배포방식에 따라서 ant 🡺 maven 🡺 grandle 을 이용한다.

프로젝트 생성방법

File 🡺 New 🡺 Spring Legacy Project

팝업창에서 Project Name을 입력하고 Simple Spring Maven을 선택한 후 Finish 버튼을 누른다.

파일구성

scr/main/java 🡺 ………java 파일

scr/main/resource 🡺 ………xml 파일

Spring 에서는 java 파일들을 일반적으로 빈이라고 부른다.

빈 등록 방법

resource에서 우측 클릭 new 🡺 other 🡺 xml file을 선택

팝업창에서 이름을 입력하고 Finish를 클릭

하단 탭에서 source를 선택

applicationContexts.xml

을 만들 때 new 🡺 Spring Bean Configuration을 선택하여 wizard로 생성한다.

프로젝트 실행 전 반드시 update를 해야 한다.

project에서 우측 클릭한 후 Maven 🡺 update Project를 클릭한다.

팝업창 발생 후 OK 버튼을 클릭하면 우측 하단에 업데이트 현황이 표시된다.

모든 update가 완료된 후 실행한다.

**<정리>**

1. Class를 정의
2. 빈으로 등록한다.
3. 실행 (스프링 컨테이너를 활용)

GenericXmlApplicationContext

AnnotationConfigApplicationContext

\*pom.xml : 배포를 위하여 라이브러리를 등록하는 곳

* **DI (Dependency Injection) 객체의존주입**
* **IOC**
* **AOP**

|  |
| --- |
| use  MemberPrint  use  MemberListPrinter  **의존주입**  use  ChangePasswordService  MemberDAO  Assembler  use  **의존주입**  Member  C.P메서스를 보유  use  use  R.R  비번 재확인  MemberRegisterService |

Spring IOC 중

XML방식으로 변경

POJO방식으로 변경

<constructor-arg> 는 bean을 설정할 때 생성자를 이용해서 의존 객체를 주입할 때 사용한다.

ref 로 인자값을 삽입하는데, 이때 bean에 설정한 id 값을 사용한다.

new a@a.com a 123 123

new [b@b.com](mailto:b@b.com) b 123 123

@Autowired 기능을 사용하기 위해서는

* Import 구문추가
* 기본생성자가 반드시 있어야 한다.

Spring container 만들기

* @Bean태그를 붙여야 한다.
* 매소드 이름은 아이디값을 사용한다.
* 의존주입은 매서드를 생성할 때 해당 메서드를 직접 호출한다.

JSP에서의 JDBC는 아래와 같은 명령문을 사용한다.

- preparestatement

- statement

- executeQuery()

- executeUpdate()

Spring에서의 JDBC는 아래와 같은 명령문을 사용한다.

* query()
* queryforObjet()
* update()

🡺 (쿼리문(select만 사용) , RowMapper, 쿼리문 내 변수를 사용하는 인수 ?의 갯수에 따라서 이하 수량이 늘어날 수 있음 )

🡺 update는 insert와 delete, update를 함께 사용한다.

**CRUD**

Insert

Select

Update

delete