

## List of Figures

1	Java Enterprise Edition logo . . . . .	2
2	AngularJS logo . . . . .	3
3	Bootstrap logo . . . . .	4
4	Bootstrap second starred on github . . . . .	6
5	Bootstrap on google trends . . . . .	7
6	Spring logo . . . . .	7
7	The request processing workflow in Spring Web MVC . . . . .	8
8	Apache Maven logo . . . . .	9
9	HTML5 logo . . . . .	9
10	CSS3 logo . . . . .	9

# Chapter 4 : Implementation

September 22, 2017

## 1 Introduction

In this chapter we will focus on the technologies.

## 2 Java Platform, Enterprise Edition (Java EE)



Figure 1: Java Enterprise Edition logo

## 3 Angular JS



Figure 2: AngularJS logo

### 3.1 Introduction

AngularJS is a structural framework for dynamic web apps. It lets you use HTML as your template language and lets you extend HTML's syntax to express your application's components clearly and succinctly. AngularJS's data binding and dependency injection eliminate much of the code you would otherwise have to write. And it all happens within the browser, making it an ideal partner with any server technology.

AngularJS is what HTML would have been, had it been designed for applications. HTML is a great declarative language for static documents. It does not contain much in the way of creating applications, and as a result building web applications is an exercise *in what do I have to do to trick the browser into doing what I want?*

The impedance mismatch between dynamic applications and static documents is often solved with:

- **a library** - a collection of functions which are useful when writing web apps. Your code is in charge and it calls into the library when it sees fit. E.g., `jQuery`.
- **frameworks** - a particular implementation of a web application, where your code fills in the details. The framework is in charge and it calls

into your code when it needs something app specific. E.g., `durandal`, `ember`, etc.

AngularJS takes another approach. It attempts to minimize the impedance mismatch between document centric HTML and what an application needs by creating new HTML constructs. AngularJS teaches the browser new syntax through a construct we call directives. Examples include:

- Data binding, as in `{{}}`
- DOM control structures for repeating, showing and hiding DOM fragments.
- Support for forms and form validation.
- Attaching new behavior to DOM elements, such as DOM event handling.
- Grouping of HTML into reusable components.

### 3.2 Features

this section include the features

### 3.3 Motivation

this section include the features

## 4 Bootstrap



Figure 3: Bootstrap logo

## 4.1 Introduction

**Bootstrap** is a free and open-source front-end web framework for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript plugins. Unlike many web frameworks, it concerns itself with front-end development only.

Bootstrap was developed by Mark Otto and Jacob Thornton at Twitter, and released as an open source product in August 2011 on GitHub.

**In June 2014 Bootstrap was the No.1 project on GitHub!**

## 4.2 Features

**Bootstrap 3** supports the latest versions of the **Google Chrome**, **Firefox**, **Internet Explorer**, **Opera**, and **Safari** (except on Windows). It additionally supports back to IE8 and the latest Firefox Extended Support Release (ESR).

Since **2.0**, **Bootstrap** supports **responsive web design**. This means the layout of web pages adjusts dynamically, taking into account the characteristics of the device used (desktop, tablet, mobile phone).

Starting with **version 3.0**, Bootstrap adopted a mobile-first design philosophy, emphasizing responsive design by default.

The **version 4.0** alpha release added **Sass** and **flexbox** support

## 4.3 Motivation

All the researchs shows that Bootstrap The most popular HTML, CSS, and JavaScript framework for developing responsive, mobile first projects on the web.

According to github, Bootstrap is the second starred repository, with **115k stars**.

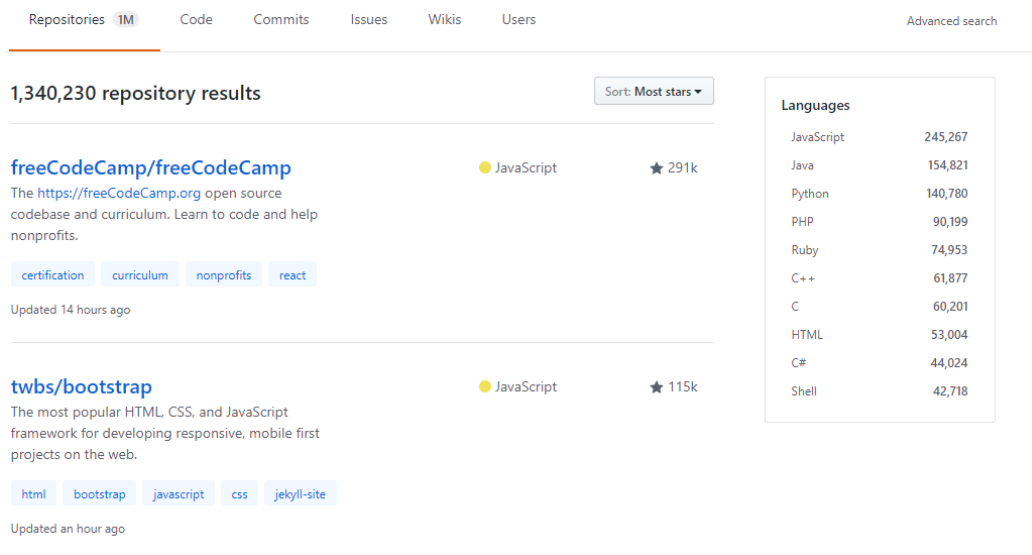


Figure 4: Bootstrap second starred on github

According to a researchs, did it by myself in the net, all the high-tech blogs encourage developers to use bootstrap.

I used a tool offred by google named **google trends** to make a comprai-son between a subjects in term of most researched.

After this comprai-son in google trends, Bootstrap also is the most googled css framework on the web , compared to the others css frameworks Eg. **Ma-terial UI** which was considered as the most popular css framework after Bootstrap.

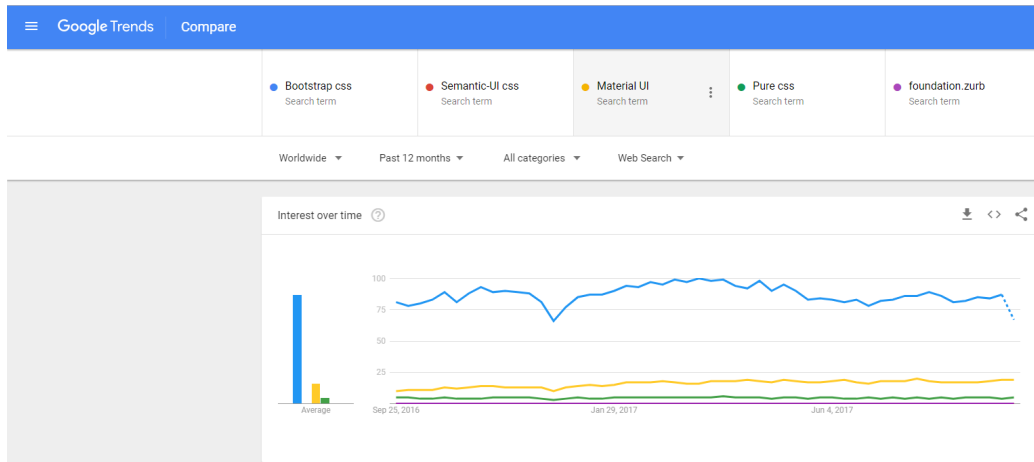


Figure 5: Bootstrap on google trends

## 5 Spring mvc framework



Figure 6: Spring logo

### 5.1 Introduction

The Spring Web model-view-controller (MVC) framework is designed around a `DispatcherServlet` that dispatches requests to handlers, with configurable handler mappings, view resolution, locale, time zone and theme resolution as well as support for uploading files. The default handler is based on the `@Controller` and `@RequestMapping` annotations, offering a wide range of flexible handling methods. With the introduction of Spring 3.0, the `@Controller` mechanism also allows you to create RESTful Web sites and applications, through the `@PathVariable` annotation and other features.

Spring's web MVC framework is, like many other web MVC frameworks, request-driven, designed around a central Servlet that dispatches requests to

controllers and offers other functionality that facilitates the development of web applications. Spring's `DispatcherServlet` however, does more than just that. It is completely integrated with the Spring IoC container and as such allows you to use every other feature that Spring has.

The request processing workflow of the Spring Web MVC `DispatcherServlet` is illustrated in the following diagram. The pattern-savvy reader will recognize that the `DispatcherServlet` is an expression of the “Front Controller” design pattern (this is a pattern that Spring Web MVC shares with many other leading web frameworks).

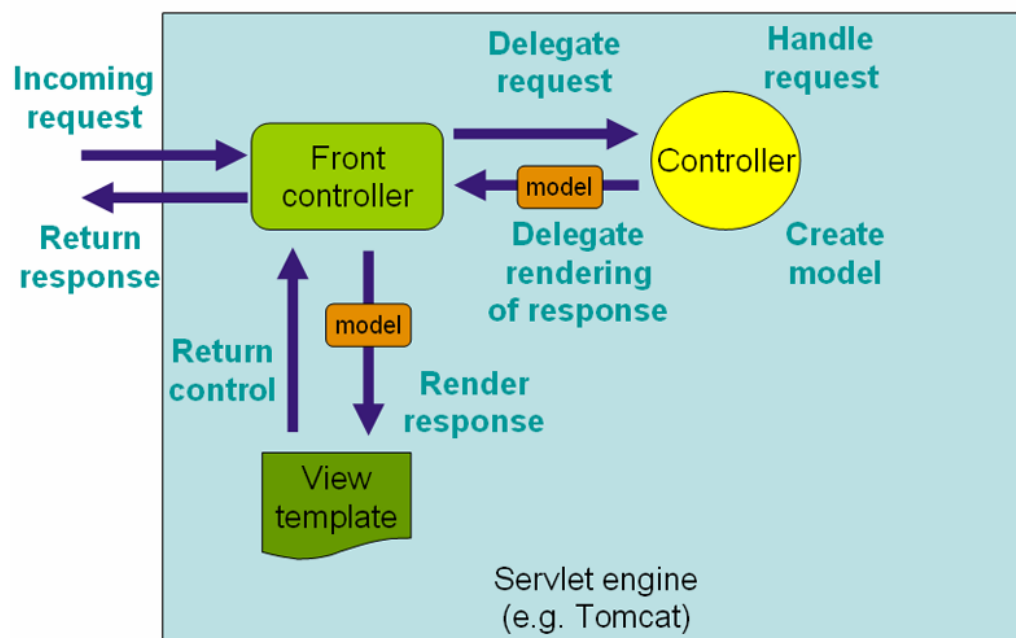


Figure 7: The request processing workflow in Spring Web MVC





Figure 8: Apache Maven logo



Figure 9: HTML5 logo

6 Apache Maven

7 HTML5

8 CSS3



Figure 10: CSS3 logo