Spring Boot

LearnCodeWith Durgesh

What is Spring Boot

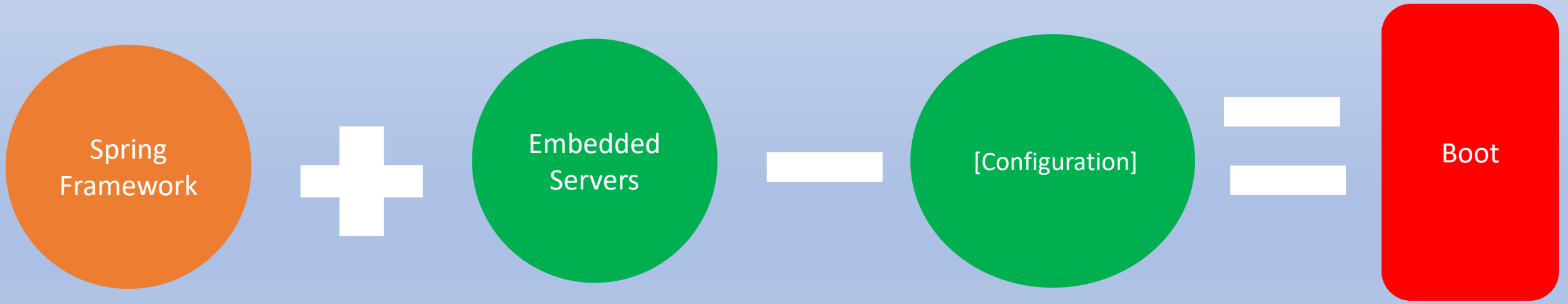
Spring Boot is module of Spring from which we speed up the development

Spring Boot makes it easy to create stand-alone, production-grade Spring based Applications that you can "just run"

LearnCodeWith Durgesh

What is Spring Boot?

It provides an easier and faster way to set up, configure, and run both simple and web-based applications.



LearnCodeWith Durgesh

What is Spring Boot?

Convention over configuration software design style.

It decreases the effort of the developer.

Opinionated Default- Automatically configure

LearnCodeWith Durgesh

What is Spring Boot?

Scan the class path and find the dependency it will automatically configure the things.

LearnCodeWith Durgesh

Advantages of Spring Boot

- It creates **stand-alone** Spring applications that can be started using Java **-jar**.
- Embed Tomcat, Jetty or Undertow directly (no need to deploy WAR files)
- Provide opinionated 'starter' dependencies to simplify your build configuration
- Automatically configure Spring and 3rd party libraries whenever possible
- Provide production-ready features such as metrics, health checks, and externalized configuration
- Absolutely no code generation and no requirement for XML configuration

LearnCodeWith Durgesh

How spring boot magic works?

No magic

Programmatic Configurations

LearnCodeWith Durgesh

Starter POM

spring-boot-starter-web

spring-boot-starter-data-jpa

Other.....

When we add starter jars
then it pull all the jars

Jar contain
META-INF/spring.factories

What should be enabled or
disabled based on some condition

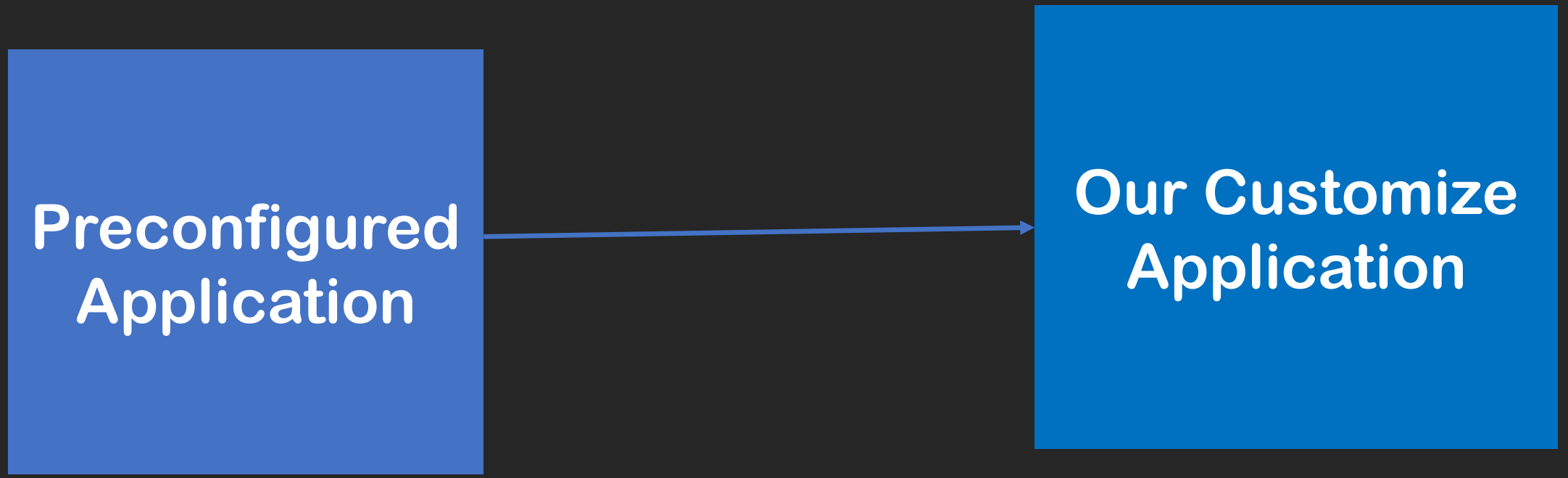
LearnCodeWith Durgesh

How to start with spring boot

1. Create a maven project and add starter dependencies
2. Use spring initializr
3. Use IDE like STS
4. Spring boot command line interface

application.properties

Application Configuration

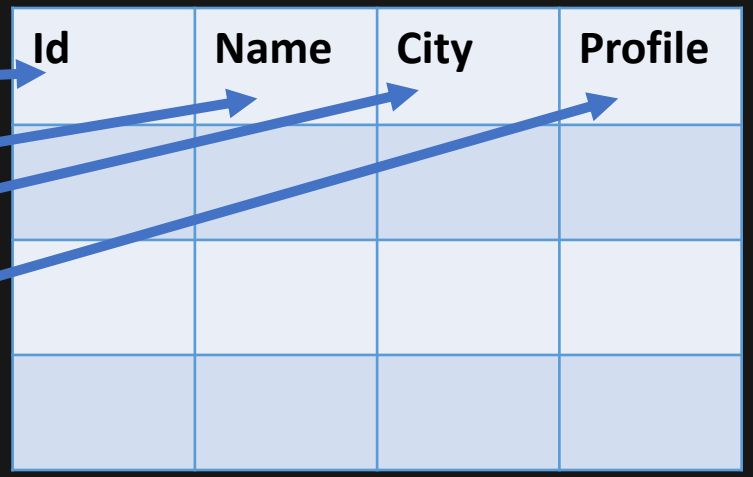


application.yml

JPA *(Java Persistent API)*

ORM

```
Class User
{
        int id
        String name
        String city
        String profile
}
```



JPA *(Java Persistent API)*

Implementation

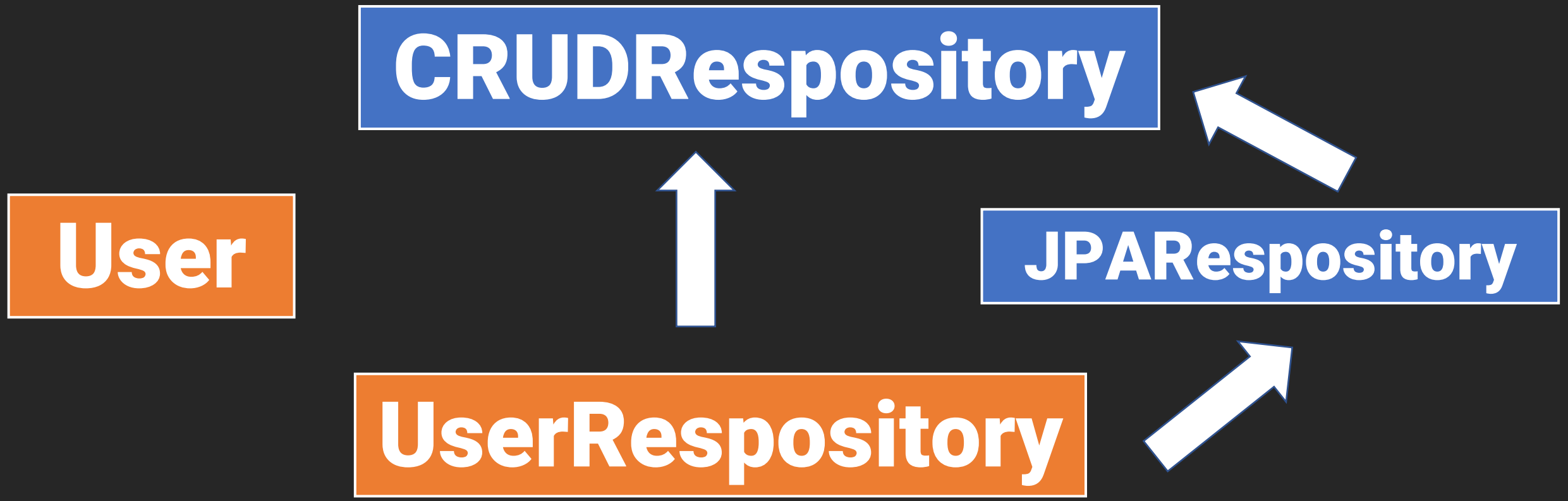
- ***Hibernate***
- ***Eclipse link***
- ***Open JPA***

EntityManagerFactory

EntityManager

- ***Create***
- ***Update***
- ***Read , Delete***

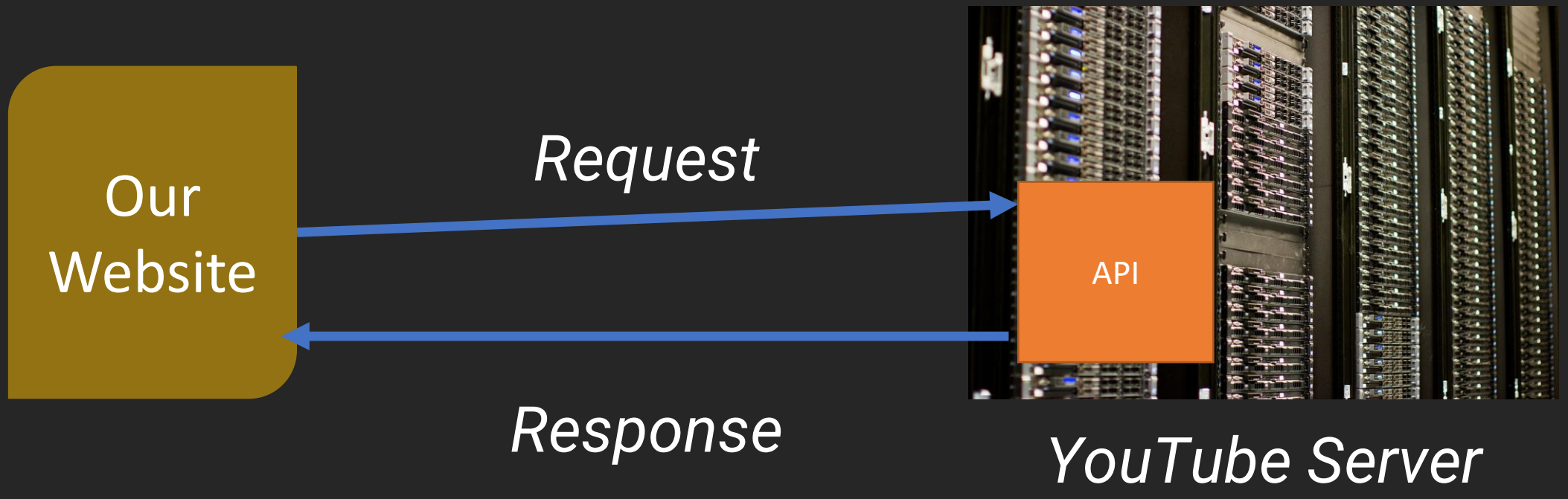
Spring Boot Makes easier to perform operation with JPA



Custom Finder Methods

API(*Application Programming Interface*)

It is a set of rules that allow programs to talk to each other. The developer creates the API on the server and allows the client to talk to it.



REST(*Representational State Transfer*)

A set of constraints to be used for creating Web services.

- Client-Server
- Stateless
- Cacheable
- Layered

For example,

POST /users: It creates a user.

GET /users/{id}: It retrieves the detail of a user.

GET /users: It retrieves the detail of all users.

DELETE /users: It deletes all users.

DELETE /users/{id}: It deletes a user.

GET /users/{id}/posts/post_id: It retrieve the detail of a specific post.

POST / users/{id}/ posts: It creates a post of the user.

HTTP also defines the following standard status code:

- **404:** RESOURCE NOT FOUND
- **200:** SUCCESS
- **201:** CREATED
- **401:** UNAUTHORIZED
- **500:** SERVER ERROR

Spring Boot DevTools

DevTools stands for **Developer Tool.** The aim of the module is to try and improve the development time while working with the Spring Boot application. Spring Boot DevTools pick up the changes and restart the application.

```
<dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-devtools</artifactId>
        <scope>runtime</scope>
</dependency>
```

Features of DevTools

1. Property Defaults
2. Automatic Restart
3. Live Reload
4. Remote Applications

IDE using For building REST API

VS Code

Setup is very important

Watch our VS code Spring

Boot Setup video

How to call API

Postman

POSTMAN

The important methods of HTTP are:

- **GET:** It reads a resource.
- **PUT:** It updates an existing resource.
- **POST:** It creates a new resource.
- **DELETE:** It deletes the resource.

Rest API

HTTP method	**URI**	**Description**	Valid HTTP status codes
POST	/books	**C**reate a book	201
GET	/books/{bookId}	**R**ead a book	200
PUT	/books/{bookId}	**U**pdate a book	200
DELETE	/books/{bookId}	**D**elete a book	204
GET	/books	Retrieve all books	200, 204, 206

Spring boot *File Upload*

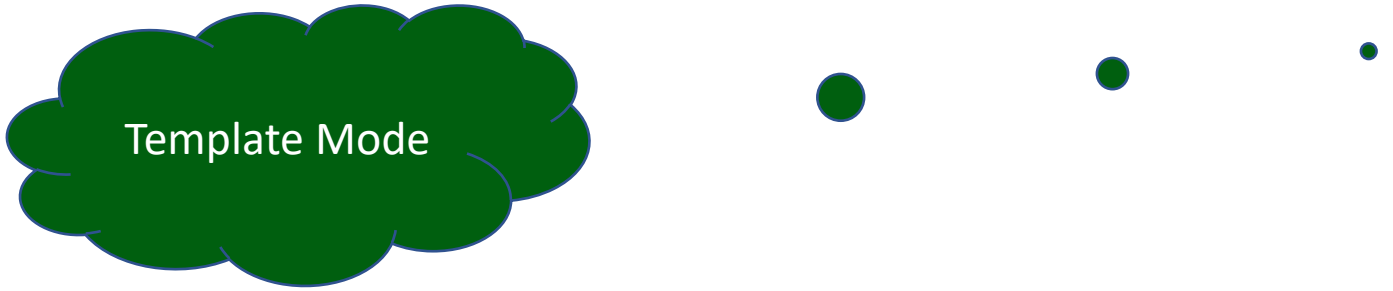
Introduction to Thymeleaf

Thymeleaf Template Engine

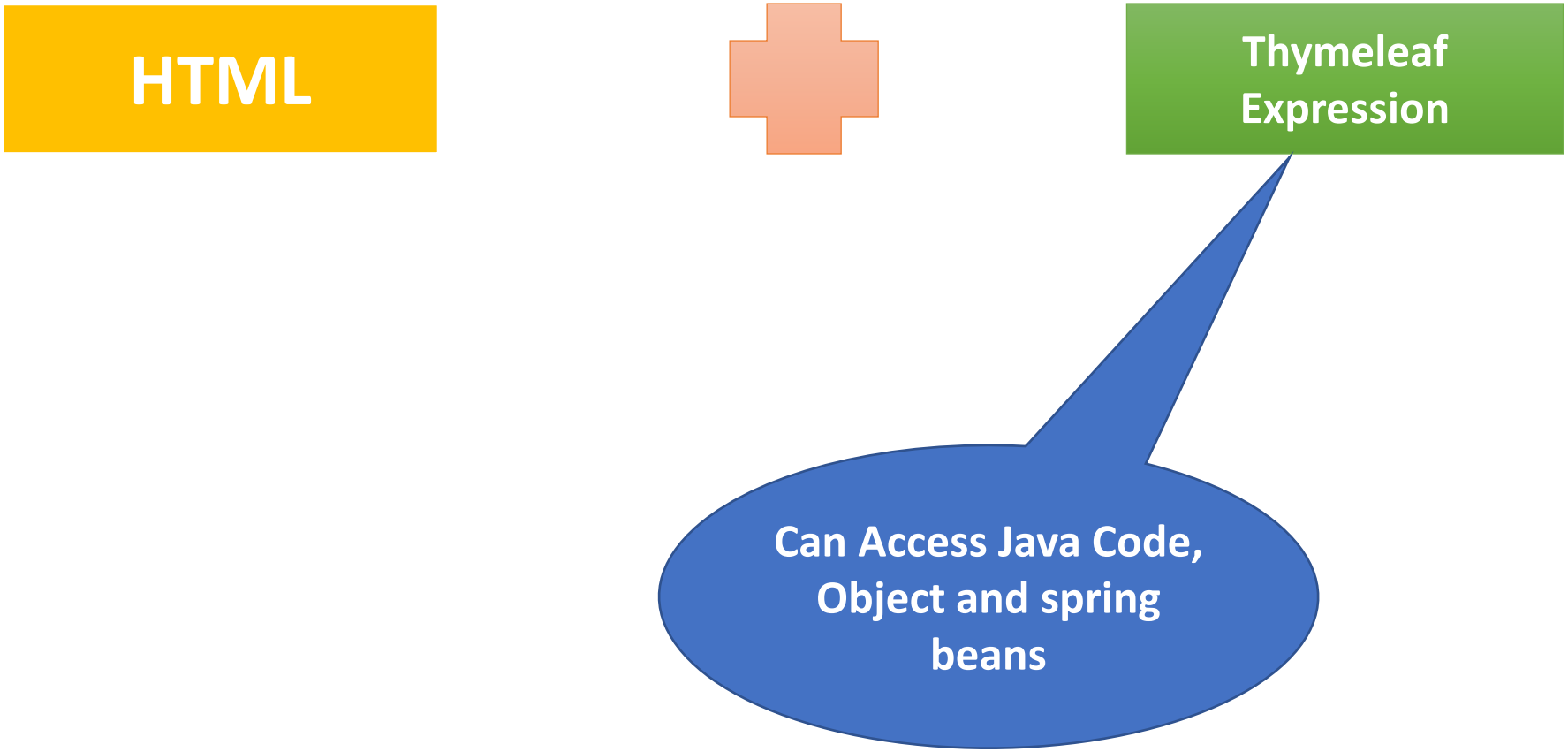
Thymeleaf is a modern server-side Java template engine for both web and standalone environments, capable of processing HTML, XML, JavaScript, CSS and even plain text.

The main goal of Thymeleaf is to provide an elegant and highly-maintainable way of creating templates

Mostly used to generate html views for web applications.



Thymeleaf Template

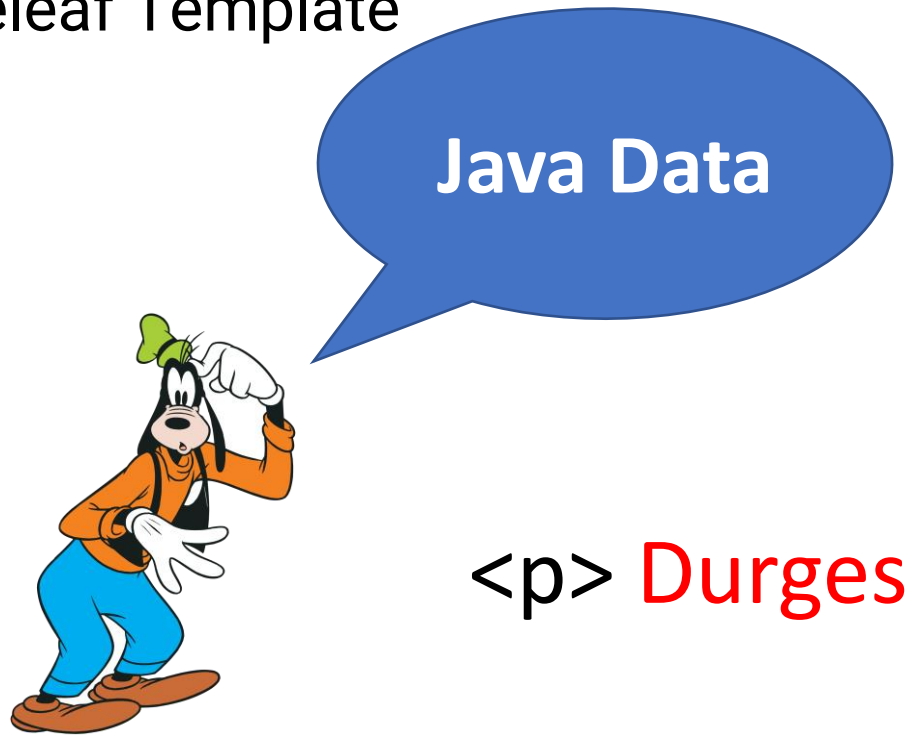


Example

```
19
20          <div class="container text-center    ">
21
22              <img class="mx-auto img-fluid rounded-circle profile-img" src=""
23                  th:src="@{'/img/'+${c.image}}" alt="" />
24
25          </div>
26
27          <h3 class="text-center mt-3"><span th:text="${c.name}"></span> ( <span t
28
29          <!--table for data -->
30
31
32          <table class="table mt-4 text-center">
33              <thead>
34                  <tr>
35
36                      <th scope="col"><i class="fa fa-envelope" aria-hidden="true'
37                      <th scope="col "><span th:text="${c.email}"></span></th>
38                  </tr>
39              </thead>
40              <tbody>
41                  <tr>
42                      <th scope="row"><i class="fa fa-phone" aria-hidden="true"></
43                      <td><span th:text="${c.phone}"></span></td>
44
45                  </tr>
46                  <tr>
47                      <th scope="row"><i class="fa fa-briefcase" aria-hidden="true
```

Thymeleaf Engine

Thymeleaf Engine will parse Thymeleaf Template



<p th:text="“${name}”>

 Durgesh

Setup
First Thymeleaf Program

Adding CSS and JS and Image

Adding Bootstrap

Conditionals
If, unless , switch

Looping {each}

Fragments
{Include, insert , replace}

Including Fragments

th:replace	th:insert	th:include

th:replace - It will actually substitute the host tag [tag where we use replace] by the fragment's. That means, It will remove the host tag and in place of host tag, it will add the specified fragment including the fragment tag.

th:insert - It will simply insert the specified fragment as the body of its host tag including the fragment tag.

th:include - It will also insert the specified fragment as the body of its host tag but excluding the fragment tag.

Inheriting Templates