//spring MVC

Spring MVC is a complete HTTP oriented MVC framework managed by the Spring Framework and

based in Servlets. It would be equivalent to JSF in the JavaEE stack. T

he most popular elements in it are classes annotated with @Controller,

where you implement methods you can access using different HTTP requests.

It has an equivalent @RestController to implement REST based APIs.

//spring boot

Spring boot is a utility for setting up applications quickly, offering an out of the

box configuration in order to build Spring powered applications. As you may know, Spring integrates

a wide range of different modules in its umbrella, as spring-core, spring-data,

spring-web (which includes Spring MVC, by the way) and so on. With this tool you can

tell Spring how many of them to use and youll get a fast setup for them

(you are allowed to change it by yourself later on).

So Spring MVC is a framework to be used in web applications and Spring boot

is a Spring based production-ready project initializer.

//AOP

In computing, aspect-oriented programming (AOP) is a programming paradigm that aims

to increase modularity by allowing the separation of cross-cutting concerns. It does

so by adding additional behavior to existing code (an advice) without modifying the code itself,

instead separately specifying which code is modified via a "pointcut" specification,

such as "log all function calls when the function's name begins with 'set'".

This allows behaviors that are not central to the business logic (such as logging) to

be added to a program without cluttering the code, core to the functionality.

AOP forms a basis for aspect-oriented software development.

//web template engine

A web template system is used in web publishing to allow web designers and developers

to work with web templates for the automatic generation of custom web pages,

such as the results from a search. This allows for reuse of the static elements of a web page,

while allowing the dynamic elements to be defined based on the parameters of the web request.

//CRUD

In computer programming, create, read, update, and delete (as an acronym CRUD)

are the four basic functions of persistent storage

// POJO = plain old java object

Java object which does not follow any of the major Java object models, conventions, or frameworks

//xmlElement

@XmlRootElement(name="customer")

@XmlAccessorType(XmlAccessType.FIELD)

public class Customer {

@XmlAttribute

protected int id;

@XmlElement

protected String fullname;

public Customer() {}

public int getId() { return this.id; }

public void setId(int id) { this.id = id; }

public String getFullName() { return this.fullname; }

}

The @javax.xml.bind.annotation.XmlRootElement annotation is put on Java

classes to denote that they are XML elements. The name() attribute of

@XmlRootElement specifies the string to use for the name of the XML element.

In our example, the annotation @XmlRootElement specifies that our Customer objects

should be marshalled into an XML element named <customer>.

The @javax.xml.bind.annotation.XmlAttribute annotation was placed on the id field of

our Customer class. This annotation tells JAXB to map the field to an id attribute

on the main <customer> element of the XML document.

In our example, the @javax.xml.bind.annotation.XmlElement annotation was placed on

the fullname field of our Customer class. This annotation tells JAXB to map the field

to a <fullname> element within the main <customer>

If we were to output an instance of our Customer class

that had an id of 42 and a name of “Bill Burke,” the outputted XML would look like this:

<customer id="42">

<fullname>Bill Burke</fullname>

</customer>

//RestTemplate

A more useful way to consume a REST web service is programmatically. To help you with that

task, Spring provides a convenient template class called RestTemplate. RestTemplate makes

interacting with most RESTful services a one-line incantation. And it can even bind that

data to custom domain types.

1- we create normal class may be with @JsonIgnoreProperties(ignoreUnknown = true)

2- then we can call in the main method

RestTemplate restTemplate = new RestTemplate();

Quote quote = restTemplate.getForObject

("http://gturnquist-quoters.cfapps.io/api/random", Quote.class);

3- or we can use @SpringBootApplication

@Bean

public RestTemplate restTemplate(RestTemplateBuilder builder) {

return builder.build();

}@Bean

public RestTemplate restTemplate(RestTemplateBuilder builder) {

return builder.build();

}

@Bean

public CommandLineRunner run(RestTemplate restTemplate) throws Exception {

return args -> {

Quote quote = restTemplate.getForObject(

"http://gturnquist-quoters.cfapps.io/api/random", Quote.class);

log.info(quote.toString());

};

}

Spring central class for synchronous client-side HTTP access. It simplifies communication

with HTTP servers, and enforces RESTful principles. It handles HTTP connections,

leaving application code to provide URLs (with possible template variables) and extract results.

//JSP

here we inject java code in the html code

//Servlet

here we inject html code in the java code

//handler mapping

like @RequestMapping("")

tells sprring which controller to call

//controller

every controller will have a model which is like map

that has key and value

modle.addAttribute("greeting","Welcome User")

controller returns the logical view name --> the view which will be displayed

// view resolver

it will return accept view name

it will display the view with the name returned from the controller

//Hibernate

framework to help integrate java application with database

we should add Hibernate and MySQL dependencies to the projects

so it helps to use @Entity and @Id

//xmlns

<?xml version="1.0" encoding="UTF-8"?> // first line to define file

In XML, it is illegal to omit the closing tag. All elements must have a closing tag

XML Attribute Values Must be Quoted

Element names cannot start with the letters xml (or XML, or Xml, etc) and cannot contain speaces

XML Namespaces provide a method to avoid element name conflicts

>>> In XML, element names are defined by the developer.

This often results in a conflict when trying to mix XML documents from different XML applications.

A user or an XML application will not know how to handle these differences.

Name conflicts in XML can easily be avoided using a name prefix.

<h:table>

<h:tr>

<h:td>Apples</h:td>

The namespace can be defined by an xmlns attribute in the start tag of an element.

The namespace declaration has the following syntax. xmlns:prefix="URI".

>>> When a namespace is defined for an element,

all child elements with the same prefix are associated with the same namespace.

<root xmlns:h="http://www.w3.org/TR/html4/"

xmlns:f="https://www.w3schools.com/furniture">

However, companies often use the namespace as

a pointer to a web page containing namespace information.

xmlDoc = parser.parseFromString(text,"text/xml");

An XML document with correct syntax is called "Well Formed".

An XML document validated against a DTD is both "Well Formed" and "Valid".

>>> DTD <!DOCTYPE note SYSTEM "Note.dtd">

DTD file is a validation file

<!DOCTYPE note

[

<!ELEMENT note (to,from,heading,body)>

<!ELEMENT to (#PCDATA)>

<!ELEMENT from (#PCDATA)>

<!ELEMENT heading (#PCDATA)>

<!ELEMENT body (#PCDATA)>

]>

#PCDATA means parse-able text data.

With a DTD, independent groups of people can agree to use a standard DTD for interchanging data.

>>> schema

XML Schema is an XML-based alternative to DTD:

can identify type and info about each tag

<xs:complexType>

<xs:sequence>

<xs:element name="to" type="xs:string"/>

<xs:element name="from" type="xs:string"/>

<xs:element name="heading" type="xs:string"/>

<xs:element name="body" type="xs:string"/>

</xs:sequence>

</xs:complexType>

Once you have the XML Schema Instance namespace available:

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

you can use the schemaLocation attribute. This attribute has two values,

separated by a space. The first value is the namespace to use. The second value is the location of the XML schema to use for that namespace:

xsi:schemaLocation="https://www.w3schools.com note.xsd"

//xsl

XSLT is a language that can be used to transform XML documents into other formats.

<xsl:for-each select="catalog/cd">

<tr>

<td><xsl:value-of select="title"/></td>

The easiest way to get the content of an element is by using the innerHTML property.