

Ans :1

```
private Optional<Integer> getId(int type, String appId, String openId) {
    Integer id = null;
    switch (type) {
        case 1:
            Type1User type1User = userRepository.getType1User(openId);
            validateUserNotNull(type1User);
            id = type1User.getId();
            break;
        case 2:
            Type2User type2User = userRepository.getType2User(openId);
            validateUserNotNull(type2User);
            id = type2User.getId();
            break;
        case 4:
            Type4User type4User = userRepository.getType4UserByOpenId(openId);
            validateUserNotNull(type4User);
            id = type4User.getId();
            break;
        case 5:
            Type5User type5User = userRepository.getType5UserByOpenId(openId);
            validateUserNotNull(type5User);
            id = type5User.getId();
            break;
        case 6:
            Type6User type6User = userRepository.getType6UserByOpenId(openId);
            validateUserNotNull(type6User);
            id = type6User.getId();
            break;
        case 8:
            Type8User type8User = userRepository.getType8User(Long.valueOf(openId));
            validateUserNotNull(type8User);
            id = type8User.getId();
            break;
        case 11:
            Type11User type11User = userRepository.getType11UserByTokenAndUID(appId,
openId);
            validateUserNotNull(type11User);
            id = type11User.getId();
            break;
        case 12:
            Type12User type12User = userRepository.getType12UserByAppIdAndOpenId(appId,
openId);
            validateUserNotNull(type12User);
            id = type12User.getId();
            break;
        case 13:
            Type13User type13User = userRepository.getType13UserByAppIdAndOpenId(appId,
openId);
            validateUserNotNull(type13User);
            id = type13User.getId();
    }
```

```
default:
break;
}
return Optional.ofNullable(id);
}
private void validateUserNotNull(Object user) {
if (user == null) {
throw new UserNotFoundException();
}
}
}
```