

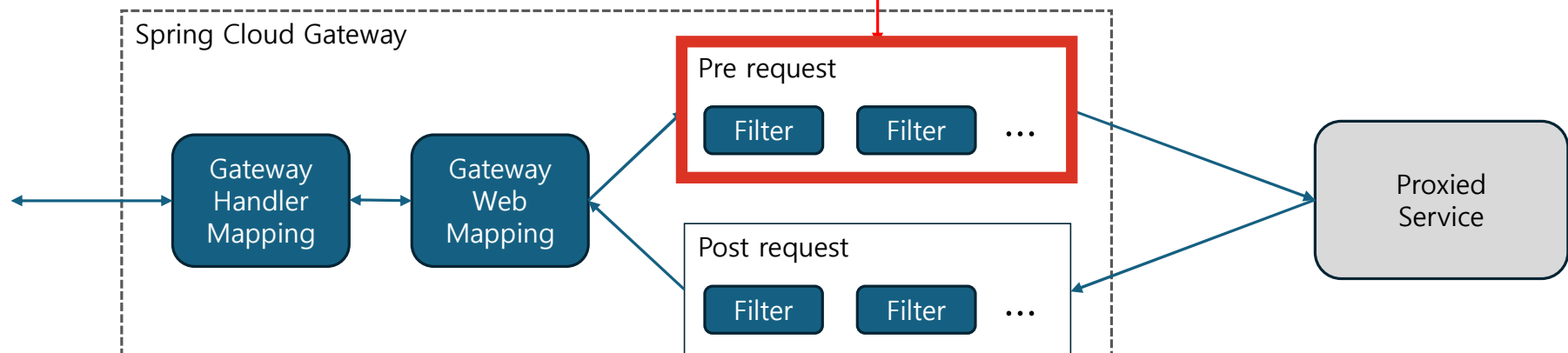
서비스 로직 작성

API Gateway 인증정보 전달하기

Spring Cloud Gateway Filter

인증정보를 여기에서 넣어보자.

- + 사용자 아이디
- + 클라이언트 디바이스 타입
- + 클라이언트 주소



```
public class HttpUtils {  
    public static String getRemoteAddr(HttpServletRequest request) {  
        String remoteAddr = request.getHeader("X-FORWARDED-FOR");  
        if (remoteAddr == null) {  
            remoteAddr = request.getRemoteAddr();  
        }  
  
        String[] pair = remoteAddr.split(":");  
        if (pair.length > 0) {  
            return pair[0];  
        } else {  
            return "";  
        }  
    }  
}
```

요청에서 remoteAddr 추출을 위한 Util 함수

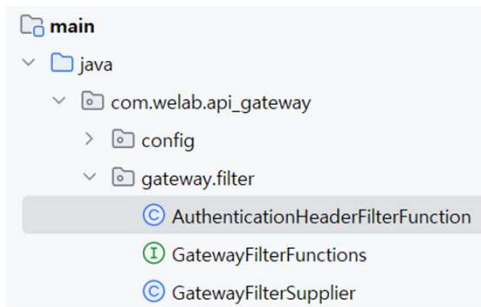
Function

+ ServerRequest
+ ServerRequest

Gateway 프로젝트 코드 작성 - AuthenticationHeaderFilterFunction

Authentication Principal

가



Address

가

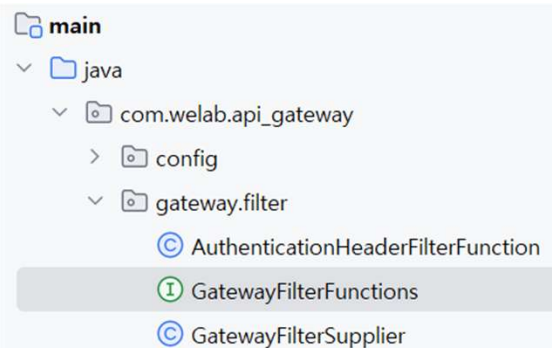
```
class AuthenticationHeaderFilterFunction {  
    public static Function<ServerRequest, ServerRequest> addHeader() {  
        return request -> {  
            ServerRequest.Builder requestBuilder = ServerRequest.from(request);  
  
            Object principal = SecurityContextHolder.getContext().getAuthentication().getPrincipal();  
            if (principal instanceof UserPrincipal userPrincipal) {  
                requestBuilder.header("X-Auth-UserId", userPrincipal.getUserId());  
  
                // 필요시 권한 정보 입력  
                // requestBuilder.header("X-Auth-Authorities", ...);  
            }  
  
            // String remoteAddr = HttpUtils.getRemoteAddr(requestBuilder.servletRequest());  
            String remoteAddr = "70.1.23.15";  
            requestBuilder.header("X-Client-Address", remoteAddr);  
  
            String device = "WEB";  
            requestBuilder.header("X-Client-Device", device);  
  
            return requestBuilder.build();  
        };  
    }  
}
```

ServerRequest에 값을 추가하는 메소드
[개발자 임의 지정]
+ UserId
+ Client-Address
+ Client-Device

가

Gateway 프로젝트 코드 작성 – GatewayFilterFunctions

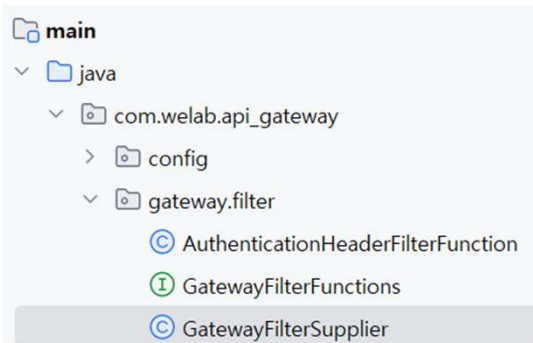
Spring Cloud Gateway
Routing Filter



```
public interface GatewayFilterFunctions {  
    @Shortcut  
    static HandlerFilterFunction<ServerResponse, ServerResponse> addAuthenticationHeader() {  
        return ofRequestProcessor(AuthenticationHeaderFilterFunction.addHeader());  
    }  
}
```

라우팅 시, addHeader() 함수를 통해 헤더에 값 추가
(추가되는 값 = 사용자 아이디, 클라이언트 디바이스 타입, 클라이언트 주소)

Gateway 프로젝트 코드 작성 – GatewayFilterSupplier



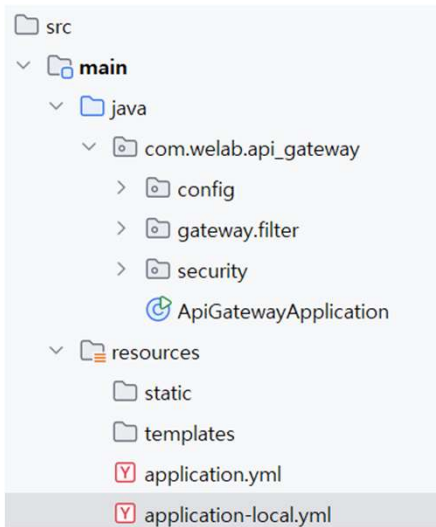
```
@Configuration
public class GatewayFilterSupplier extends SimpleFilterSupplier {
    public GatewayFilterSupplier() {
        super(GatewayFilterFunctions.class);
    }
}
```

GatewayFilterFunctions 클래스 내의
addAuthenticationHeader() 메소드를 필터링 행동으로 등록.

라우팅 시, AuthenticationHeaderFilterFunction 클래스의 addHeader() 메소드 적용

Gateway 프로젝트 설정

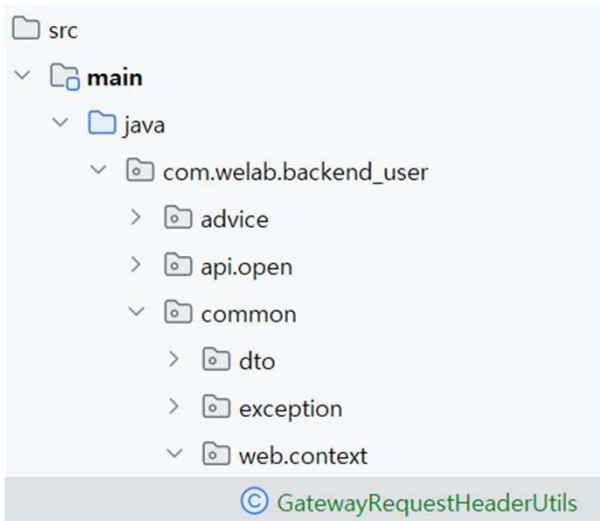
Gateway 필터 지정



```
spring:
  cloud:
    gateway:
      mvc:
        routes: # 라우팅 설정
          - id: backend-user
            predicates: # 라우팅 조건
              - Path=/api/user/**
            uri: lb://backend-user
            filters:
              - AddAuthenticationHeader
          - id: backend-alim
            predicates: # 라우팅 조건
              - Path=/api/alim/**
            uri: lb://backend-alim
            filters:
              - AddAuthenticationHeader
```

@Shortcut으로 명시해야 함

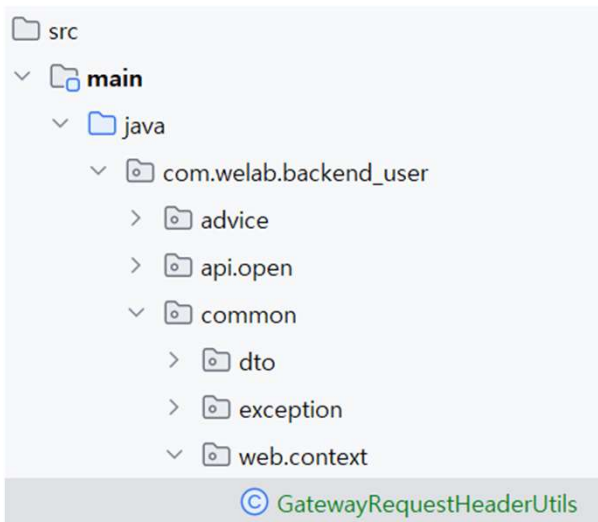
User 프로젝트 코드 작성 – GatewayRequestHeaderUtils



```
public class GatewayRequestHeaderUtils {  
    public static String getRequestHeaderParamAsString(String key) {  
        ServletRequestAttributes requestAttributes =  
            (ServletRequestAttributes) RequestContextHolder.currentRequestAttributes();  
        return requestAttributes.getRequest().getHeader(key);  
    }  
}
```

헤더에서 key에 해당하는 value 추출

User 프로젝트 코드 작성 – GatewayRequestHeaderUtils (계속)



```
public static String getUserId() {
    String userId = getRequestParamAsString("X-Auth-UserId");
    if (userId == null) {
        return null;
    }

    return userId;
}

public static String getClientDevice() {
    String clientDevice = getRequestParamAsString("X-Client-Device");
    if (clientDevice == null) {
        return null;
    }

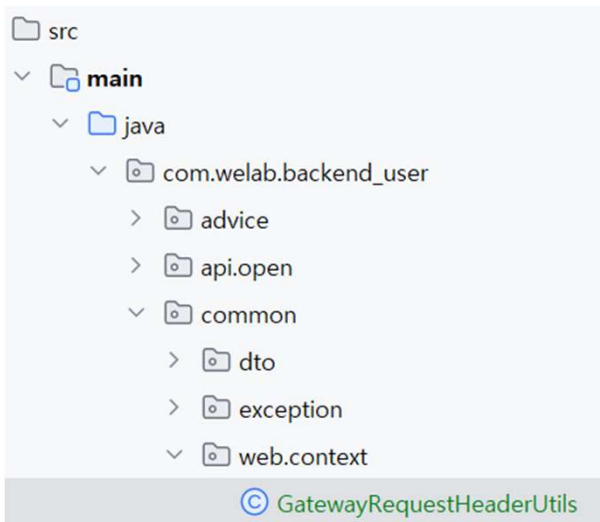
    return clientDevice;
}

public static String getClientAddress() {
    String clientAddress = getRequestParamAsString("X-Client-Address");
    if (clientAddress == null) {
        return null;
    }

    return clientAddress;
}
```

헤더에서 각 Key 맞는 Value 추출
(UserId, Client Device, Client Address)

User 프로젝트 코드 작성 – GatewayRequestHeaderUtils (계속)



```
public static String getUserIdOrThrowException() {
    String userId = getUserId();
    if (userId == null) {
        throw new NotFound("헤더에 userId 정보가 없습니다.");
    }

    return userId;
}

public static String getClientDeviceOrThrowException() {
    String clientDevice = getClientDevice();
    if (clientDevice == null) {
        throw new NotFound("헤더에 사용자 디바이스 정보가 없습니다.");
    }

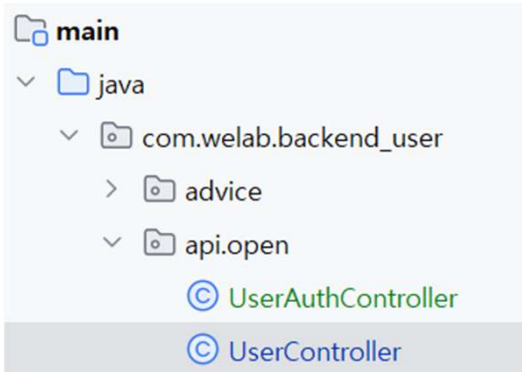
    return clientDevice;
}

public static String getClientAddressOrThrowException() {
    String clientAddress = getClientAddress();
    if (clientAddress == null) {
        throw new NotFound("헤더에 사용자 IP 주소 정보가 없습니다.");
    }

    return clientAddress;
}
}
```

헤더에서 각 Key 맞는 Value 추출. null 시, 에러 처리
(UserId, Client Device, Client Address)

User 프로젝트 코드 작성 – GatewayRequestHeaderUtils 동작 확인



```
@Slf4j
@RestController
@RequestMapping(value = "/api/user/v1", produces = MediaType.APPLICATION_JSON_VALUE)
@RequiredArgsConstructor
public class UserController {
    private final RemoteAlimService remoteAlimService;

    @GetMapping(value = "/test")
    public ApiResponseDto<String> test() {
        // String response = remoteAlimService.sms();
        String userId = GatewayRequestHeaderUtils.getUserIdOrThrowException();
        log.info("userId = {}", userId);
        return ApiResponseDto.createOk(userId);
    }
    ...
}
```

API Gateway 가
"X - Auth - UserId" Value