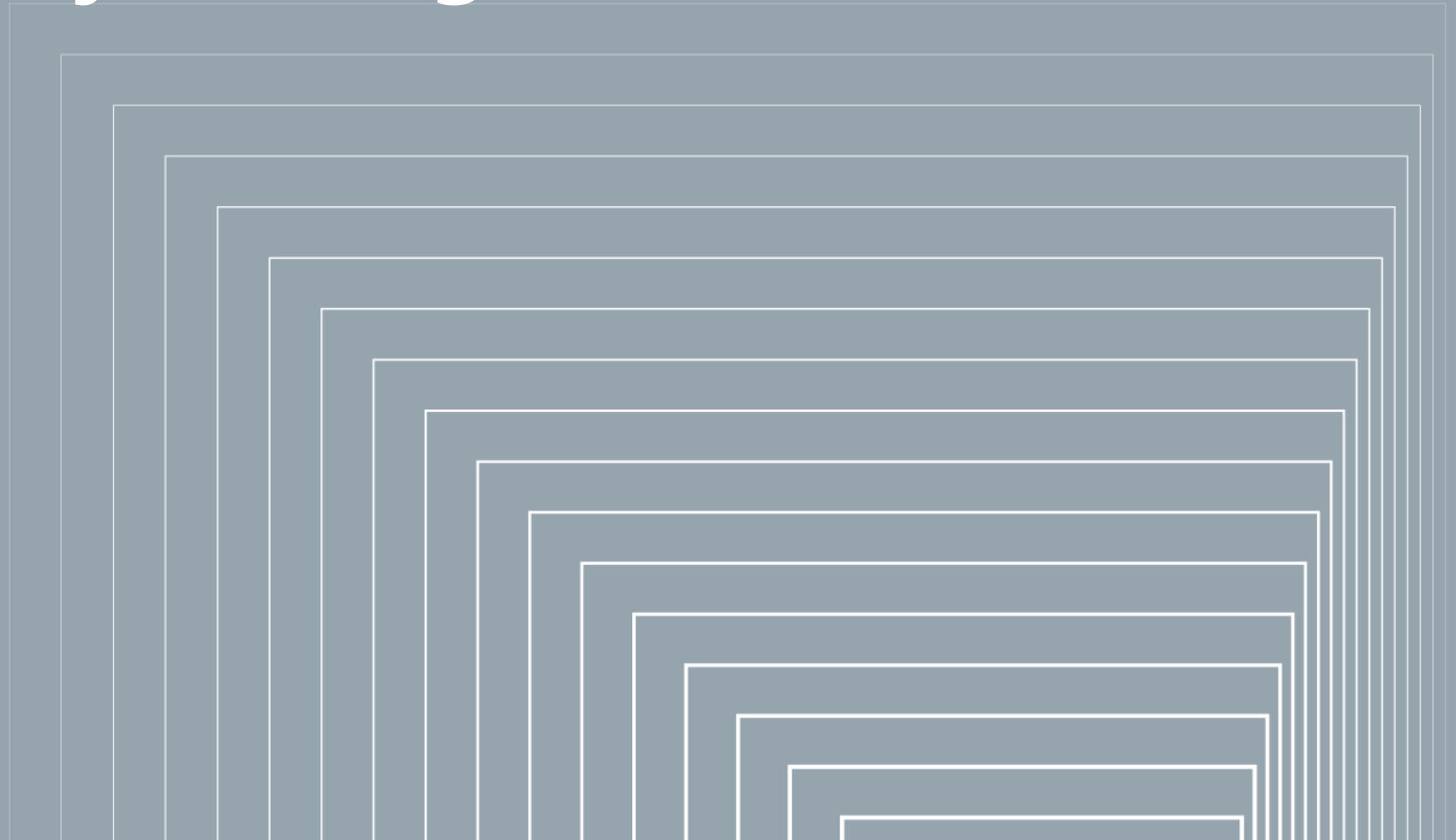
NHN Academy

5. Dependency Management and Main Function



Dependency management

Spring Boot Starter

- Spring Framework 관련 기술을 사용하기 위한 의존성 관리 세트
- 40개 이상의 Spring Boot starter를 Spring Boot에서 제공
- 3rd Party 에서 제공

Starter 이름	설명
spring-boot-starter-parent	spring boot 프로젝트에서 상속 받아야 할 pom
spring-boot-starter	Auto Configuration 을 포함한 핵심 starter, logging, yaml 지원
spring-boot-starter-web	RESTful, Web 애플리케이션 구축을 위한 starter, 내장 tomcat 포함
spring-boot-starter-amqp	Spring AMQP, Rabbit MQ 사용을 위한 설정
spring-boot-starter-mail	Java mail 을 사용하기 위한 설정, spring framework의 메일 발송기능

Dependency management

Spring Boot Starter

· Pivotal Software사의 공식 starter는 spring-boot-starter-* 패턴으로 명명한다.
· spring-boot-starter-* 의 라이브러리 의존성을 추가하는 것 만으로도 기본 설정으로 기능이 동작한다.
· 공식 starter가 아닌 경우는 spring-boot로 시작하지 않아야 한다. 보통 {function}-spring-boot-starter 과 같이 명명

Dependency management

spring-boot-starter-parent

- spring-boot-starter-parent는 spring-boot-dependencies를 상속
- spring-boot 버전별로 지원하는 라이브러리 의존성 목록(Bills of Materials)
- spring-boot 버전을 업그레이드하면 라이브러리 의존성도 모두 자동 업그레이드

```
<parent>
 <groupId>org.springframework.boot</groupId>
 <artifactId>spring-boot-dependencies</artifactId>
 <version>2.6.0</version>
 <relativePath>../../spring-boot-dependencies</relativePath>
</parent>
```

Dependency management

spring-boot-dependencies

- 사용하는 라이브러리의 버전을 property 로 관리

```
<properties>
<activemq.version>5.15.8</activemq.version>
<antlr2.version>2.7.7</antlr2.version>
<appengine-sdk.version>1.9.71</appengine-sdk.version>
<artemis.version>2.6.4</artemis.version>
<aspectj.version>1.9.2</aspectj.version>
<assertj.version>3.11.1</assertj.version>
...
</properties>
```

Dependency management

spring-boot-dependencies

· dependencyManagement 로 사용할 라이브러리의 버전을 미리 지정

```
<dependencymanagement>
  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot</artifactId>
      <version>2.1.3.RELEASE</version>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-web</artifactId>
      <version>2.1.3.RELEASE</version>
    </dependency>
...
  </dependencies>
</dependencymanagement>
```

Dependency management

spring-boot-starter-web

· spring-core, spring-web, spring-webmvc, 내장 tomcat 서버 및 관련 라이브러리 설정을 일괄처리

```
<dependencies>
  <dependency>
    <groupid>org.springframework.boot</groupid>
    <artifactid>spring-boot-starter</artifactid>
    <version>2.1.3.RELEASE</version>
    <scope>compile</scope>
  </dependency>
  <dependency>
    <groupid>org.springframework.boot</groupid>
    <artifactid>spring-boot-starter-json</artifactid>
    <version>2.1.3.RELEASE</version>
    <scope>compile</scope>
  </dependency>
```

Main Class Function

spring-boot 프로젝트의 main 메소드

· public static void main 메서드에서 SpringApplication.run을 실행시킨다.
· args는 command 라인에서 보낸 인자를 전달한다. (--debug, --spring.profiles.active)

SpringApplication 실행

Main Class Function

Static 메서드

- 일반적인 사용법

```
@SpringBootApplication
public class StudentApplication {

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

}
```

Main Class Function

생성자 사용

- static 메소드 내부에 동일한 구현이 있다.

```
@SpringBootApplication
public class StudentApplication {

    public static void main(String[] args) {
        SpringApplication application = new SpringApplication(StudentApplication.class);
        application.run(args);
    }

}
```

Main Class Function

NHN Academy

builder 사용

· 빌더로 여러 개의 web context를 구성할 수 있으며 parent-child 의 계층구조로 설정가능

```
@SpringBootApplication
public class StudentApplication {

   public static void main(String[] args) {
      new SpringApplicationBuilder()
         .sources(Student.class).web(WebApplicationType.NONE)
         .child(FirstChildConfig.class).web(WebApplicationType.SERVLET)
         .sibling(SecondChildConfig.class).web(WebApplicationType.SERVLET)
         .run(args);
   }

}
```

Custom Banner

spring-boot 프로젝트의 banner custom

```
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::   v2.6.2

2021-02-03 10:33:25.224  INFO 17321 --- [           main] o.s.b.d.s.s.SpringApplicationExample    : Starting Spr
2021-02-03 10:33:25.226  INFO 17900 --- [           main] o.s.b.d.s.s.SpringApplicationExample    : No active pr
2021-02-03 10:33:26.046  INFO 17321 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat init
2021-02-03 10:33:26.054  INFO 17900 --- [           main] o.apache.catalina.core.StandardService   : Starting se
2021-02-03 10:33:26.055  INFO 17900 --- [           main] org.apache.catalina.core.StandardEngine  : Starting Se
2021-02-03 10:33:26.097  INFO 17900 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializir
2021-02-03 10:33:26.097  INFO 17900 --- [           main] w.s.c.ServletWebServerApplicationContext : Root WebApp
2021-02-03 10:33:26.144  INFO 17900 --- [           main] s.tomcat.SampleTomcatApplication         : ServletCont
2021-02-03 10:33:26.376  INFO 17900 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat star
2021-02-03 10:33:26.384  INFO 17900 --- [           main] o.s.b.d.s.s.SpringApplicationExample    : Started Samp
```

Custom Banner

시작코드 수정

· org.springframework.boot.Banner 인터페이스를 구현하여 custom banner 개발

```
@SpringBootApplication
public class MyApplication {

  public static void main(String[] args) {
    SpringApplication application = new SpringApplication(MyApplication.class);
    application.setBannerMode(Banner.Mode.OFF);
    application.run(args);
  }

}
```

Custom Banner

Resource 파일

- src/main/resource/banner.txt 파일을 생성하여 기본 banner 대체
- banner.gif, banner.jpg, banner.png 를 src/main/resources 에 복사

Custom Banner

Banner에 사용할 수 있는 변수

· banner.txt 에는 다음의 변수를 사용할 수 있다.

Starter 이름	설명
${application.version}	MANIFEST.MF 에 설정한 애플리케이션의 버전
${application.formattd-version}	${application.version} 을 (v1.0.0) 형태로 포멧
${spring-boot.version}	사용하는 spring-boot 의 버전
${spring-boot.formatted-version}	${spring-boot.version} 을 (v1.0.0) 형태로 포멧
${Ansi.NAME} ..	Ansi escape 코드의 이름을 지정
${application.title}	MANIFEST.MF 에 설정한 애플리케이션 이름

[실습] tomcat 을 jetty 로 실행하도록 변경

NHN Academy

목표

· 계좌 정보를 제공하는 API 를 jetty 로 동작하도록 합니다.

힌트

· spring-boot-starter-web 에서 spring-boot-starter-tomcat 을 exclude 한다.
· spring-boot-starter-jetty 의 의존성을 추가한다.

예상시간

· 5분

NHN Academy

Q&A

NHN Academy

감사합니다