Lombok이란? 및 Lombok 활용법

Lombok의 필요성

기존 코드

```
public class Store extends Common {

    private String companyName;                                    // 상호명
    private String industryTypeCode;                               // 업종코드
    private String businessCodeName;                               // 업태명
    private String industryName;                                   // 업종명(종
목명)
    private String telephone;                                      // 전화번호
    private String regionMoneyName;                                // 사용가능한
지역화폐 명
    private boolean isBmoneyPossible;                              // 지류형 지
역화폐 사용가능 여부
    private boolean isCardPossible;                                // 카드형 지
역화폐 사용가능 여부
    private boolean isMobilePossible;                              // 모바일형
지역화폐 사용가능 여부
    private String lotnoAddr;                                      // 소재지 지
번주소
    private String roadAddr;                                       // 소재지 도
로명주소
    private String zipCode;                                        // 우편번호
    private double longitude;                                      // 경도
    private double latitude;                                       // 위도
    private String sigunCode;                                      // 시군 코드
    private String sigunName;                                      // 시군 이름

    public String getCompanyName() {
        return companyName;
    }

    public void setCompanyName(String companyName) {
        this.companyName = companyName;
    }

    public String getIndustryTypeCode() {
        return industryTypeCode;
    }

    public void setIndustryTypeCode(String industryTypeCode) {
        this.industryTypeCode = industryTypeCode;
    }

    public String getBusinessCodeName() {
        return businessCodeName;
```

```
    }

    public void setBusinessCodeName(String businessCodeName) {
        this.businessCodeName = businessCodeName;
    }

    public String getIndustryName() {
        return industryName;
    }

    public void setIndustryName(String industryName) {
        this.industryName = industryName;
    }

    public String getTelephone() {
        return telephone;
    }

    public void setTelephone(String telephone) {
        this.telephone = telephone;
    }

    public String getRegionMoneyName() {
        return regionMoneyName;
    }

    public void setRegionMoneyName(String regionMoneyName) {
        this.regionMoneyName = regionMoneyName;
    }

    public boolean isBmoneyPossible() {
        return isBmoneyPossible;
    }

    public void setBmoneyPossible(boolean bmoneyPossible) {
        isBmoneyPossible = bmoneyPossible;
    }

    public boolean isCardPossible() {
        return isCardPossible;
    }

    public void setCardPossible(boolean cardPossible) {
        isCardPossible = cardPossible;
    }

    public boolean isMobilePossible() {
        return isMobilePossible;
    }

    public void setMobilePossible(boolean mobilePossible) {
        isMobilePossible = mobilePossible;
    }
```

```
    public String getLotnoAddr() {
        return lotnoAddr;
    }

    public void setLotnoAddr(String lotnoAddr) {
        this.lotnoAddr = lotnoAddr;
    }

    public String getRoadAddr() {
        return roadAddr;
    }

    public void setRoadAddr(String roadAddr) {
        this.roadAddr = roadAddr;
    }

    public String getZipCode() {
        return zipCode;
    }

    public void setZipCode(String zipCode) {
        this.zipCode = zipCode;
    }

    public double getLongitude() {
        return longitude;
    }

    public void setLongitude(double longitude) {
        this.longitude = longitude;
    }

    public double getLatitude() {
        return latitude;
    }

    public void setLatitude(double latitude) {
        this.latitude = latitude;
    }

    public String getSigunCode() {
        return sigunCode;
    }

    public void setSigunCode(String sigunCode) {
        this.sigunCode = sigunCode;
    }

    public String getSigunName() {
        return sigunName;
    }

    public void setSigunName(String sigunName) {
        this.sigunName = sigunName;
```

```
    }

}
```

Lombok 이란?

- 어노테이션 기반으로 코드를 자동완성
- Getter, Setter, Equals, ToString 등 다양한 방면의 코드를 자동완성

```
@Getter
@Setter
public class Store extends Common {

    private String companyName;                                    // 상호명
    private String industryTypeCode;                               // 업종코드
    private String businessCodeName;                               // 업태명
    private String industryName;                                   // 업종명(종
목명)
    private String telephone;                                      // 전화번호
    private String regionMoneyName;                                // 사용가능한
지역화폐 명
    private boolean isBmoneyPossible;                              // 지류형 지
역화폐 사용가능 여부
    private boolean isCardPossible;                                // 카드형 지
역화폐 사용가능 여부
    private boolean isMobilePossible;                              // 모바일형
지역화폐 사용가능 여부
    private String lotnoAddr;                                      // 소재지 지
번주소
    private String roadAddr;                                       // 소재지 도
로명주소
    private String zipCode;                                        // 우편번호
    private double longitude;                                      // 경도
    private double latitude;                                       // 위도
    private String sigunCode;                                      // 시군 코드
    private String sigunName;                                      // 시군 이름

}
```

Lombok의 장점

- 어노테이션 기반의 코드 생성을 통한 생산성 향상
- 반복되는 코드 다이어트를 통한 가독성 및 유지보수성 향상
- Getter, Setter 외에 빌더 패턴이나 로그 생성 등 다양한 방면으로 활용 가능

Lombok의 기능 및 사용 예제

@Getter @Setter

- 클래스 이름 위 : 모든 변수들에 적용
- 변수 이름 위 : 해당 변수에만 적용

```
@Getter
public class Store extends Common {

    @Setter
    private String companyName;                                    // 상호명
    private String industryTypeCode;                               // 업종코드
    private String businessCodeName;                               // 업태명
    private String industryName;                                   // 업종명(종
목명)
    private String telephone;                                      // 전화번호
    private String regionMoneyName;                                // 사용가능한
지역화폐 명
    private boolean isBmoneyPossible;                              // 지류형 지
역화폐 사용가능 여부
    private boolean isCardPossible;                                // 카드형 지
역화폐 사용가능 여부
    private boolean isMobilePossible;                              // 모바일형
지역화폐 사용가능 여부
    private String lotnoAddr;                                      // 소재지 지
번주소
    private String roadAddr;                                       // 소재지 도
로명주소
    private String zipCode;                                        // 우편번호
    private double longitude;                                      // 경도
    private double latitude;                                       // 위도
    private String sigunCode;                                      // 시군 코드
    private String sigunName;                                      // 시군 이름

}
```

- 모든 변수에 Getter
- companyName에만 Setter

@AllArgsConstructor

- AllArgsConstructor는 모든 변수를 사용하는 생성자 자동완성

```
@Getter
@AllArgsConstructor
public class Store extends Common {

    private String companyName;                                    // 상호명
    private String industryTypeCode;                               // 업종코드
    private String businessCodeName;                               // 업태명
    private String industryName;                                   // 업종명(종
목명)
    private String telephone;                                      // 전화번호
```

```
    private String regionMoneyName;                                    // 사용가능한
지역화폐 명
    private boolean isBmoneyPossible;                                  // 지류형 지
역화폐 사용가능 여부
    private boolean isCardPossible;                                    // 카드형 지
역화폐 사용가능 여부
    private boolean isMobilePossible;                                  // 모바일형
지역화폐 사용가능 여부
    private String lotnoAddr;                                          // 소재지 지
번주소
    private String roadAddr;                                           // 소재지 도
로명주소
    private String zipCode;                                            // 우편번호
    private double longitude;                                          // 경도
    private double latitude;                                           // 위도
    private String sigunCode;                                          // 시군 코드
    private String sigunName;                                          // 시군 이름

    /* AllArgsConstructor를 통해 아래의 생성자를 자동 생성할 수 있다.
    public Store(String companyName, String industryTypeCode, String
businessCodeName, String industryName, String telephone,
    String regionMoneyName, boolean isBmoneyPossible, boolean
isCardPossible, boolean isMobilePossible, String lotnoAddr,
    String roadAddr, String zipCode, double longitude, double latitude,
String sigunCode, String sigunName) {
        this.companyName = companyName;
        this.industryTypeCode = industryTypeCode;
        this.businessCodeName = businessCodeName;
        this.industryName = industryName;
        this.telephone = telephone;
        this.regionMoneyName = regionMoneyName;
        this.isBmoneyPossible = isBmoneyPossible;
        this.isCardPossible = isCardPossible;
        this.isMobilePossible = isMobilePossible;
        this.lotnoAddr = lotnoAddr;
        this.roadAddr = roadAddr;
        this.zipCode = zipCode;
        this.longitude = longitude;
        this.latitude = latitude;
        this.sigunCode = sigunCode;
        this.sigunName = sigunName;
    }
    */
}
```

@NoArgsConstructor

- 어떤 변수도 사용하지않는 기본 생성자를 자동완성

```
@Getter
@NoArgsConstructor
```

```
public class Store extends Common {

    private String companyName;                                    // 상호명
    private String industryTypeCode;                               // 업종코드
    private String businessCodeName;                               // 업태명
    private String industryName;                                   // 업종명(종
목명)
    private String telephone;                                      // 전화번호
    private String regionMoneyName;                                // 사용가능한
지역화폐 명
    private boolean isBmoneyPossible;                              // 지류형 지
역화폐 사용가능 여부
    private boolean isCardPossible;                                // 카드형 지
역화폐 사용가능 여부
    private boolean isMobilePossible;                              // 모바일형
지역화폐 사용가능 여부
    private String lotnoAddr;                                      // 소재지 지
번주소
    private String roadAddr;                                       // 소재지 도
로명주소
    private String zipCode;                                        // 우편번호
    private double longitude;                                      // 경도
    private double latitude;                                       // 위도
    private String sigunCode;                                      // 시군 코드
    private String sigunName;                                      // 시군 이름

    /* NoArgsConstructor를 통해 아래의 생성자를 자동 생성할 수 있다.
    public Store() {

    }
    */
}
```

@RequiredArgsConstructor

- 특정 변수만 활용하는 생성자를 자동완성
- 생성자 인자로 추가할 변수에 @NonNull 또는 final

```
@Getter
@RequiredArgsConstructor
public class Store extends Common {

    @NonNull
    private String companyName;                                    // 상호명
    private final String industryTypeCode;                         // 업종코드
    private String businessCodeName;                               // 업태명
    private String industryName;                                   // 업종명(종
목명)
    private String telephone;                                      // 전화번호
    private String regionMoneyName;                                // 사용가능한
```

```
지역화폐 명
    private boolean isBmoneyPossible;                                   // 지류형 지
역화폐 사용가능 여부
    private boolean isCardPossible;                                     // 카드형 지
역화폐 사용가능 여부
    private boolean isMobilePossible;                                   // 모바일형
지역화폐 사용가능 여부
    private String lotnoAddr;                                           // 소재지 지
번주소
    private String roadAddr;                                            // 소재지 도
로명주소
    private String zipCode;                                             // 우편번호
    private double longitude;                                           // 경도
    private double latitude;                                            // 위도
    private String sigunCode;                                           // 시군 코드
    private String sigunName;                                           // 시군 이름

    /* RequiredArgsConstructor 통해 아래의 생성자를 자동 생성할 수 있다.
    public Store(String companyName, String industryTypeCode) {
        this.companyName = companyName;
        this.industryTypeCode = industryTypeCode;
    }
    */

}
```

@EqualsAndHashCode

- equals함수와 hashcode 자동 생성

```
@RequiredArgsConstructor
@EqualsAndHashCode(of = {"companyName", "industryTypeCode"}, callSuper =
false))
public class Store extends Common {

    @NonNull
    private String companyName;                                         // 상호명
    @NonNull
    private String industryTypeCode;                                    // 업종코드
    private String businessCodeName;                                    // 업태명
    private String industryName;                                        // 업종명(종
목명)
    private String telephone;                                           // 전화번호
    private String regionMoneyName;                                     // 사용가능한
지역화폐 명
    private boolean isBmoneyPossible;                                   // 지류형 지
역화폐 사용가능 여부
    private boolean isCardPossible;                                     // 카드형 지
역화폐 사용가능 여부
    private boolean isMobilePossible;                                   // 모바일형
지역화폐 사용가능 여부
```

```
    private String lotnoAddr;                                   // 소재지 지
번주소
    private String roadAddr;                                    // 소재지 도
로명주소
    private String zipCode;                                     // 우편번호
    private double longitude;                                   // 경도
    private double latitude;                                    // 위도
    private String sigunCode;                                   // 시군 코드
    private String sigunName;                                   // 시군 이름

}
```

- A객체의 companyName과 IndustryTypeCode와 B객체의 companyName과 IndustryTypeCode가 동일하면 같은 객체

@Builder

- 객체의 생성에 Builder 패턴을 적용시켜 줌.
- 클래스 위에 선언 : 모든 변수들에 대해 build
- 생성자를 만들고 생성자 위에 선언

```
@Getter
@NoArgsConstructor
public class Store extends Common {

    private String companyName;                                 // 상호명
    private String industryTypeCode;                            // 업종코드
    private String businessCodeName;                            // 업태명
    private String industryName;                                // 업종명(종
목명)
    private String telephone;                                   // 전화번호
    private String regionMoneyName;                             // 사용가능한
지역화폐 명
    private boolean isBmoneyPossible;                           // 지류형 지
역화폐 사용가능 여부
    private boolean isCardPossible;                             // 카드형 지
역화폐 사용가능 여부
    private boolean isMobilePossible;                           // 모바일형
지역화폐 사용가능 여부
    private String lotnoAddr;                                   // 소재지 지
번주소
    private String roadAddr;                                    // 소재지 도
로명주소
    private String zipCode;                                     // 우편번호
    private double longitude;                                   // 경도
    private double latitude;                                    // 위도
    private String sigunCode;                                   // 시군 코드
    private String sigunName;                                   // 시군 이름

    @Builder
    public Store(String companyName, String industryTypeCode){
```

```
        this.companyName = companyName;
        this.industryTypeCode = industryTypeCode;
    }

}

@RestController
@RequestMapping(value = "/store")
@Log4j2
public class StoreController {

    @GetMapping(value = "/init")
    private ResponseEntity init(){
        Store store = Store.builder()
                .companyName("회사이름")
                .industryTypeCode("업종코드")
                .build();

        return ResponseEntity.ok(store);
    }

}
```