الاسم : زياد فيصل
قسم هندسة البرمجيات

Spring Boot

من **Spring Boot**في عالم تطوير التطبيقات الحديثة، أصبحت إطارا العمل مثل الأدوات الأساسية التي يعتمد عليها المطورون لبناء تطبيقات ويب وتطبيقات هي جزء من إطار عمل Spring Boot .بكفاءة وسرعة (Microservices) ميكروسيرفيس (Auto-) الشهير، ولكنها تقدم مستوى أعلى من التبسيط والتكوين التلقائي Spring ،. مما يجعلها خيارًا مثاليًا للمبتدئين والمحترفين على حد سواء(configuration

، ميزاتها، بنيتها الأساسية، وكيفية Spring Boot في هذا البحث، سنستعرض **مفهوم استخدامها** لبناء تطبيقات قوية وسهلة الصيانة.

1.ما هيSpring Boot ؟

Spring Bootهي إطار عمل مفتوح المصدر يعتمد على لغة **Java**ويهدف إلى تبسيط عملية إنشاء تطبيقات Spring القائمة على الخوادم .(Server-side Applications) تم تطويرها من قبل **Pivotal Software**، وهي تعتمد على إطار عمل Spring الأساسي ولكن مع توفير إعدادات مسبقة (Default Configurations) تتيح إنشاء تطبيقات سريعة دون الحاجة إلى تكوين يدوي معقد.

أهميةSpring Boot

- توفر **بدءًا سريعًا** للمشاريع بفضل الأدوات مثل .**Spring Initializr**
- تدعم **التكوين التلقائي** (Auto-configuration)مما يقلل من الجهد المبذول في إعداد التطبيق.
- تحتوي على خادم مدمج) مثل Tomcat أو (Jetty مما يجعل نشر التطبيق أسهل.
- تدعم **RESTful APIs**و**قواعد البيانات** بسهولة.

2.ميزاتSpring Boot

تتميز Spring Boot بالعديد من الميزات التي تجعلها من أفضل الخيارات لتطوير التطبيقات، ومن أبرزها:

أ. التكوين التلقائي(Auto-configuration)

تقوم Spring Boot تلقائيًا بإعداد التطبيق بناءً على المكتبات المضافة (Dependencies)، مما يقلل الحاجة إلى كتابة أكواد تكوين طويلة.

ب. خادم مدمج(Embedded Server)

توفر Spring Boot خوادم مثل **Tomcat** أو **Jetty** مدمجة داخل التطبيق، مما يلغي الحاجة إلى نشر التطبيق على خادم خارجي أثناء التطوير.

ج. أدوات CLI (Command Line Interface)

يمكن للمطورين استخدام أدوات سطر الأوامر لإنشاء مشاريع Spring Boot بسرعة.

د. دعم قواعد البيانات(Spring Data JPA)

توفر Spring Boot تكاملًا سلسًا مع قواعد البيانات مثل **MySQL, PostgreSQL, MongoDB** عبر **Spring Data JPA**.

هـ. نظام إدارة التبعيات(Dependency Management)

باستخدام **Maven** أو **Gradle**، يمكن إضافة المكتبات بسهولة دون القلق حول توافق الإصدارات.

و. مراقبة التطبيق(Spring Boot Actuator)

تتيح Spring Boot مراقبة التطبيق وإدارته عبر نقاط نهاية (Endpoints) مثل `/health` و`/metrics`.

---

3. بنية مشروع Spring Boot

يتكون مشروع Spring Boot نموذجيًا من الهيكل التالي:

أ. ملف التشغيل الرئيسي(Main Class)

يحتوي على الدالة `main()` التي تشغل التطبيق باستخدام `SpringApplication.run()`.

```
@SpringBootApplication
```

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

ب. المتحكمات(Controllers)

تستخدم للتعامل مع طلبات HTTP وتوجيهها إلى الخدمات المناسبة.

```
@RestController
@RequestMapping("/api")
public class UserController {
@GetMapping("/hello")
public String sayHello() {
return "Hello, Spring Boot!";
}
}
```

ج. النماذج والخدمات(Models & Services)

- **النماذج (Entities)**: تمثل جداول قاعدة البيانات.
- **الخدمات (Services)**: تحتوي على المنطق التجاري للتطبيق.

4.كيفية إنشاء مشروع Spring Boot

يمكن إنشاء مشروع Spring Boot عبر:

أ. استخدام Spring Initializr (https://start.spring.io)

١. اختر خيارات المشروع (Maven/Gradle) ، لغة Java ، إصدار Spring Boot.
٢. أضف التبعيات (Dependencies) مثل **Spring Web, Spring Data JPA**.
٣. انقر على **Generate** لتحميل المشروع.

ب. استخدام محرك التطوير) مثل IntelliJ IDEA أو Eclipse)

١. افتح IDE واختر **New Project → Spring Initializr**.
٢. اتبع الخطوات السابقة لإنشاء المشرو

Full Stack

Full Stack Developerهو مبرمج قادر على العمل على جميع طبقات تطبيق الويب أو البرنامج، بما في ذلك:

- **الواجهة الأمامية :(Frontend)** الجزء الذي يتفاعل معه المستخدمون (مثل التصميم، الأزرار، النماذج).
- **الواجهة الخلفية :(Backend)** الجزء المسؤول عن معالجة البيانات، قواعد البيانات، والخوادم.
- **قواعد البيانات :(Database)** تخزين واسترجاع البيانات.
- **إدارة الخادم :(Server & Deployment)** نشر التطبيق على السحابة أو الخوادم.

.3المهارات المطلوبة لتصبحFull Stack Developer

أ. مهارات الواجهة الأمامية(Frontend)

١. **:HTML & CSS**لبناء هيكل الصفحات وتنسيقها.
٢. **:JavaScript (ES6+)**لإضافة التفاعل الديناميكي.
٣. **أطر عمل Frontend مثل:**
   - React.js
   - Angular
   - Vue.js
٤. **أدوات التصميم والتجربة:(UX/UI)**
   -) Figma, Adobe XDلتصميم الواجهات.(
   -) Tailwind CSS, Bootstrapلتصميم سريع.(

ب. مهارات الواجهة الخلفية(Backend)

١. **لغة برمجة خلفية مثل:**
   - JavaScript (Node.js)
   - Python (Django, Flask)
   - Java (Spring Boot)
   - PHP (Laravel)
٢. **قواعد البيانات:(SQL & NoSQL)**
   -) MySQL, PostgreSQLقواعد بيانات علائقية.(

- MongoDB, Firebase) قواعد بيانات غير علائقية.(

٣. **APIsوتقنيات التواصل:**

- RESTful APIs
- GraphQL
- WebSockets) للتواصل في الوقت الحقيقي.(

ج. مهارات DevOps والنشر

١. **أنظمة التحكم بالإصدار:(Version Control)**

- Git & GitHub/GitLab

٢. **نشر التطبيقات:(Deployment)**

- Docker, Kubernetes) للحاويات.(
- AWS, Azure, Heroku) للنشر السحابي.(

٣. **إدارة الخوادم:**

- Linux, Nginx, Apache.

Full Stack Software Engineer

Full Stack Software Engineerهو محترف تقني يمتلك:

- القدرة على تصميم وتنفيذ **أنظمة برمجية كاملة** من البداية إلى النهاية
- المعرفة العميقة بجميع طبقات التطويرFrontend) ، Backend، DevOps، وقواعد البيانات(
- المهارات الهندسية لحل المشكلات المعقدة واتخاذ القرارات المعمارية

.3المهارات التقنية المتقدمة المطلوبة

3.1مهارات هندسية أساسية

١. **هندسة البرمجيات**:

- أنماط التصميم(Design Patterns)

- مبادئSOLID
- الخوارزميات وهياكل البيانات

٢. **هندسة النظم**:

- تصميم الأنظمة الموزعة
- تحجيم الأنظمة(Scaling)
- تحسين الأداء

3.2التقنيات الحديثة

- **الحوسبة السحابية المتقدمة**:
  - Kubernetesو Docker Swarm
  - Serverless Architecture
  - Microservices Deployment
- **أمان المعلومات**:
  - OAuth 2.0
  - JWT
  - تشفير البيانات

.4العمليات والمنهجيات الهندسية

4.1دورة حياة التطوير

١. **التخطيط الهندسي**:

- تحليل المتطلبات
- تصميم العمارة
- نمذجة البيانات

٢. **التنفيذ**:

- التطوير الموجه بالاختبار(TDD)
- التكامل المستمر(CI/CD)

٣. **الصيانة**:

- المراقبة(Monitoring)
- تحليل الأداء
- التحسين المستمر

4.2منهجيات العمل

- Agile/Scrumالمتقدم
- DevOps Culture
- Site Reliability Engineering (SRE)

Full Stack Website

موقع الويب المتكامل هو تطبيق ويب يشمل:

- **واجهة المستخدم (Frontend)** التي يتفاعل معها الزوار
- **الخلفية (Backend)** التي تتعامل مع البيانات والعمليات المنطقية
- **قاعدة البيانات** لتخزين المعلومات
- **البنية التحتية** التي تستضيف التطبيق

3.المكونات الأساسية لموقع ويب متكامل

3.1واجهة المستخدم(Frontend)

١. **الهيكل الأساسي**:
   - HTML5
   - CSS3/SASS
   - JavaScript (ES6+)
٢. **أطر العمل الحديثة**:
   - React.js
   - Angular
   - Vue.js
٣. **أدوات التصميم**:
   - Webpack
   - Babel
   - Tailwind CSS

3.2الخلفية(Backend)

١. **لغات البرمجة**:
   - Node.js
   - Python (Django/Flask)
   - PHP (Laravel)
   - Java (Spring)
٢. **المهام الأساسية**:
   - معالجة الطلبات
   - إدارة الجلسات
   - تنفيذ العمليات المنطقية

3.3قاعدة البيانات

١. **قواعد بيانات**:**SQL**
   - MySQL
   - PostgreSQL
   - Microsoft SQL Server
٢. **قواعد بيانات**:**NoSQL**
   - MongoDB
   - Firebase
   - Cassandra

3.4البنية التحتية

١. **الخوادم**:
   - Apache
   - Nginx
   - Microsoft IIS
٢. **الحلول السحابية**:
   - AWS
   - Google Cloud
   - Microsoft Azure

.4عملية تطوير موقع ويب متكامل

4.1مرحلة التخطيط

- تحليل المتطلبات
- تصميم واجهة المستخدم
- تصميم بنية البيانات

4.2مرحلة التطوير

١. **تطوير الواجهة الأمامية**:
   - بناء المكونات
   - تطبيق التصميم
   - إضافة التفاعل
٢. **تطوير الخلفية**:
   - إنشاء واجهاتAPI
   - تنفيذ العمليات المنطقية
   - ربط قاعدة البيانات

4.3مرحلة النشر

- إعداد الخادم
- تكوين قاعدة البيانات
- نشر التطبيق
- ضبط الأمان