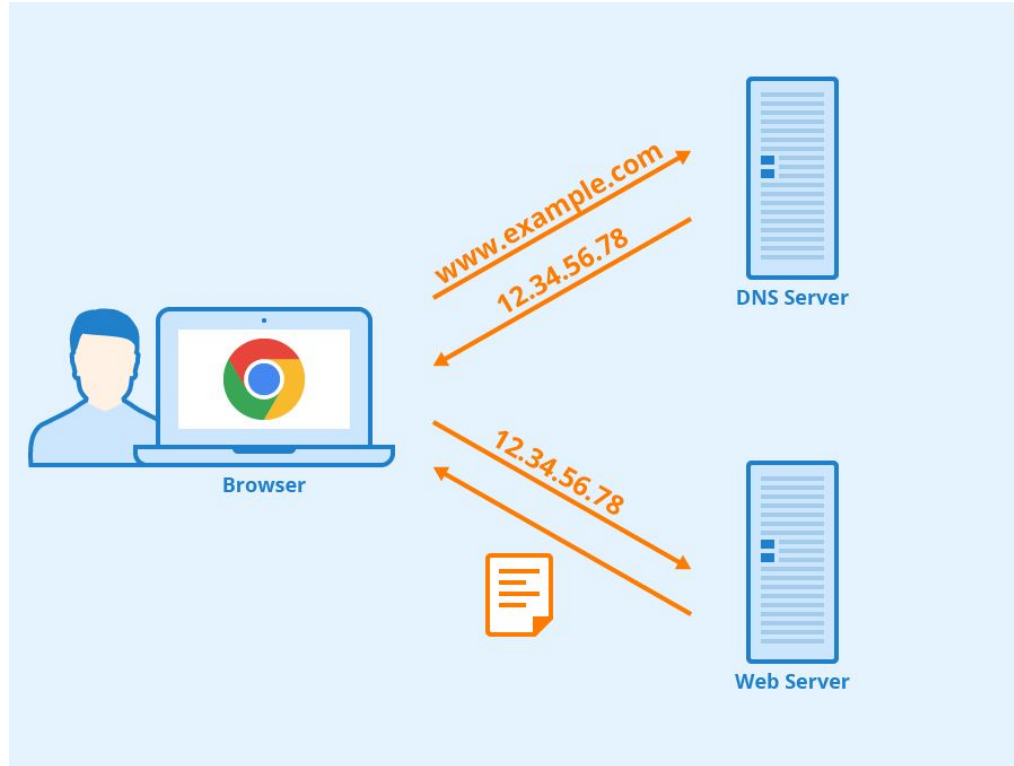


DNS - DOMAIN NAME SYSTEM

- DNS is a system that manages a databases of network addresses. It allows the usage user-friendly names instead of IP addresses. Such user-friendly name is called a domain name.



DNS - DOMAIN NAME SYSTEM



HTTP REQUEST

HTTP request is a set of data that client sends to server.

It consists of:

- Method
- URL
- Headers
- Body



HTTP REQUEST METHODS

- POST method is used to submit an entity to the specified resource. It often results in creating a new object and storing it.
- GET method requests retrieve data. When a web browser loads a web page it uses the GET method.
- PUT method is used to replace a resource with it's new representation, attached as the request body.



HTTP REQUEST METHODS

- DELETE method deletes specified resource.
- PATCH method applies a partial modification to a specified resource.
- OPTIONS method is used to get description of the communication options for the specified resource.



HTTP HEADERS

A http header can be used to pass additional information with the request. It often contains metadata - information about the request itself.

- Authorization - specifies the authorization method and credentials.
- Content-type - specifies the format of data in request body.
- Cookie, Set-Cookie - can pass a cookie value or set cookie at the client.
- Accept - specifies the type of data that should be sent back.
- Location - a URL to redirect to.
- User-Agent - informs about the client (application type/operation system etc).



HTTP RESPONSE

A HTTP server responses to a request with a response.

HTTP response consists of:

- Status line (includes status code)
- Headers
- Body



HTTP REDIRECTION STATUS CODES

A response with a status code of 3xx is a HTTP redirect. If a browser receives such response it redirects to the location specified as the value of the Location header. The user is often unaware of the redirection.



HTTP REDIRECTION STATUS CODES

- 200 (OK) - response includes requested data
- 201 (Created) - sent data/document is stored on the server.
- 400 (Bad Request) - request is built incorrectly
- 401 (Unauthorized) - requested resource requires authorization
- 403 (Forbidden) - authorization does not meet required credentials.
- 404 (Not Found) - requested resource does not exist for the URL.
- 500 (Internal Server Error) - An error occurred at the server-side.



HTTP RESPONSE STATUS CODES

Status code wraps up the response for a request.

Status code is a 3-digit integer and it is categorized by it's first digit.

- 1xx: Informational
- 2xx: Success
- 3xx: Redirection
- 4xx: Client error
- 5xx: Server error



PING

Ping is a command line tool to check if a resource can be reached. Basic usage is: ping [address].

Some of the additional parameters (for Windows OS):

- t will ping till it is forced to stop (with Ctrl+C)
- n [count] specifies how many requests will be sent
- 6 forces the usage of IPv6 (if pinging with a domain name)



Dependency in Java



Dependency In Java

- A Java class has a dependency on another class, if it uses an instance of this class.
- We call this a class dependency.
- The dependency class can be:
 - Created inside main class
 - Inserted using constructor or setter



Dependency Injection

- In software engineering, dependency injection is a technique in which an object receives other objects that it depends on, called dependencies.
- Typically, the receiving object is called a client and the passed-in ('injected') object is called a service.
- The code that passes the service to the client is called the injector. Instead of the client specifying which service it will use, the injector tells the client what service to use.
- The 'injection' refers to the passing of a dependency (a service) into the client that uses it.



Dependency Inversion

- Definitions:
 - High-level modules should not depend on low-level modules. Both should depend on the abstraction.
 - Abstractions should not depend on details. Details should depend on abstractions.
- Clients should not depend of implementation of services, they should depend of interface of that services



Welcome to Spring



Introduction

- Spring Boot is an open source Java-based framework used to create a micro Service.
- It is developed by Pivotal Team and is used to build stand-alone and production ready spring applications.



What is Spring Boot?

- Spring Boot provides a good platform for Java developers to develop a stand-alone and production-grade spring application that you can just run.
- You can get started with minimum configurations without the need for an entire Spring configuration setup.



Advantages

- Spring Boot offers the following advantages to its developers –
 - Easy to understand and develop spring applications
 - Increases productivity
 - Reduces the development time



Spring Boot Starters

- Handling dependency management is a difficult task for big projects. Spring Boot resolves this problem by providing a set of dependencies for developers convenience.
- <https://start.spring.io/>



Spring Boot Starters

- Handling dependency management is a difficult task for big projects. Spring Boot resolves this problem by providing a set of dependencies for developers convenience.
- <https://start.spring.io/>



Dependency Manager



Dependencies

- A Dependency is an external standalone program module (library) that can be as small as a single file or as large as a collection of files and folders organized into packages that performs a specific task.
- Example:
 - Spring
 - Spring-data
 - Guava



Dependency Manager

- Dependency managers are software modules that coordinate the integration of external libraries or packages into larger application stack.
- Dependency managers use configuration files like
 - `Composer.json`
 - `Package.json`
 - `build.gradle`
 - `pom.xml`



Dependency Manager

- What dependency to get
- What version of the dependency in particular
- Which repository to get them from.

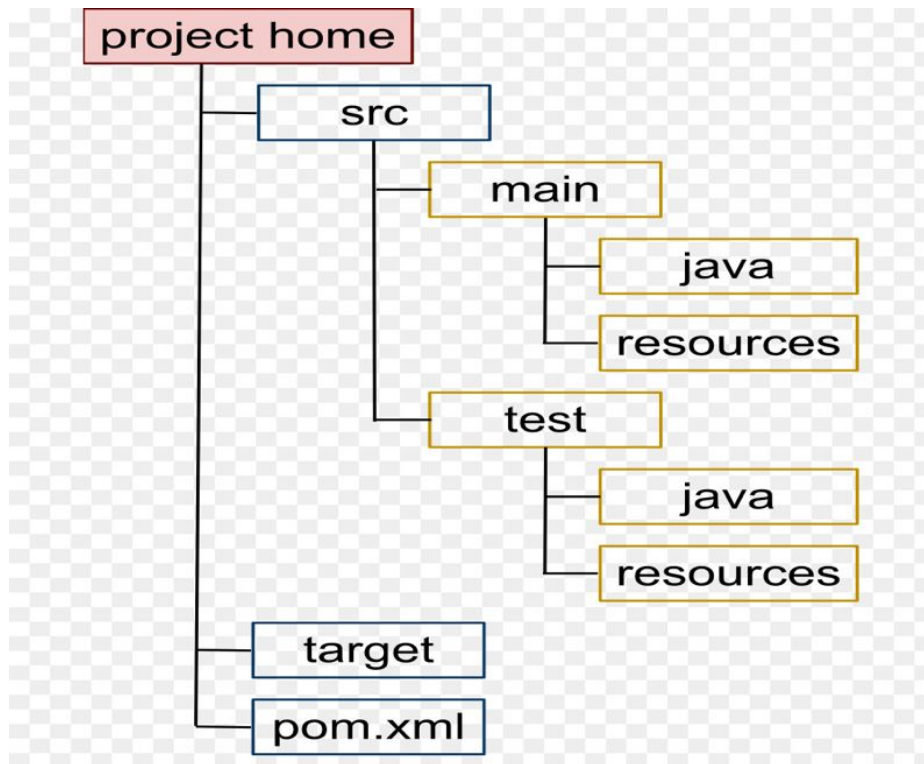


Maven

- Maven is Yiddish word that means accumulator of information.
- It's simply a project management tool and it's used mostly as
 - a build tool
 - dependency manager for any Java-based project.
- Maven is for Java just like npm is for JavaScript and composer is for php.
There are other technical differences, but they can be seen as similar tools for different languages in this context.



Maven directory



Nice to read

- <https://medium.com/@pascalskillz/beginner-introduction-to-maven-and-dependency-management-in-java-aa133d0ed348>
- <https://medium.com/prodsters/what-are-dependency-managers-26d7d907deb8#:~:text=Dependency%20managers%20are%20software%20modules,use%20configuration%20files%20like%20composer.>



Hello World in Spring



First Spring Project

```
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.EnableAutoConfiguration;

@EnableAutoConfiguration
public class DemoApplication {
    public static void main(String[] args) {
        SpringApplication.run(DemoApplication.class, args);
    }
}
```



Q&A





MOBILE / ACADEMY