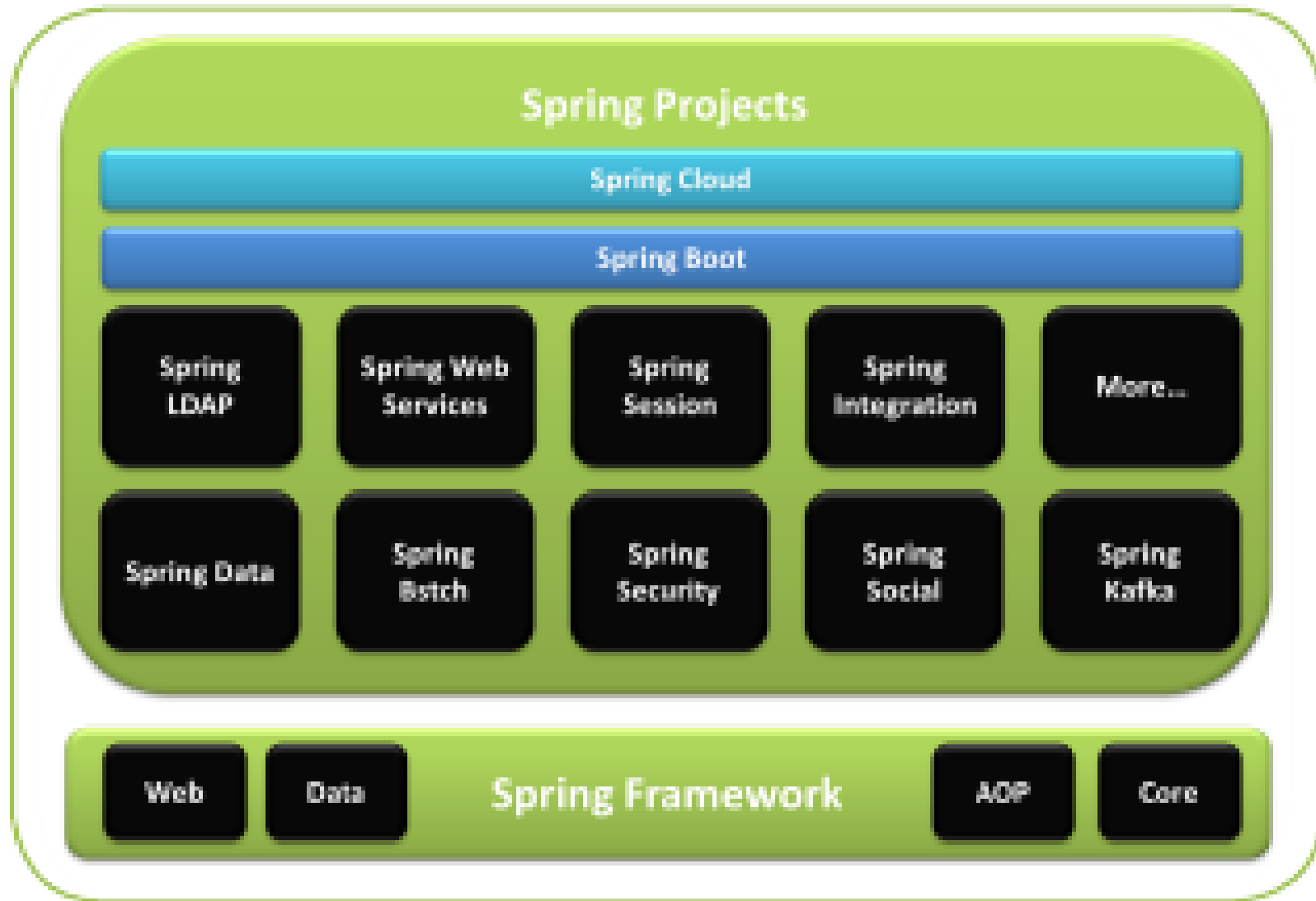
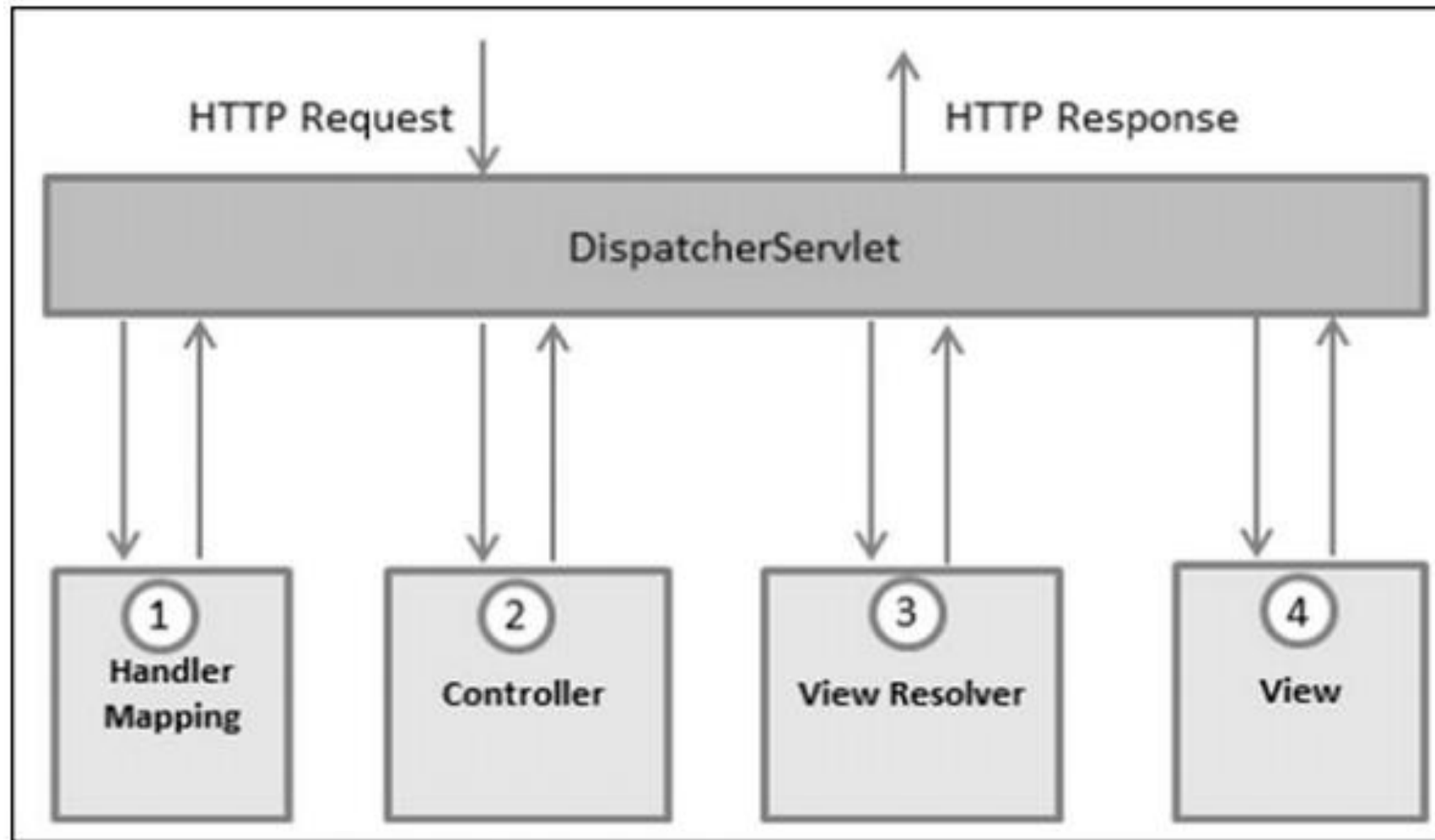


Spring Web MVC

Проекти Spring



Spring Web MVC



Створення Spring MVC в IntelliJ IDEA 2022

- ▶ Створити Web-application
- ▶ Додати залежність для Spring-webmvc
- ▶ Зареєструвати Dispatcher-servlet в Web-xml
- ▶ Створити Spring-конфігурацію за іменем сервлета
- ▶ Розмітити ViewResolver
- ▶ Додати контролер
- ▶ Реалізувати логіку в контролері і додати відповідні сторінки

XML-конфігурація Spring MVC

```
<servlet>
    <servlet-name>springDispatcher</servlet-name>
<servlet-class>org.springframework.web.servlet.
    DispatcherServlet
</servlet-class>
    <init-param>
        <param-name>contextConfigLocation</param-name>
        <param-value>/WEB-INF/servletContext.xml
        </param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
    <servlet-name>springDispatcher</servlet-name>
    <url-pattern>/</url-pattern>
</servlet-mapping>
```

XML-конфігурація Spring MVC

```
<context-param>
  <param-name>contextConfigLocation</param-name>
  <param-value>/WEB-INF/rootContext.xml
</param-value>
</context-param>
<listener>
  <listener-class>org.springframework.web.context.
    ContextLoaderListener
  </listener-class>
</listener>
<servlet>
  <servlet-name>springDispatcher</servlet-name>
  <servlet-class>org.springframework.web.servlet.
    DispatcherServlet
  </servlet-class>
```

XML-конфігурація Spring MVC

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:context="http://www.springframework.org/schema/context"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/context
https://www.springframework.org/schema/context/spring-context.xsd">

    <context:component-scan base-package = "net.starbasic.springmvc" />

    <bean id="viewResolver"
          class = "org.springframework.web.servlet.view.InternalResourceViewResolver">
        <property name = "prefix" value = "/views/" />
        <property name = "suffix" value = ".jsp" />
    </bean>

</beans>
```

Контролер

1. @Controller
2. @RequestMapping
3. @PathVariable
4. @RequestParam
5. @ModelAttribute
6. @RequestBody and @ResponseBody
7. @RequestHeader and @ResponseHeader

Методи контролера

```
@GetMapping(path = {"/","/home-page"})  
public String homePage(){  
    return "page1";  
}
```

```
@RequestMapping(value = "/own-product", method = RequestMethod.GET)  
public ModelAndView showProduct(){  
    ModelAndView result = new ModelAndView("page3");  
    result.addObject("product",  
        new Product("Ручка","Пуркер",300));  
    return result;  
}
```

Методи контролера

```
@RequestMapping(value = "/add-product",
    method = RequestMethod.GET)
    public String addProductForm(){
return "page4";
    }
    @RequestMapping(value = "/add-product",
        method = RequestMethod.POST,
        consumes = {"text/plain; charset=UTF-8", "application/*; charset=UTF-8"})
    public ModelAndView addProduct(Product product) {
        ModelAndView result = new ModelAndView("page3");
        result.addObject("product", new Product(product));
        return result;
    }
```

Методи контролера

```
@RequestMapping(value = "/student", method = RequestMethod.GET)  
public ModelAndView student() {
```

```
    return new ModelAndView("student", "command", new Student());  
}
```

```
@RequestMapping(value = "/addStudent", method = RequestMethod.POST)
```

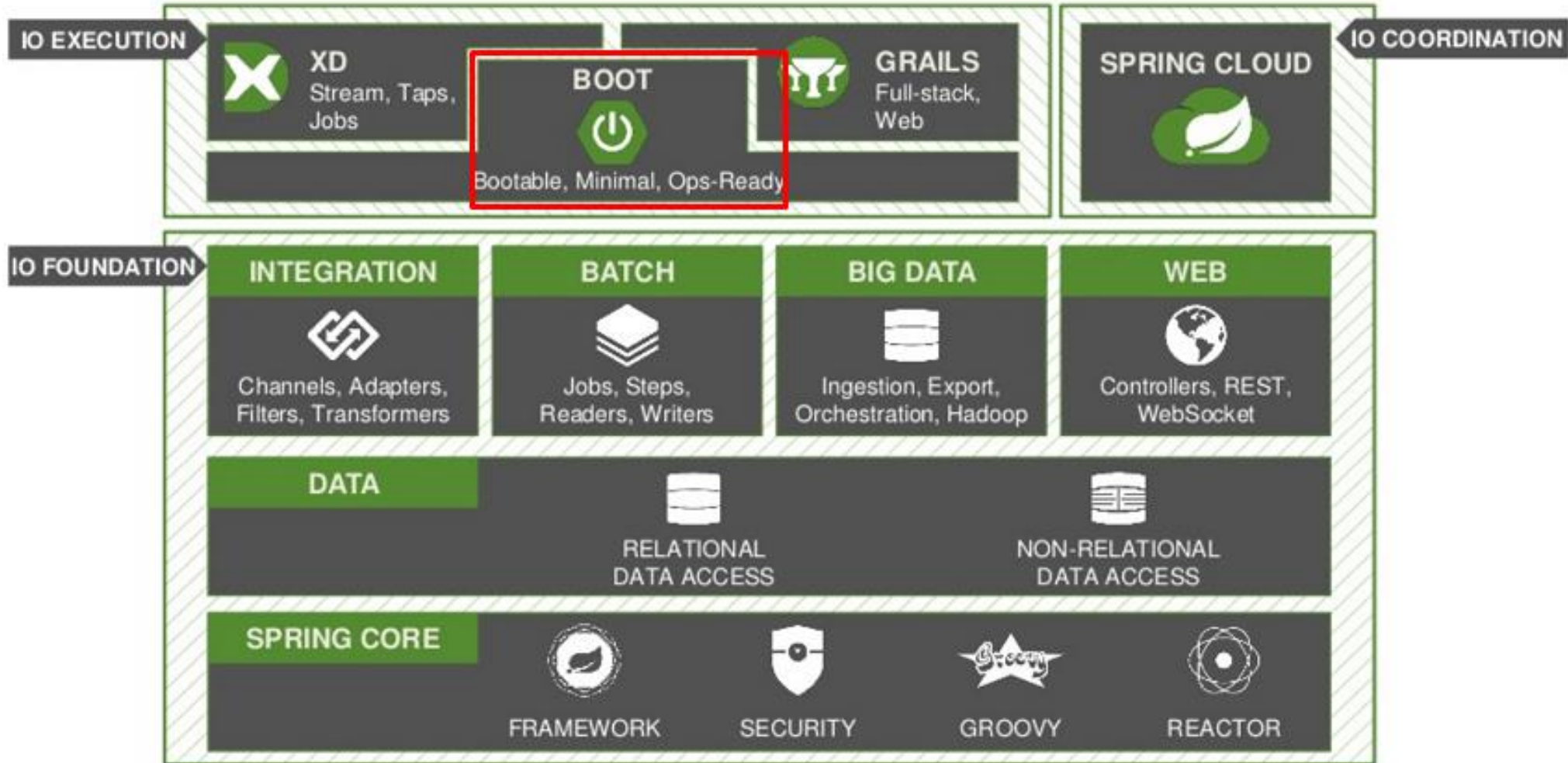
```
    public String addStudent(Student student,  
                             ModelMap model) {  
        model.addAttribute("name", student.getName());  
        model.addAttribute("age", student.getAge());  
        model.addAttribute("id", student.getId());
```

```
        return "result";  
    }
```

Що таке Spring Boot?

- Spring Boot - це модуль Spring Framework, що забезпечує функцію швидкої розробки додатків RAD (Rapid Application Development).
- Spring Boot використовується для створення автономних програм на основі Spring.
- Spring Boot не вимагає конфігурації XML
- Це стандарт для конфігурації проектування програмного забезпечення, що підвищує продуктивність розробників.

Що таке Spring Boot?



Spring Boot Application

- Використовує сценарії збірки Maven або Graddle.
- Об'єднує залежності в зручні стартер-пакети для окремих задач: web, jdbc, jpa, і т.п.
- Головний клас `@SpringBootApplication` запускає контекст Spring, скануючи при цьому компоненти з класів конфігурації власного пакета.
- Поверх контексту Spring Boot можна запускати Java Web Server Tomcat (або Jetty)
- Структуру проекту можна задати в IntelliJ Idea Ultimate, або на сайті start.spring.io і розробляти з використанням IntelliJ Idea Community, чи іншої IDE

Spring Boot Application

Spring Boot за замовчуванням використовує View Engine Thymleaf(є і інші варіанти), але можна налаштувати проект з використанням JSP. Для цього потрібно:

- Створити проект Spring Boot з spring-boot-starter-web пакетом.
- Додати в структуру проекту папку webapp, за потреби WEB-INF.
- Додати залежності для парсингу і хостингу jsp: servlet-api, jasper, jstl і т.п.
- Вказати шлях до jsp-сторінок в application.properties, наприклад:

`spring.mvc.view.prefix=/views/`

`spring.mvc.view.suffix=.jsp`