스프링시작하기

스프링 API 12장

박명회

1.1 application.yml

dev,prod 설정

```yaml
server:
  port: 8080
  servlet:
    context-path: /


spring:
  # todo 개발 데이터베이스 연결
  datasource:
    url: jdbc:h2:mem:test
    username: sa
    password:
    driver-class-name: org.h2.Driver


  jpa:
    hibernate:
      ddl-auto: create
    show-sql: true # 콘솔창에 sql 출력
    properties:
      hibernate:
        format_sql: true # sql 예쁘게 출력
        default_batch_fetch_size: 500 # 일대다 컬렉션 조회 시 성능 최적화
    open-in-view: false # 영속성 컨텍스트의 생존 범위를 트랜잭션 범위로 한정

  servlet:
    multipart:
      max-file-size: 10MB # 파일 업로드 요청 시 하나의 파일 크기를 10MB 제한
      max-request-size: 100MB # 파일 업로드 요청 시 모든 파일 크기합을 100MB 제한
```

1.2 application.yml

```yaml
logging:
  level:
    org.hibernate.type: trace # 콘솔창에 조건에 바인딩되는 값 및 조회 결과 출력
    com.app: debug # todo 패키지 수정
```

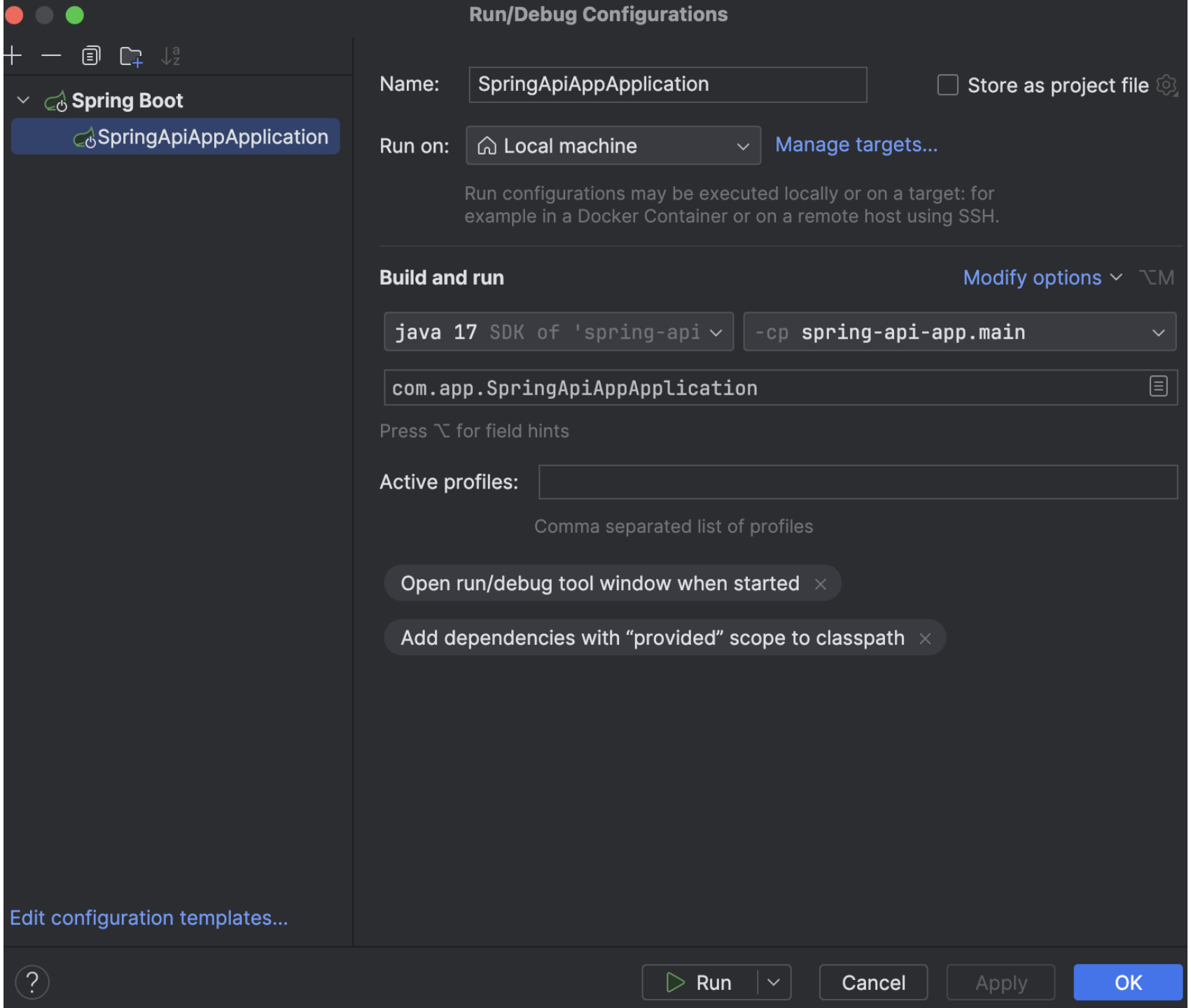
SQL 구문에 바인딩 된 값 처리

1.2 Environment 체크

```java
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.core.env.Environment;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;

@Controller
public class ProfileController {

    @Autowired
    private Environment environment;

    @GetMapping("/profile")
    public String showProfile(Model model) {
        String[] activeProfiles = environment.getActiveProfiles()
        model.addAttribute("profiles", activeProfiles);
        return "profile";
    }
}
```

1.3 프로젝트 생성

Edit Config



1.4 spring cloud openfeign

```
ext {
    set('springCloudVersion', "2023.0.0")
}
```

```
spring:
  cloud:
    openfeign:
      client:
        config:
          default:
            connectTimeout: 5000
            readTimeout: 5000
```

```
@Service
@RequiredArgsConstructor
public class ExampleService {
    1 usage
    private final ExampleClient exampleClient;

    1 usage
    public String getResource() {
        return exampleClient.getResource();
    }
}
```

```
package com.example.mhapi.mh;

import lombok.RequiredArgsConstructor;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequiredArgsConstructor
public class ExampleController {
    1 usage
    private final  ExampleService exampleService;

    @GetMapping("mh/test")
    public ResponseEntity<String> mhtest(){

        String result = exampleService.getResource();

        return ResponseEntity.ok(result);
    }

    @GetMapping("testjson")
    public ResponseEntity<String> testjson(){
        return ResponseEntity.ok( body: "gwegwe");
    }
}
```

1.4 spring cloud openfeign

```java
import org.springframework.cloud.openfeign.FeignClient;
import org.springframework.web.bind.annotation.GetMapping;

1 usage
@FeignClient(name = "test", url = "https://www.dhlottery.co.kr/common.do?method=getLottoNumber&drwNo=861")
public interface ExampleClient {

    1 usage
    @GetMapping("")
    String getResource();
}
```

https://www.dhlottery.co.kr/common.do?method=getLottoNumber&drwNo=861

1.4 spring cloud openfeign

```
2 usages
@Slf4j
public class FeignClientExceptionErrorDecoder implements ErrorDecoder {

    1 usage
    private ErrorDecoder errorDecoder = new Default();

    @Override
    public Exception decode(String methodKey, Response response) {
        log.error("{} 요청 실패, status : {}, reason : {}", methodKey, response.status(), response.reason());
        FeignException exception = FeignException.errorStatus(methodKey, response);
        HttpStatus httpStatus = HttpStatus.valueOf(response.status());
        if(httpStatus.is5xxServerError()) {
            return new RetryableException(
                    response.status(),
                    exception.getMessage(),
                    response.request().httpMethod(),
                    exception,
                    null,
                    response.request()
            );
        }
        return errorDecoder.decode(methodKey, response);
    }

}
```

1.4 실행결과

소스 참고

https://github.com/dron512/feignClient

고맙습니다.