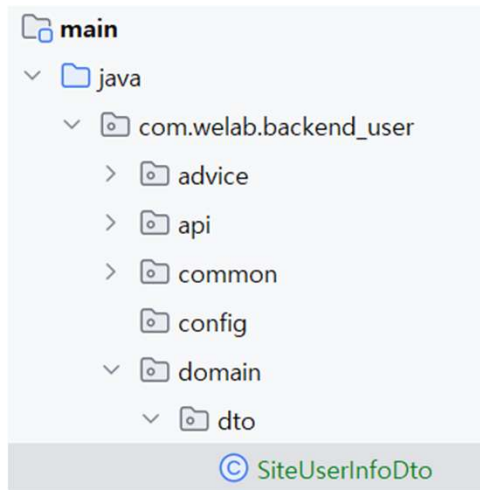


서비스 로직 작성

Post Service 알림톡 전송 요청

User 프로젝트 코드 작성 – SiteUserInfo 추가



```
@Getter
@Setter
public class SiteUserInfoDto {
    private String userId;

    private String phoneNumber;

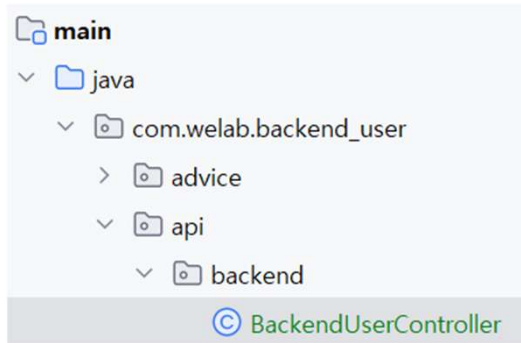
    public static SiteUserInfoDto fromEntity(SiteUser siteUser) {
        SiteUserInfoDto dto = new SiteUserInfoDto();

        dto.userId = siteUser.getUserId();
        dto.phoneNumber = siteUser.getPhoneNumber();

        return dto;
    }
}
```

Post에서 ALim 서비스를 요청하기 위해서는 UserId와 PhoneNumber 데이터가 필요
User 서비스에서 UserId와 PhoneNumber 데이터를 Post로 전송해야 함

User 프로젝트 코드 작성 – BackendUserController 추가



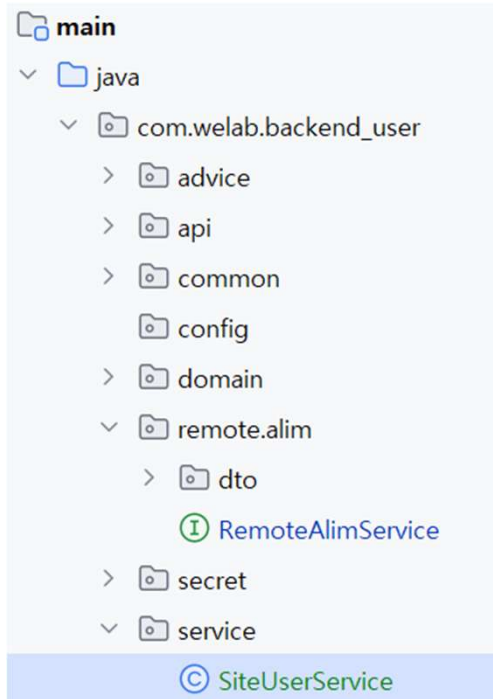
```
@Slf4j
@RestController
@RequestMapping(value = "/backend/user/v1", produces = MediaType.APPLICATION_JSON_VALUE)
@RequiredArgsConstructor
public class BackendUserController {
    private final SiteUserService siteUserService;

    @GetMapping(value = "/user/{userId}")
    public ApiResponseDto<SiteUserInfoDto> userInfo(@PathVariable String userId) {
        SiteUserInfoDto userInfoDto = siteUserService.userInfo(userId);

        return ApiResponseDto.createOk(userInfoDto);
    }
}
```

SiteUserInfoDto 요청에 대한 응답을 수행하는 컨트롤러

User 프로젝트 코드 작성 – SiteUserService 수정



```
@Slf4j
@Service
@RequiredArgsConstructor
public class SiteUserService {
    private final SiteUserRepository siteUserRepository;
    private final TokenGenerator tokenGenerator;

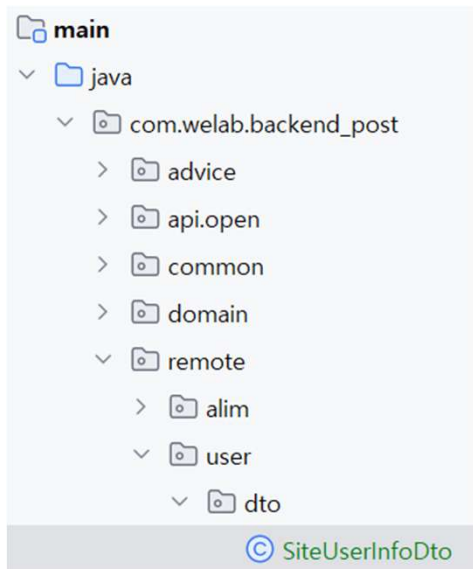
    ...

    @Transactional(readOnly = true)
    public SiteUserInfoDto userInfo(String userId) {
        SiteUser user = siteUserRepository.findById(userId);
        if (user == null) {
            throw new NotFound("사용자를 찾을 수 없습니다.");
        }

        return SiteUserInfoDto.fromEntity(user);
    }
}
```

추가

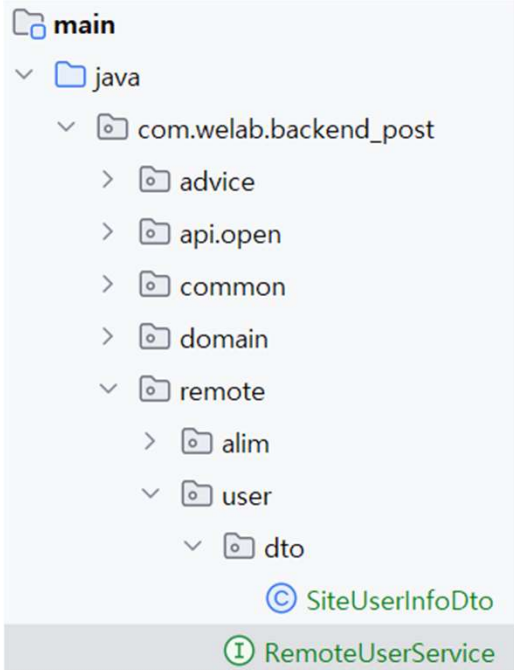
Post 프로젝트 코드 작성 – SiteUserInfoDto 추가



```
@Getter
@Setter
public class SiteUserInfoDto {
    private String userId;

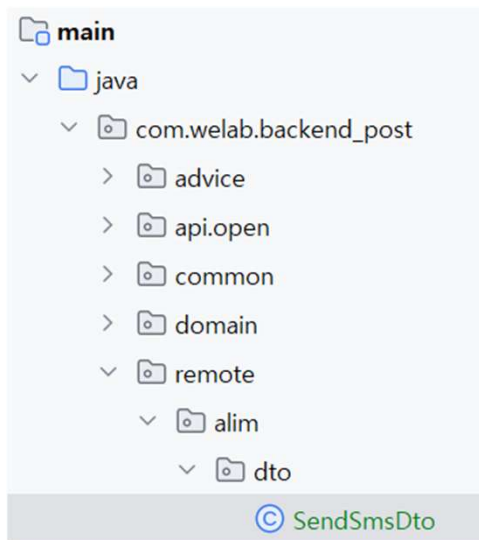
    private String phoneNumber;
}
```

Post 프로젝트 코드 작성 – RemoteUserService 추가



```
@FeignClient(name = "backend-user", path = "/backend/user/v1")  
public interface RemoteUserService {  
    @GetMapping(value = "/user/{userId}")  
    public ApiResponseDto<SiteUserInfoDto> userInfo(@PathVariable String userId);  
}
```

Post 프로젝트 코드 작성 – SendSmsDto 추가

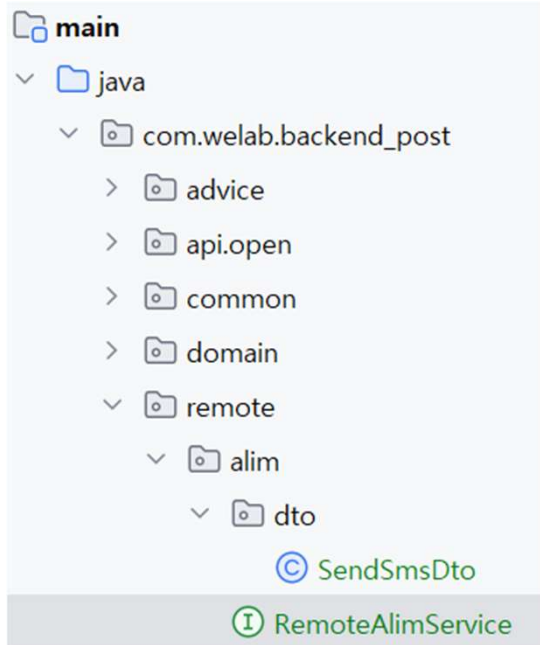


```
@NoArgsConstructor(access = AccessLevel.PRIVATE)
public class SendSmsDto {
    @Getter
    @Setter
    public static class Request {
        private String userId;
        private String phoneNumber;
        private String title;
        private String message;
    }

    @Getter @Setter
    public static class Response {
        private String result;
    }
}
```

Alim 서비스로의 요청/응답을 수행하기 위한 DTO

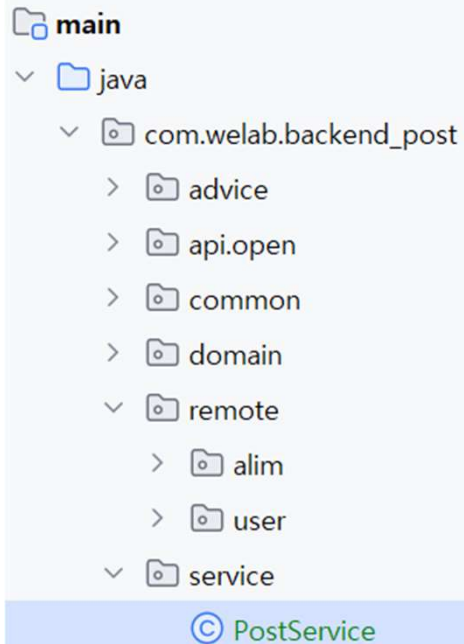
Post 프로젝트 코드 작성 – RemoteAlimService 추가



```
@FeignClient(name = "backend-alim", path = "/backend/alim/v1")
public interface RemoteAlimService {
    @GetMapping(value = "/sms")
    public String sms();

    @PostMapping(value = "/sms")
    public SendSmsDto.Response sendSms(@RequestBody SendSmsDto.Request request);
}
```


Post 프로젝트 코드 작성 – PostService 수정



```
@Slf4j
@Service
@RequiredArgsConstructor
public class PostService {
    private final PostRepository postRepository;
    private final PostCommentRepository postCommentRepository;
    private final RemoteUserService remoteUserService;
    private final RemoteAlimService remoteAlimService;

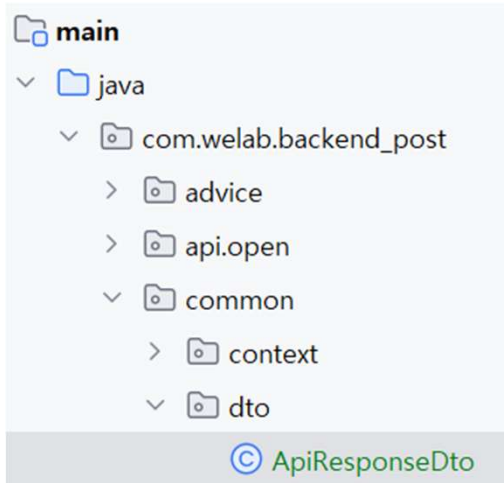
    @Transactional
    public void addPostComment(PostCommentCreateDto createDto) {
        ...

        // 알림톡 보내기 위해 사용자 정보 조회
        SiteUserInfoDto userInfoDto = remoteUserService.userInfo(post.getUserId()).getData();

        // 알림톡 전송 요청
        SendSmsDto.Request requestDto = new SendSmsDto.Request();
        requestDto.setUserId(userInfoDto.getUserId());
        requestDto.setPhoneNumber(userInfoDto.getPhoneNumber());
        requestDto.setTitle("댓글 달림");
        requestDto.setMessage("댓글이 달렸습니다.");

        remoteAlimService.sendSms(requestDto);
    }
}
```

Post 프로젝트 코드 작성 – ApiResponseDto 수정



```
@Getter
@NoArgsConstructor
public class ApiResponseDto<T> {
    private String code;
    private String message;
    private T data;

    private ApiResponseDto(String code, String message) {
        this.code = code;
        this.message = message;
    }
}
```

FeignClient에서 Remote API 응답으로 사용하기 위함