



API 명세서

▼ 회원 관련 API

Member Domain

```
@Entity
@Getter @Setter @ToString
@Table(name = "member")
public class Member {
    @Column(name = "member_id")
    private Long id; // 예약자 번호

    private String major; // 전공

    @Column(name = "sno")
    private String studentNo; // 학번

    @Column(name = "phn")
    private String phoneNo; // 전화번호

    @Column(name = "auth")
    private AuthState authority; // UNI_STUDENT, POST_STUDENT, PROFESSOR, OFFICE

    private String name; // 회원의 이름

    @Column(name = "pwd")
    private String password;

    @Column(name = "cnt")
    private int cnt;
}
```

MemberDto Domain

```
public class MemberDto {
    private Long id; // 예약자 번호
    private String studentNo; // 학번 (Login-ID)
    private String name;
    private int major;
    private String phn;
    private String email;
    private AuthState authority;
    private int reservationCnt;
    private int noShowCnt;
}
```

회원 정보 관련 API

Aa index	method	URL	Description	권한 설명
<u>01-01</u>	GET	<u>/member/list</u>	전체 멤버 리스트 가져오기	
<u>01-02</u>	GET	<u>/member/{sno}</u>	해당 sno(학번)의 member 가져오기	
<u>01-03</u>	DELETE	<u>/member/delete/{sno}</u>	해당 sno(학번)의 member 정보 삭제	
<u>01-04</u>	GET	<u>/member/find-phone/{sno}</u>	해당 sno(학번)의 member의 전화번호 뒷자리 가져오기	
<u>01-05</u>	POST	<u>/member/set-password/{sno}</u>	해당 sno(학번)의 member의 비밀번호를 전화번호 뒷자리로 초기화 하기	
<u>01-06</u>	PUT	<u>/member/update-pwd/{sno}</u>	해당 sno(학번)의 member의 비밀번호를 수정 한다	
<u>01-07</u>	POST	<u>/member/insert</u>	새로운 회원 추가	
<u>01-08</u>	POST	<u>/set-cnt/{sno}</u>	회원의 권한에 따른 cnt 설정	

▼ 로그인 관련 API

LoginDto

```
public class LoginDto {
    private String studentNo; // 학번 (Login-ID)
    private String password;
}
```

로그인 정보 관련 API

Aa index	method	URL	Description	권한 설명
<u>01-01</u>	POST	<u>/auth/login</u>	로그인하기	
<u>01-02</u>	GET	<u>/auth/logout</u>	로그아웃하기	
<u>01-03</u>	GET	<u>/auth/checkLogin</u>	로그인여부 확인	

▼ 예약 관련 API

TimDto

```

public class Time {
    LocalDateTime start;
    LocalDateTime end;

    public Time(LocalDateTime start, LocalDateTime end) {
        this.start = start;
        this.end = end;
    }
}

```

ReservationInfo

```

public class ReservationInfo {
    private LocalDateTime start; // 예약 시작 시간
    private LocalDateTime end; // 예약 종료 시간
    private ReservationStatus status; // 예약 상태 RESERVED, FINISHED
    private Boolean regular; // 정기 예약 여부
}

```

Reservation

```

@Entity
@Setter @Getter @ToString
@Table(name = "reservation_log")
public class Reservation {
    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "reservation_id")
    private Long id; // 예약 번호

    private LocalDateTime start; // 예약 시작 시간

    private LocalDateTime end; // 예약 종료 시간

    private String status; // 예약 상태 (RESERVED, FINISHED, CANCELED)

    private Boolean regular; // 정기 예약 여부

    @OneToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "room_id")
    private Room room;

    @OneToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "member_id")
    private Member member;
}

```

DateByRoomDto

```

public class DateByRoomDto {
    private Integer year;
    private Integer month;
    private Integer day;
}

```

```
private Long roomId;
}
```

MonthDateByRoomDto

```
public class MonthDateByRoomDto {
    private Integer year;
    private Integer month;
    private Long roomId;
}
```

DateDto

```
private Integer year;
private Integer month;
private Integer day;
```

Reservation Domain

예약 정보 관련 API

Aa index	method	URL	Description	권한 설명
<u>01-02</u>	GET	<u>/reserve/time-check</u>	선택된 예약 시간이 빈 상태인지 확인	
<u>01-03</u>	GET	<u>/reserve/get</u>	해당 reservation_id의 reservation 객체 정보 반환	
<u>01-04</u>	GET	<u>/reserve/manager-list</u>	(관리자 전용)전체 예약 리스트 반환	
<u>01-05</u>	GET	<u>/reserve/user-list</u>	현재 로그인된 사람의 전체(REERVED, FINISHED, CANCELED) 예약 리스트 반환	
<u>01-0</u>	GET	<u>/reserve/user-list-canceled</u>	해당 로그인된 사람의 취소된(CANCELED) 예약 리스트 반환	
<u>01-0</u>	GET	<u>/reserve/user-list-reserved</u>	해당 로그인된 사람의 예약 상태(REERVED)인 예약 리스트 반환	
<u>01-08</u>	GET	<u>/reserve/user-list-finished</u>	해당 로그인된 사람의 끝난(FINISHED) 예약 리스트 반환	
<u>01-08</u>	GET	<u>/reserve/list-using-sno</u>	해당 학번의 전체 예약 리스트 반환	
<u>01-17</u>	POST	<u>/reserve/time-list-room</u>	연도-월-일 데이터를 보내주면 해당 날짜의 예약실 별 예약 시간 리스트를 반환	

Aa index	method	URL	Description	권한 설명
<u>01-17</u>	POST	<u>/reserve/get-reserve-list-all</u>	연도-월-일 데이터를 보내주면 해당 날짜의 예약실 별 예약 정보 리스트를 반환	
<u>01-18</u>	POST	<u>/reserve/today-time-check-room</u>	연도-월-일 데이터를 보내주면 해당 날짜의 예약실별 시간대별 예약 여부 리스트를 반환	
<u>01-18</u>	POST	<u>/reserve/today-time-check</u>	연도-월-일 데이터를 보내주면 해당 날짜의 예약실별 시간대별 예약 여부 리스트를 반환	
<u>01-19</u>	POST	<u>/reserve/month-reserve-check</u>	연도-월 데이터를 보내주면 한달동안의 날짜의 시간대별 예약 여부 리스트를 반환	
<u>01-09</u>	DELETE	<u>/reserve/delete-one</u>	해당 reservation_id의 reservation 객체 정보 삭제	
<u>01-10</u>	DELETE	<u>/reserve/login-delete-all</u>	현재 로그인된 사람의 전체 예약 내역 삭제	
<u>01-11</u>	DELETE	<u>/reserve/delete-all</u>	해당 학번의 사람의 전체 예약 내역 삭제	
<u>01-12</u>	DELETE	<u>/reserve/reset</u>	모든 예약 내역 삭제(관리자 전용)	
<u>01-13</u>	PUT	<u>/reserve/set-status-reserved</u>	예약 초기 상태로 설정	
<u>01-13</u>	PUT	<u>/reserve/set-status-finished</u>	예약 상태 끝 → 예약 상태 업데이트	
<u>01-14</u>	PUT	<u>/reserve/set-status-canceled</u>	예약 상태 취소 → 예약 상태 업데이트	
<u>01-16</u>	POST	<u>/reserve/</u>	선택된 예약 시간이 빈 상태라면, 현재 로그인된 사람으로 예약	
<u>01-17</u>	POST	<u>/reserve/today-reserve-cnt-room</u>	날짜를 입력하면 해당 roomId별 날짜의 예약 건수를 반환	
<u>01-17</u>	POST	<u>/reserve/today-reserve-cnt-all</u>	날짜를 입력하면 해당 날짜의 총 예약 건수를 반환	
<u>01-17</u>	PATCH	<u>/reserve/check-noshow</u>	사용자가 인증 요청을 할 때 이용하는 url 로 예약된 시간대에 요청을 할경우 노쇼 인증이됩니다. 예약된 시간대가 아닐경우 요청을 할경우 오류가 뜨게 됩니다.	
<u>01-17</u>	GET	<u>/reserve/get-reserve-cnt-all</u>	FINISHED 상태의 전체 예약의 수 반환	
<u>01-17</u>	GET	<u>/reserve/get-reserve-cnt-authority</u>	FINISHED 상태의 권한별 예약 수 반환	

▼ 관리자 관련 API

Management Domain

```
@Table(name = "MANAGEMENT")
public class Management {
    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    // 예약 가능 횟수
    @ColumnDefault("3")
    private int univ_cnt; // 대학생
    @ColumnDefault("5")
    private int post_cnt; // 대학원생
    @ColumnDefault("10")
    private int office_cnt; // 사무실
    @ColumnDefault("10")
    private int pro_cnt; // 교수

    // 예약 시간 간격 간격 기준: 시간
    @ColumnDefault("2")
    private int univ_hour_gap; // 대학생
    @ColumnDefault("3")
    private int post_hour_gap; // 대학원생
    @ColumnDefault("6")
    private int office_hour_gap; // 사무실
    @ColumnDefault("6")
    private int pro_hour_gap; // 교수

    // 예약 시간 간격 간격 기준: 일
    @ColumnDefault("2")
    private int univ_day_gap; // 대학생
    @ColumnDefault("7")
    private int post_day_gap; // 대학원생
    @ColumnDefault("365")
    private int office_day_gap; // 사무실
    @ColumnDefault("365")
    private int pro_day_gap; // 교수

    // 예약 가능 시간 간격 기준: 주
    @ColumnDefault("1")
    private int univ_week_gap; // 대학생
    @ColumnDefault("2")
    private int post_week_gap; // 대학원생
    @ColumnDefault("12") // (3 month)
    private int office_week_gap; // 사무실
    @ColumnDefault("12") // (3 month)
    private int pro_week_gap; // 교수
}
```

ManagementDto

```
public class ManagementDto {
    private int univ_cnt; // 대학생
    private int post_cnt; // 대학원생
    private int office_cnt; // 사무실
    private int pro_cnt; // 교수
}
```

```

// 예약 시간 간격 간격 기준: 시간
private int univ_hour_gap; // 대학생
private int post_hour_gap; // 대학원생
private int office_hour_gap; // 사무실
private int pro_hour_gap; // 교수

// 예약 시간 간격 간격 기준: 일
private int univ_day_gap; // 대학생
private int post_day_gap; // 대학원생
private int office_day_gap; // 사무실
private int pro_day_gap; // 교수

// 예약 가능 시간 간격 기준: 주
private int univ_week_gap; // 대학생
private int post_week_gap; // 대학원생
private int office_week_gap; // 사무실
private int pro_week_gap; // 교수
}

```

관리자 관련 API

Aa index	≡ method	🔗 URL	≡ Description	≡ 권한 설명
<u>01-01</u>	PUT	<u>manage/cnt-reset</u>	전체 회원의 권한별 예약 가능 횟수 초기화	
<u>01-02</u>	PUT	<u>manage/update-cnt</u>	세 권한별 예약 가능 횟수 업데이트	
<u>01-03</u>	PUT	<u>manage/update-hour-gap</u>	세 권한별 사용가능시간대설정	
<u>01-03</u>	PUT	<u>manage/update-day-gap</u>	세 권한별 사용가능 일수설정	
<u>01-04</u>	GET	<u>/manage/univ-cnt</u>	학생 권한 cnt 반환	
<u>01-05</u>	GET	<u>/manage/post-cnt</u>	대학원생 권한 cnt 반환	
<u>01-06</u>	GET	<u>/manage/pro-cnt</u>	교수 권한 cnt 반환	
<u>01-06</u>	GET	<u>/manage/office-cnt</u>	사무실 권한 cnt 반환	
<u>01-07</u>	GET	<u>/manage/univ-hour-gap</u>	학생 권한 hour gap반환	
<u>01-08</u>	GET	<u>/manage/post-hour-gap</u>	대학원생 권한 hour gap 반환	
<u>01-09</u>	GET	<u>/manage/office-hour-gap</u>	사무실 권한 hour gap 반환	
<u>01-09</u>	GET	<u>/manage/pro-hour-gap</u>	교수 권한 hour gap 반환	
<u>01-07</u>	GET	<u>/manage/univ-day-gap</u>	학생 권한 day gap반환	

Aa index	≡ method	🔗 URL	≡ Description	≡ 권한 설명
<u>01-08</u>	GET	<u>/manage/post-day-gap</u>	대학원생 권한 day gap 반환	
<u>01-09</u>	GET	<u>/manage/office-day-gap</u>	사무실 권한 day gap 반환	
<u>01-09</u>	GET	<u>/manage/pro-day-gap</u>	교수 권한 day gap 반환	
<u>01-10</u>	GET	<u>/manage/</u>	전체 관리정보 객체 반환	
<u>01-10</u>	GET	<u>/manage/member</u>	학생 검색 로직 → 이름 또는 학번	
<u>01-10</u>	GET	<u>/manage/date</u>	날짜별 예약 리스트 반환	
<u>01-10</u>	GET	<u>/manage/no-show</u>	입력된 NoShowCnt 이상의 값의 noshow 값을 가진 멤버리스트 반환	
<u>01-10</u>	PATCH	<u>/manage/update</u>	management의 전체 정보 수정	
<u>01-10</u>	GET	<u>/manage/get-noshow/{sno}</u>	해당 학번의 멤버의 노쇼 횟수 반환	
<u>01-10</u>	GET	<u>/manage/get-noshow-all</u>	전체 멤버의 노쇼 횟수 반환	
<u>01-10</u>	PATCH	<u>/reset-noshow/{sno}</u>	해당 학번의 노쇼 횟수를 0으로 리셋	
<u>01-10</u>	PATCH	<u>/reset-noshow-all</u>	전체 멤버의 노쇼 횟수를 0으로 리셋	
<u>01-10</u>	GET	<u>/manage/get-noshow-authority</u>	입력된 authority 에 해당하는 멤버들의 노쇼 횟수 반환	

▼ 회의실 관련 API

Room Domain

```

@Entity
@Data
@DynamicInsert
@Table(name = "room")
public class Room {
    @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "room_id")
    private Long id; // 회의실 id

    private String name; // 회의실 이름

    @Enumerated(EnumType.STRING)
    @Column(name = "building_name")
    private SejongBuildingName buildingName;
    /** 회의실 빌딩이름 (집현관, 대양홀, 모짜르트홀, 군자관, 광개토관, 이당관, 진관홀, 용덕관, 영실관, 충무관, 율곡관,

```



```

다산관, 학술정보원, 우정당, 대양AI센터, 세종관, 학생회, 새날관, 무방관)**/

@ColumnDefault("0")
private int cap; // 수용인원

private String info; // 회의실 상세정보

@ColumnDefault("0")
private int board; // 화이트보드 개수

@ColumnDefault("0")
private int tv; // 티비 개수

@ColumnDefault("0")
private int com; // 컴퓨터 개수

private String picture; // 사진 경로

@Column(columnDefinition = "TINYINT(1)", nullable = false)
@ColumnDefault("0")
private Boolean empty; // 회의실 예약 가능 여부

@Column(columnDefinition = "TINYINT(1)", nullable = false)
@ColumnDefault("0")
private Boolean wifi; // 와이파이 여부

@Column(columnDefinition = "TINYINT(1)", nullable = false)
@ColumnDefault("0")
private Boolean bim_projector; // 빔프로젝터 여부

}

```

회의실 정보 관련 API

Aa index	method	URL	Description	권한 설명
<u>01-01</u>	GET	<u>/room/detail/{room_id}</u>	해당 id의 회의실 정보 가져오기	
<u>01-02</u>	PATCH	<u>/room/update</u>	회의실 정보 수정	
<u>01-03</u>	DELETE	<u>/room/delete/{room_id}</u>	회의실 정보 삭제	
<u>01-04</u>	POST	<u>/room/insert</u>	회의실 정보 추가	
<u>01-05</u>	GET	<u>/room/list</u>	전체 회의실 리스트 반환	

▼ 공지사항 관련 API

Notice Domain

```

@Column(name = "notice_id")
private Long id; // 공지 id

```

```
private String title; // 공지 제목

@Lob
private String content; // 공지 내용

@Column(name = "created_at")
@ColumnDefault(value = "CURRENT_TIMESTAMP")
private LocalDateTime createdAt = LocalDateTime.now(); // 공지 게시글 생성 날짜
```

CreateRequestNoticeDto

```
private String title; // 공지 제목

private String content; // 공지 내용
```

UpdateRequestNoticeDto

```
private Long noticeId;
private String title; // 공지 제목
private String content; // 공지 내용
```

공지사항 관련 API

Aa index	method	URL	Description	권한 설명
<u>01-01</u>	POST	<u>/notice/insert</u>	공지 게시글 입력	
<u>01-02</u>	GET	<u>/notice/list</u>	공지 게시글 리스트 반환	
<u>01-03</u>	GET	<u>/notice/detail/{notice_id}</u>	공지 게시글 상세 정보	
<u>01-03</u>	PATCH	<u>/notice/update</u>	공지 게시글 수정	
<u>01-03</u>	DELETE	<u>/notice/delete/{notice_id}</u>	공지 게시글 삭제	

▼ 노쇼 관련 API

노쇼 관련 API

Aa index	method	URL	Description	권한 설명
<u>01-01</u>	PATCH	<u>/reserve/check-noshow</u>	예약한 시간대에 노쇼를 체크 요청	
<u>01-02</u>	GET	<u>/reserve/get-noshow/{sno}</u>	해당 학번의 멤버의 노쇼 cnt 값을 반환한다.	

Aa index	⋮ method	🔗 URL	≡ Description	≡ 권한 설명
<u>01-03</u>	PATCH	<u>/reserve/reset-noshow/{sno}</u>	해당 학번의 노쇼 cnt 값을 0으로 리셋한다.	

▼ 이메일 관련 API

노쇼 관련 API

Aa index	⋮ method	🔗 URL	≡ Description	≡ 권한 설명
<u>01-01</u>	POST	<u>/email</u>	예약한 시간대에 노쇼를 체크 요청	
<u>01-02</u>	POST	<u>/email-regular</u>	정기예약 이메일을 보낸다.	

▼ 관리자 관련 API

Admin

```
private Long id; // admin primary id

private String loginId; // login id

private String phoneNo; // 전화번호

private String email; // email

private String password;

private AuthState authority; // 권한 UNI_STUDENT, POST_STUDENT, PROFESSOR, OFFICE
```

AdminDto

```
private String loginId; // login id

private String phoneNo; // 전화번호

private String email; // email

private AuthState authority; // 권한 UNI_STUDENT, POST_STUDENT, PROFESSOR, OFFICE
```

UpdateAdminDto

```
private String phoneNo; // 전화번호
private String email; // email
private String password; // password
```

관리자 관련 API

Aa index	⋮ method	🔗 URL	≡ Description	≡ 권한 설명
<u>01-01</u>	GET	<u>/admin</u>	관리자 정보 반환	
<u>01-03</u>	PATCH	<u>/admin/update</u>	관리자 정보 수정	