

# 프로젝트 문서

## Sabang - DB 정의서

2019-04-16

..

팀 명 : 1 조 (Sabnag)

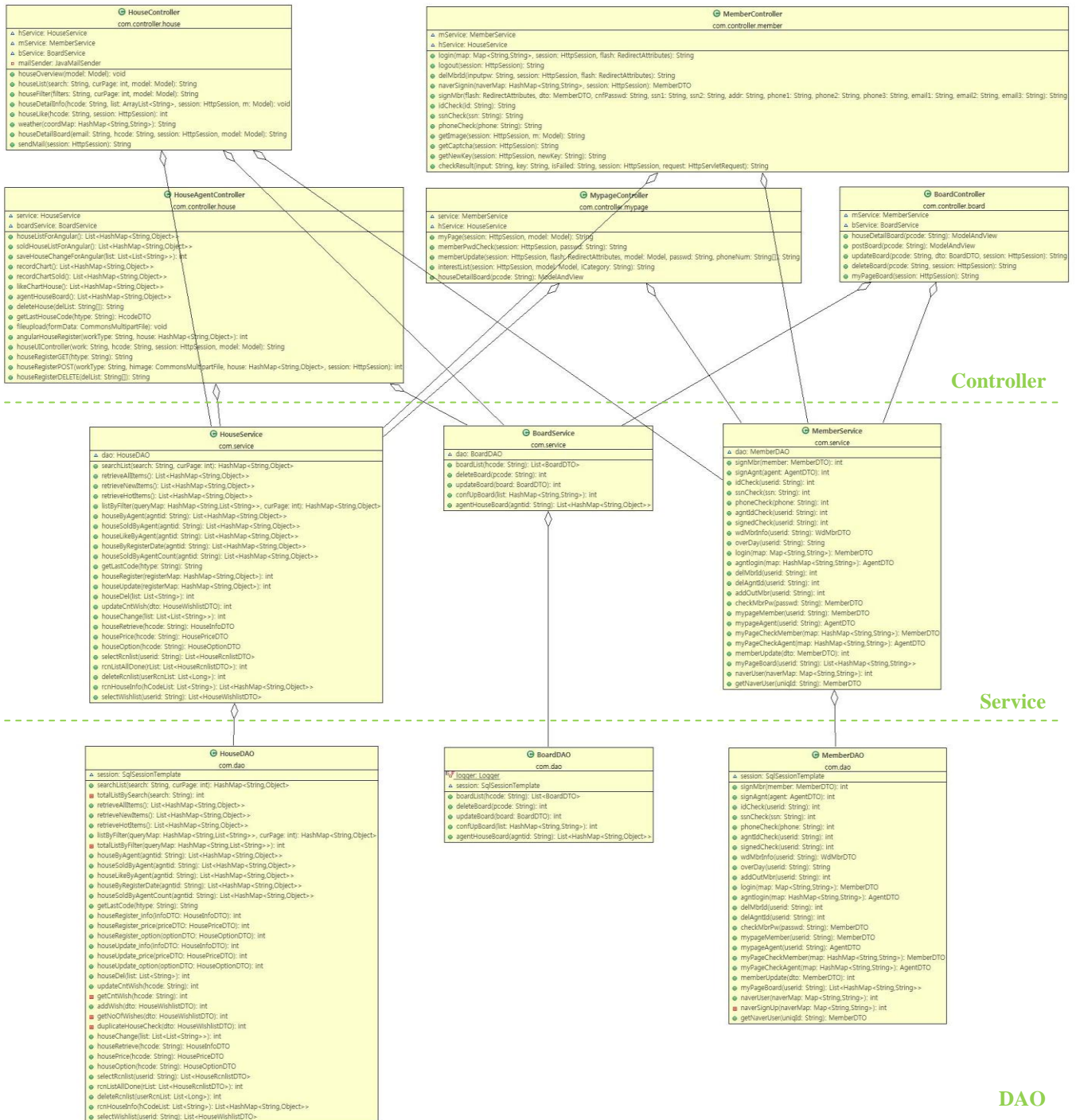
민희정 박태서 함효정

# 목차

1. 클래스 다이어그램 Overview .....	3
i. Controller/Service/DAO .....	3
ii. DTO & Other Class .....	4
2. 클래스 다이어그램 Detail .....	5
i. Controller .....	5
① MemberController .....	5
② MyPageController .....	6
③ BoardController .....	7
④ HouseController .....	8
⑤ HouseAgentController .....	9
ii. Service .....	11
① MemberService .....	11
② HouseService .....	13
③ BoardService .....	15
iii. DAO .....	16
① MemberDAO .....	16
② HouseDAO .....	18
③ BoardDAO .....	21
iv. Other Class .....	22
④ LoginCheckInterCeptor .....	22

## 1. 클래스 다이어그램 Overview

## i. Controller/Service/DAO



## ii. DTO & Other Class



## 2. 클래스 다이어그램 Detail


### i. Controller

#### ① MemberController

com.controller.member.MemberController	
<ul style="list-style-type: none"> <li>▲ mService: MemberService</li> <li>▲ hService: HouseService</li> </ul>	
<ul style="list-style-type: none"> <li>● login(map: Map&lt;String,String&gt;, session: HttpSession, flash: RedirectAttributes): String</li> <li>● logout(session: HttpSession): String</li> <li>● delMbrId(inputpw: String, session: HttpSession, flash: RedirectAttributes): String</li> <li>● naverSignin(naverMap: HashMap&lt;String,String&gt;, session: HttpSession): MemberDTO</li> <li>● signMbr(flash: RedirectAttributes, dto: MemberDTO, cnfPasswd: String, ssn1: String, ssn2: String, addr: String, phone1: String, phone2: String, phone3: String, email1: String, email2: String, email3: String): String</li> <li>● idCheck(id: String): String</li> <li>● ssnCheck(ssn: String): String</li> <li>● phoneCheck(phone: String): String</li> <li>● getImage(session: HttpSession, m: Model): String</li> <li>● getCapcha(session: HttpSession): String</li> <li>● getNewKey(session: HttpSession, newKey: String): String</li> <li>● checkResult(input: String, key: String, isFailed: String, session: HttpSession, request: HttpServletRequest): String</li> </ul>	









Method Summary	
public String	<b>login</b> (RedirectAttributes falsh, HttpSession session, Map<String, String> map) // 로그인 메소드
public String	<b>logout</b> (HttpSession session) // 로그아웃 메소드
public String	<b>delMbrId</b> (HttpSession session, RedirectAttributes falsh, String inputpw) // 회원탈퇴 메소드
Public MemberDTO	<b>naverSignin</b> (HashMap<String, String > naverMap, HttpSession session) // 네이버 SNS 로그인
private String	<b>signMbr</b> (RedirectAttributes falsh, MemberDTO dto, String cnfPasswd, String ssn1, String ssn2, String addr, String phone1, String phone2, String phone3, String email, String email2, String email3) // 입력한 정보로 회원 가입하는 메소드
public String	<b>idCheck</b> (String id) // 아이디 중복 조회 메소드
public String	<b>ssnCheck</b> (String ssn) // 주민등록번호 중복 조회 메소드
public String	<b>phoneCheck</b> (String phone) // 핸드폰번호 중복 조회 메소드
public String	<b>getImage</b> (HttpSession session, Model m) // 네이버 캡차 api key 로 이미지를 가져오는 메소드
public String	<b>getCapcha</b> (HttpSession session) // 네이버 캡차 api key 를 발급 받는 메소드
public String	<b>getNewKey</b> (HttpSession session, String newKey) // ajax 로 새로운 key 를 발급 받는 메소드
public String	<b>checkResult</b> (String input, String key) // 캡차 입력 값을 비교하여 논리값을 반환하는 메소드

## ② MyPageController

 com.controller.mypage.MypageController
▲ service: MemberService ▲ hService: HouseService
● myPage(session: HttpSession, model: Model): String ● memberPwdCheck(session: HttpSession, passwd: String): String ● memberUpdate(session: HttpSession, flash: RedirectAttributes, model: Model, passwd: String, phoneNum: String[]): String ● interestList(session: HttpSession, model: Model, iCategory: String): String ● houseDetailBoard(pcode: String): ModelAndView

Method Summary	
public String	<b>myPage</b> (HttpSession session, Model model) // 로그인 정보를 통해 회원정보를 조회하는 메소드
public String	<b>memberPwdCheck</b> (HttpSession session, String passwd) // 입력한 비밀번호와 회원정보의 비밀번호를 비교하는 메소드
public String	<b>memberUpdate</b> (HttpSession session, String[] phoneNum, String passwd, RedirectAttribute flash) // 회원정보 수정 메소드
public String	<b>interestList</b> (HttpSession session, String iCategory, Model model) // 로그인 정보를 통해 최근 본 방과 찜한 방을 조회하는 메소드
public ModelAndView	<b>houseDetailBoard</b> (String pcode) // 매물 상세 정보를 조회하는 메소드

## ③ BoardController

 com.controller.board.BoardController
 mService: MemberService  bService: BoardService
 houseDetailBoard(pcode: String): ModelAndView  postBoard(pcode: String): ModelAndView  updateBoard(pcode: String, dto: BoardDTO, session: HttpSession): String  deleteBoard(pcode: String, session: HttpSession): String  myPageBoard(session: HttpSession): String

## Method Summary

public ModelAndView	<b>houseDetailBoard</b> (String pcode) // 매물 상세페이지에서 문의글을 조회하는 메소드
public ModelAndView	<b>postBoard</b> (String pcode) // 문의글 작성 메소드
public String	<b>updateBoard</b> (BoardDTO dto, HttpSession session, String pcode) // 문의글 수정 메소드
public String	<b>deleteBoard</b> (HttpSession session, String pcode) // 문의글 삭제 메소드
public String	<b>myPageBoard</b> (HttpSession session) // 회원의 문의사항을 조회하는 메소드

## ④ HouseController

com.controller.house.HouseController	
△ hService: HouseService	
△ mService: MemberService	
△ bService: BoardService	
▣ mailSender: JavaMailSender	
● houseOverview(model: Model): void	
● houseList(search: String, curPage: int, model: Model): String	
● houseFilter(filters: String, curPage: int, model: Model): String	
● houseDetailInfo(hcode: String, list: ArrayList<String>, session: HttpSession, m: Model): void	
● houseLike(hcode: String, session: HttpSession): int	
● weather(coordMap: HashMap<String,String>): String	
● houseDetailBoard(email: String, hcode: String, session: HttpSession, model: Model): String	
● sendMail(session: HttpSession): String	

## Method Summary

public void	<b>houseOverview</b> (Model model) // 신촌 페이지로 처음 들어올 때 화면에 신매물/핫매물 리스트 보여줌
public String	<b>houseList</b> (String search, int curPage, Model model) // 매물 검색 리스트
public String	<b>houseFilter</b> (String filters, int curPage, Model model) // 매물 필터 리스트
Public void	<b>houseDetailInfo</b> (HttpSession session, ArrayList<String> list, String hcode, Model m) // 매물 상세 정보를 조회하는 메소드
private int	<b>houseLike</b> (String hcode, HttpSession session) // 매물 상세 페이지에서 매물 찜하기 버튼 눌렀을 때 wishlist, house_info 에 cntwish 컬럼 업데이트
public String	<b>weather</b> (HashMap<String, String> coordMap) // 매물 상세 페이지에서 날씨 정보 출력
public String	<b>houseDetailBoard</b> (String hcode, String email, HttpSession session, Model model) // 매물 상세 페이지에서 게시판 조회하는 메소드
public String	<b>sendMail</b> (HttpSession session) // 중개인에게 메일을 전송하는 메소드



## ⑤ HouseAgentController

com.controller.house.HouseAgentController	
▲ service: HouseService	
▲ boardService: BoardService	
● houseListForAngular(): List<HashMap<String, Object>>	
● soldHouseListForAngular(): List<HashMap<String, Object>>	
● saveHouseChangeForAngular(list: List<List<String>>): int	
● recordChart(): List<HashMap<String, Object>>	
● recordChartSold(): List<HashMap<String, Object>>	
● likeChartHouse(): List<HashMap<String, Object>>	
● agentHouseBoard(): List<HashMap<String, Object>>	
● deleteHouse(delList: String[]): String	
● getLastHouseCode(hType: String): HcodeDTO	
● fileupload(himage: CommonsMultipartFile): void	
● angularHouseRegister(workType: String, house: HashMap<String, Object>): int	
● houseUIController(work: String, hcode: String, session: HttpSession, model: Model): String	
● houseRegisterGET(hType: String): String	
● houseRegisterPOST(workType: String, himage: CommonsMultipartFile, house: HashMap<String, Object>, session: HttpSession): int	
● houseRegisterDELETE(delList: String[]): String	

## Method Summary

public List<HashMap<String, Object>>	<b>houseListForAngular</b> () // Angular 로 구현한 화면에서 중개인 페이지로 들어왔을 때 매물 정보 반환
public List<HashMap<String, Object>>	<b>soldHouseListForAngular</b> () // Angular 로 구현한 화면에서 중개인 페이지로 들어왔을 때 중개완료 매물 반환
public int	<b>saveHouseChangeForAngular</b> (List<List<String>> list) // Angular 로 구현한 화면에서 중개인 페이지로 들어왔을 때 매물중/매물중개완료 리스트 변환 후 저장하면 변환
Public List<HashMap<String, Object>>	<b>recordChart</b> () // Angular 로 구현한 화면에서 중개인 페이지로 들어왔을 때 중개중매물 일자별/중개인 아이디별로 가져와서 그래프의 데이터로 사용
private List<HashMap<String, Object>>	<b>recordChartSold</b> () // Angular 로 구현한 화면에서 중개인 페이지로 들어왔을 때 중개완료 매물을 일자별/중개인 아이디별로 가져와서 그래프의 데이터로 사용
public List<HashMap<String, Object>>	<b>likeChartHouse</b> () // Angular 로 구현한 화면에서 중개인 페이지로 들어왔을 때 매물별/중개인별 좋아요가 많은 순으로 매물 반환
public List<HashMap<String, Object>>	<b>agentHouseBoard</b> () // Angular 로 구현한 화면에서 중개인 페이지로 들어왔을 때 중개인이 올린 매물에 대한 문의사항 게시판을 보여줌
public String	<b>deleteHouse</b> (String[] delList) // 매물 삭제
public HcodeDTO	<b>getLastHouseCode</b> (String hType) // Angular 로 구현한 화면에서 중개인 페이지로 들어왔을 때 매물 등록 시 매물의 마지막 코드 반환
public void	<b>fileupload</b> (CommonsMultipartFile himage) // Angular 로 구현한 화면에서 중개인 페이지로 들어왔을 때 매물 등록에 파일업로드 기능(구현중)

## Method Summary

public int	<b>angularHouseRegister</b> (HashMap<String, Object> house, String workType) // Angular 로 구현한 화면에서 중개인 페이지로 들어왔을 때 매물등록
public String	<b>houseUIController</b> (String work, String hcode, HttpSession session, Model model) // 로그인한 에이전트의 중개중/중개완료 매물 리스트와 날짜별 수치, 좋아요 많은 순 매물 리스트 반환
public String	<b>houseRegisterGET</b> (String htype) // 요청된 매물 타입의 마지막 매물 코드 반환
public int	<b>houseRegisterPOST</b> (CommonsMultipartFile himage, String workType, HashMap<String, Object> house, HttpSession session) // 매물등록
Public String	<b>houseRegisterDELETE</b> (String[] delList) // 매물삭제

## ii. Service

### ① MemberService

com.service.MemberService
▲ dao: MemberDAO ● signMbr(member: MemberDTO): int ● signAgnt(agent: AgentDTO): int ● idCheck(userid: String): int ● ssnCheck(ssn: String): int ● phoneCheck(phone: String): int ● agntIdCheck(userid: String): int ● signedCheck(userid: String): int ● wdMbrInfo(userid: String): WdMbrDTO ● overDay(userid: String): String ● login(map: Map<String,String>): MemberDTO ● agntlogin(map: HashMap<String,String>): AgentDTO ● delMbrId(userid: String): int ● delAgntId(userid: String): int ● addOutMbr(userid: String): int ● checkMbrPw(passwd: String): MemberDTO ● mypageMember(userid: String): MemberDTO ● mypageAgent(userid: String): AgentDTO ● myPageCheckMember(map: HashMap<String,String>): MemberDTO ● myPageCheckAgent(map: HashMap<String,String>): AgentDTO ● memberUpdate(dto: MemberDTO): int ● myPageBoard(userid: String): List<HashMap<String,String>> ● naverUser(naverMap: Map<String,String>): int ● getNaverUser(uniqId: String): MemberDTO

## Method Summary

public int	<b>signMbr</b> (MemberDTO member) // 입력한 정보로 회원 가입하는 메소드
public int	<b>idCheck</b> (String userid) // 아이디 중복 조회 메소드
public int	<b>ssnCheck</b> (String ssn) // 주민등록번호 중복 조회 메소드
Public int	<b>phoneCheck</b> (String phone) // 전화번호 중복 조회 메소드
private int	<b>signedCheck</b> (String userid) // 회원 가입할 때 입력한 아이디를 통해 가입 이력을 조회하는 메소드
public WdMbrDTO	<b>wdMbrInfo</b> (String userid) // 탈퇴회원 정보 조회 메소드
public String	<b>overDay</b> (String userid) // 회원탈퇴 시 24시간이 지난 시간을 반환하는 메소드
public MemberDTO	<b>login</b> (Map<String, String> map) // 로그인 메소드

Method Summary	
public int	<b>delMbrId</b> (String userid) // 회원탈퇴 메소드
public int	<b>addOutMbr</b> (String userid) // 탈퇴회원 정보를 저장하는 메소드
public MemberDTO	<b>checkMbrPw</b> (String passwd) // 회원탈퇴 시 회원의 비밀번호를 확인하는 메소드
public MemberDTO	<b>myPageMember</b> (String userid) // 로그인 정보를 통해 회원정보를 조회하는 메소드
public MemberDTO	<b>myPageCheckMember</b> (HashMap<String, String> map) // 마이페이지에서 입력한 정보를 통해 회원정보를 조회하는 메소드
public int	<b>memberUpdate</b> (Member dto) // 회원정보 수정 메소드
Public List<HashMap<String,String>>	<b>myPageBoard</b> (String userid) // 회원의 문의사항을 조회하는 메소드
public int	<b>naverUser</b> (Map<String, String> naverMap) // NAVER SNS 로그인 시 해당 유저가 회원가입이 되어 있지 않으면 회원가입시킴
public MemberDTO	<b>getNaverUser</b> (String uniqId) // NAVER SNS 로그인 유저의 회원정보 반환

## ② HouseService

com.service.HouseService	
▲ dao: HouseDAO	
●	searchList(search: String, curPage: int): HashMap<String, Object>
●	retrieveAllItems(): List<HashMap<String, Object>>
●	retrieveNewItems(): List<HashMap<String, Object>>
●	retrieveHotItems(): List<HashMap<String, Object>>
●	listByFilter(queryMap: HashMap<String, List<String>>, curPage: int): HashMap<String, Object>
●	houseByAgent(agntid: String): List<HashMap<String, Object>>
●	houseSoldByAgent(agntid: String): List<HashMap<String, Object>>
●	houseLikeByAgent(agntid: String): List<HashMap<String, Object>>
●	houseByRegisterDate(agntid: String): List<HashMap<String, Object>>
●	houseSoldByAgentCount(agntid: String): List<HashMap<String, Object>>
●	getLastCode(hotype: String): String
●	houseRegister(registerMap: HashMap<String, Object>): int
●	houseUpdate(registerMap: HashMap<String, Object>): int
●	houseDel(list: List<String>): int
●	updateCntWish(dto: HouseWishlistDTO): int
●	houseChange(list: List<List<String>>): int
●	houseRetrieve(hcode: String): HouseInfoDTO
●	housePrice(hcode: String): HousePriceDTO
●	houseOption(hcode: String): HouseOptionDTO
●	selectRcnlist(userid: String): List<HouseRcnlistDTO>
●	rcnListAllDone(rList: List<HouseRcnlistDTO>): int
●	deleteRcnlist(userRcnList: List<Long>): int
●	rcnHouseInfo(hCodeList: List<String>): List<HashMap<String, Object>>
●	selectWishlist(userid: String): List<HouseWishlistDTO>

## Method Summary

public HashMap<String, Object>	<b>searchList</b> (String search, int curPage) // 매물 검색 리스트 반환
public List<HashMap<String, Object>>	<b>retrieveAllItems</b> () // 전체 매물 반환
Public List<HashMap<String, Object>>	<b>retrieveNewItems</b> () // 신매물 반환
private List<HashMap<String, Object>>	<b>retrieveHotItems</b> () // 인기매물 반환
public HashMap<String, Object>	<b>listByFilter</b> (HashMap<String, List<String>> queryMap, int cutPage) // 매물 필터 리스트
public HashMap<String, Object>	<b>houseByAgent</b> (String agntid) // 로그인한 중개인이 등록한 매물 리스트
public List<HashMap<String, Object>>	<b>houseSoldByAgent</b> (String agntid) // 로그인한 중개인이 등록한 매물 중 중개완료된 리스트

## Method Summary

public List<HashMap<String, Object>>	<b>houseLikeByAgent</b> (String agntid) // 로그인한 중개인이 등록한 매물 중 좋아요 많은 순
public List<HashMap<String, Object>>	<b>houseByRegisterDate</b> (String agntid) // 로그인한 중개인이 등록한 매물의 등록일자별 리스트
public List<HashMap<String, Object>>	<b>houseSoldByAgentCount</b> (String agntid) // 로그인한 중개인이 등록한 매물 중 중개완료된 매물의 수
public String	<b>getLastCode</b> (String htype) // 매물등록시 요청된 매물 타입의 마지막 코드 반환
public int	<b>houseRegister</b> (HashMap<String, Object> registerMap) // 매물 등록
public int	<b>houseUpdate</b> (HashMap<String, Object> registerMap) // 매물 수정
public int	<b>houseDel</b> (List<String> list) // 매물 삭제
public int	<b>updateCntWish</b> (String hcode) // 화면에서 좋아요 버튼 눌렀을 시 해당 매물의 좋아요 수치 증가
public int	<b>houseChange</b> (List<List<String>> list) // 매물 중개중/중개완료 여부 저장
public HouseInfoDTO	<b>houseRetrive</b> (String hcode) // 매물 코드를 통해 매물의 위치, 매물명 등의 정보를 조회하는 메소드
public HousePriceDTO	<b>housePrice</b> (String hcode) // 매물 코드를 통해 매물의 가격 등의 정보를 조회하는 메소드
public HouseOptionDTO	<b>houseOption</b> (String hcode) // 매물 코드를 통해 매물의 옵션을 조회하는 메소드
public List<HouseRcnlistDTO>	<b>selectRcnlist</b> (String userid) // 로그인 정보를 통해 최근 본 방을 조회하는 메소드
public int	<b>rcnListAllDone</b> (List<HouseRcnlistDTO> rList) // 최근 본 방을 저장하는 메소드
public int	<b>deleteRcnlist</b> (List<Long> userRcnList) // 최근 본 방을 삭제하는 메소드
public List<HashMap<String, Object>>	<b>rcnHouseInfo</b> (List<String> hCodeList) // 회원의 최근 본 방과 찜한 방의 매물 코드를 통해 매물 정보를 조회하는 메소드
public List<HouseWishlistDTO>	<b>selectWishlist</b> (String userid) // 로그인 정보를 통해 찜한 방을 조회하는 메소드

### ③ BoardService

com.service.BoardService	
▲ dao: BoardDAO	
● boardList(hcode: String): List<BoardDTO>	
● deleteBoard(pcode: String): int	
● updateBoard(board: BoardDTO): int	
● confUpBoard(list: HashMap<String,String>): int	
● agentHouseBoard(agentid: String): List<HashMap<String,Object>>	

Method Summary	
public List<BoardDTO>	<b>boardList</b> (String hcode) // 매물 코드를 통해 문의글을 조회하는 메소드
public int	<b>deleteBoard</b> (String pcode) // 문의글을 삭제하는 메소드
public int	<b>updateBoard</b> (BoardDTO board) // 문의글을 수정하는 메소드
public List<HashMap<String, Object>>	<b>agentHouseBoard</b> (String agentid) // Angular 로 구현한 화면에서 중개인 페이지에 들어왔을 때 로그인한 중개인이 올린 매물과 관련되 문의사항 반환

## iii. DAO

## ① MemberDAO

com.dao.MemberDAO	
△ session: SqlSessionTemplate	
● signMbr(member: MemberDTO): int	
● signAgnt(agent: AgentDTO): int	
● idCheck(userid: String): int	
● ssnCheck(ssn: String): int	
● phoneCheck(phone: String): int	
● agntIdCheck(userid: String): int	
● signedCheck(userid: String): int	
● wdMbrInfo(userid: String): WdMbrDTO	
● overDay(userid: String): String	
● addOutMbr(userid: String): int	
● login(map: Map<String,String>): MemberDTO	
● agntlogin(map: HashMap<String,String>): AgentDTO	
● delMbrId(userid: String): int	
● delAgntId(userid: String): int	
● checkMbrPw(passwd: String): MemberDTO	
● mypageMember(userid: String): MemberDTO	
● mypageAgent(userid: String): AgentDTO	
● myPageCheckMember(map: HashMap<String,String>): MemberDTO	
● myPageCheckAgent(map: HashMap<String,String>): AgentDTO	
● memberUpdate(dto: MemberDTO): int	
● myPageBoard(userid: String): List<HashMap<String,String>>	
● naverUser(naverMap: Map<String,String>): int	
■ naverSignUp(naverMap: Map<String,String>): int	
● getNaverUser(uniqId: String): MemberDTO	

## Method Summary

public int	<b>signMbr</b> (MemberDTO member) // 입력한 정보로 회원 가입하는 메소드
public int	<b>idCheck</b> (String userid) // 아이디 중복 조회 메소드
public int	<b>ssnCheck</b> (String ssn) // 주민등록번호 중복 조회 메소드
Public int	<b>phoneCheck</b> (String phone) // 전화번호 중복 조회 메소드
private int	<b>signedCheck</b> (String userid) // 가입할 때 입력한 아이디를 통해 가입 이력을 조회하는 메소드
public WdMbrDTO	<b>wdMbrInfo</b> (String userid) // 탈퇴회원 정보 조회 메소드



## Method Summary

public String	<b>overDay</b> (String userid) // 회원탈퇴 시 24 시간이 지난 시간을 반환하는 메소드
public int	<b>addOutMbr</b> (String userid) // 탈퇴회원 정보를 저장하는 메소드
public MemberDTO	<b>login</b> (Map<String, String> map) // 로그인 메소드
public int	<b>delMbrId</b> (String userid) // 회원탈퇴 메소드
public MemberDTO	<b>checkMbrPw</b> (String passwd) // 회원탈퇴 시 회원의 비밀번호를 확인하는 메소드
public MemberDTO	<b>myPageMember</b> (String userid) // 로그인 정보를 통해 회원정보를 조회하는 메소드
public MemberDTO	<b>myPageCheckMember</b> (HashMap<String, String> map) // 마이페이지에서 입력한 정보를 통해 회원정보를 조회하는 메소드
public int	<b>memberUpdate</b> (Member dto) // 회원정보 수정 메소드
Public List<HashMap<String,String>>	<b>myPageBoard</b> (String userid) // 회원의 문의사항을 조회하는 메소드
public int	<b>naverUser</b> (Map<String, String> naverMap) // NAVER SNS 로그인 시 해당 유저가 회원가입이 되어 있는 지 여부 확인
private int	<b>naverSignUp</b> (Map<String, String> naverMap) // NAVER SNS 로그인 한 유저의 회원가입
public MemberDTO	<b>getNaverUser</b> (String uniqId) // NAVER SNS 로그인 유저의 회원정보 반환

## ② HouseDAO

com.dao.HouseDAO	
▲ session: SqlSessionTemplate	
● searchList(search: String, curPage: int): HashMap<String, Object>	
■ totalListBySearch(search: String): int	
● retrieveAllItems(): List<HashMap<String, Object>>	
● retrieveNewItem(): List<HashMap<String, Object>>	
● retrieveHotItems(): List<HashMap<String, Object>>	
● listByFilter(queryMap: HashMap<String, List<String>>, curPage: int): HashMap<String, Object>	
■ totalListByFilter(queryMap: HashMap<String, List<String>>): int	
● houseByAgent(agntid: String): List<HashMap<String, Object>>	
● houseSoldByAgent(agntid: String): List<HashMap<String, Object>>	
● houseLikeByAgent(agntid: String): List<HashMap<String, Object>>	
● houseByRegisterDate(agntid: String): List<HashMap<String, Object>>	
● houseSoldByAgentCount(agntid: String): List<HashMap<String, Object>>	
● getLastCode(htype: String): String	
● houseRegister_info(infoDTO: HouseInfoDTO): int	
● houseRegister_price(priceDTO: HousePriceDTO): int	
● houseRegister_option(optionDTO: HouseOptionDTO): int	
● houseUpdate_info(infoDTO: HouseInfoDTO): int	
● houseUpdate_price(priceDTO: HousePriceDTO): int	
● houseUpdate_option(optionDTO: HouseOptionDTO): int	
● houseDel(list: List<String>): int	
● updateCntWish(hcode: String): int	
■ getCntWish(hcode: String): int	
● addWish(dto: HouseWishlistDTO): int	
■ getNoOfWishes(dto: HouseWishlistDTO): int	
■ duplicateHouseCheck(dto: HouseWishlistDTO): int	
● houseChange(list: List<List<String>>): int	
● houseRetrieve(hcode: String): HouseInfoDTO	
● housePrice(hcode: String): HousePriceDTO	
● houseOption(hcode: String): HouseOptionDTO	
● selectRcnlist(userid: String): List<HouseRcnlistDTO>	
● rcnListAllDone(rList: List<HouseRcnlistDTO>): int	
● deleteRcnlist(userRcnList: List<Long>): int	
● rcnHouseInfo(hCodeList: List<String>): List<HashMap<String, Object>>	
● selectWishlist(userid: String): List<HouseWishlistDTO>	

## Method Summary

public HashMap <String, Object>	<b>searchList</b> (String search, int curPage) // 매물 검색 리스트 반환
private int	<b>totalListBySearch</b> (String search) // 검색 매물 수
public List<HashMap <String, Object>>	<b>retrieveAllItems</b> () // 전체 매물 반환
Public List<HashMap <String, Object>>	<b>retrieveNewItem</b> () // 신매물 반환
private List<HashMap <String, Object>>	<b>retrieveHotItems</b> () // 인기매물 반환

Method Summary	
public <b>HashMap</b> <b>&lt;String, Object&gt;</b>	<b>listByFilter</b> ( <b>HashMap</b> < <b>String</b> , <b>List</b> < <b>String</b> >> queryMap, <b>int</b> cutPage) // 매물 필터 리스트
private <b>int</b>	<b>totalListByFilter</b> ( <b>HashMap</b> < <b>String</b> , <b>List</b> < <b>String</b> >> queryMap) // 매물 필터 리스트 수
public <b>HashMap</b> <b>&lt;String, Object&gt;</b>	<b>houseByAgent</b> ( <b>String</b> agntid) // 로그인한 중개인이 등록한 매물 리스트
public <b>List</b> < <b>HashMap</b> <b>&lt;String, Object&gt;&gt;</b>	<b>houseSoldByAgent</b> ( <b>String</b> agntid) // 로그인한 중개인이 등록한 매물 중 중개완료된 리스트
public <b>List</b> < <b>HashMap</b> <b>&lt;String, Object&gt;&gt;</b>	<b>houseLikeByAgent</b> ( <b>String</b> agntid) // 로그인한 중개인이 등록한 매물 중 좋아요 많은 순
public <b>List</b> < <b>HashMap</b> <b>&lt;String, Object&gt;&gt;</b>	<b>houseByRegisterDate</b> ( <b>String</b> agntid) // 로그인한 중개인이 등록한 매물의 등록일자별 리스트
public <b>List</b> < <b>HashMap</b> <b>&lt;String, Object&gt;&gt;</b>	<b>houseSoldByAgentCount</b> ( <b>String</b> agntid) // 로그인한 중개인이 등록한 매물 중 중개완료된 매물의 수
public <b>String</b>	<b>getLastCode</b> ( <b>String</b> htype) // 매물등록시 요청된 매물 타입의 마지막 코드 반환
public <b>int</b>	<b>houseRegister_info</b> ( <b>HouseInfoDTO</b> infoDTO) // house_info 테이블의 매물 등록
Public <b>int</b>	<b>houseRegister_price</b> ( <b>HousePriceDTO</b> priceDTO) // house_price 테이블의 매물 등록
public <b>int</b>	<b>houseRegister_option</b> ( <b>HouseOptionDTO</b> optionDTO) // house_option 테이블의 매물 등록
public <b>int</b>	<b>houseUpdate_info</b> ( <b>HouseInfoDTO</b> infoDTO) // house_info 테이블의 매물 수정
public <b>int</b>	<b>houseUpdate_price</b> ( <b>HousePriceDTO</b> priceDTO) // house_price 테이블의 매물 수정
public <b>int</b>	<b>houseUpdate_option</b> ( <b>HouseOptionDTO</b> optionDTO) // house_option 테이블의 매물 수정
public <b>int</b>	<b>houseDel</b> ( <b>List</b> < <b>String</b> > list) // 매물삭제
public <b>int</b>	<b>updateCntWish</b> ( <b>String</b> hcode) // 화면에서 좋아요 버튼 눌렀을 시 해당 매물의 좋아요 수치 증가
private <b>int</b>	<b>getCntWish</b> ( <b>String</b> hcode) // 매물 cntwish 값 가져오기
private <b>int</b>	<b>addWish</b> ( <b>HouseWishlistDTO</b> dto) // 유저가 찜한 매물 저장하기
private <b>int</b>	<b>getNoOfWishes</b> ( <b>HouseWishlistDTO</b> dto) // 한 유저당 찜 한 개수 가지고 오기
public <b>int</b>	<b>duplicateHouseCheck</b> ( <b>HouseWishlistDTO</b> dto) // 유저가 동일 매물 이미 찜했는지 않았는지 체크
public <b>int</b>	<b>houseChange</b> ( <b>List</b> < <b>List</b> < <b>String</b> >> list) // 매물 중개중/중개완료 여부 저장

## Method Summary

public <a href="#">HouseInfoDTO</a>	<b>houseRetrive</b> ( <a href="#">String</a> hcode) // 매물 코드를 통해 매물의 위치, 매물명 등의 정보를 조회하는 메소드
public <a href="#">HousePriceDTO</a>	<b>housePrice</b> ( <a href="#">String</a> hcode) // 매물 코드를 통해 매물의 가격 등의 정보를 조회하는 메소드
public <a href="#">HouseOptionDTO</a>	<b>houseOption</b> ( <a href="#">String</a> hcode) // 매물 코드를 통해 매물의 옵션을 조회하는 메소드
public <a href="#">List</a> <a href="#">&lt;HouseRcnlistDTO&gt;</a>	<b>selectRcnlist</b> ( <a href="#">String</a> userid) // 로그인 정보를 통해 최근 본 방을 조회하는 메소드
public <a href="#">int</a>	<b>rcnListAllDone</b> ( <a href="#">List</a> <a href="#">&lt;HouseRcnlistDTO&gt;</a> rList) // 최근 본 방을 저장하는 메소드
public <a href="#">int</a>	<b>deleteRcnlist</b> ( <a href="#">List</a> <a href="#">&lt;Long&gt;</a> userRcnList) // 최근 본 방을 삭제하는 메소드
public <a href="#">List</a> <a href="#">&lt;HashMap</a> <a href="#">&lt;String, Object&gt;&gt;</a>	<b>rcnHouseInfo</b> ( <a href="#">List</a> <a href="#">&lt;String&gt;</a> hCodeList) // 회원의 최근 본 방과 찜한 방의 매물 코드를 통해 매물 정보를 조회하는 메소드
public <a href="#">List</a> <a href="#">&lt; HouseWishlistDTO&gt;</a>	<b>selectWishlist</b> ( <a href="#">String</a> userid) // 로그인 정보를 통해 찜한 방을 조회하는 메소드

## ③ BoardDAO



com.dao.BoardDAO	
+	logger: Logger
△	session: SqlSessionTemplate
●	boardList(hcode: String): List<BoardDTO>
●	deleteBoard(pcode: String): int
●	updateBoard(board: BoardDTO): int
●	confUpBoard(list: HashMap<String,String>): int
●	agentHouseBoard(agntid: String): List<HashMap<String,Object>>

## Method Summary

public List<BoardDTO>	<b>boardList</b> (String hcode) // 매물 코드를 통해 문의글을 조회하는 메소드
public int	<b>deleteBoard</b> (String pcode) // 문의글을 삭제하는 메소드
public int	<b>updateBoard</b> (BoardDTO board) // 문의글을 수정하는 메소드
public List<HashMap<String, Object>>	<b>agentHouseBoard</b> (String agentid) // Angular 로 구현한 화면에서 중개인 페이지에 들어왔을 때 로그인한 중개인이 올린 매물과 관련되 문의사항 반환

#### iv. Other Class

##### ④ LoginCheckInterCeptor

 com.interceptor.LoginCheckInterceptor
 preHandle(request: HttpServletRequest, response: HttpServletResponse, handler: Object): boolean

### Method Summary

public boolean	<b>preHandle</b> (HttpServletRequest request, HttpServletResponse response, Object handler) // 로그인 session 이 유지되고 있는지 확인
----------------	---