



API 명세서

# 일 번호	통신방향	주요 개발 현 황	주요 관련 기능	세부 기능	Method Name	Description	다중 선택	HTTP Method	Endpoint URL	Event	Request	Response	Aa 기타 추가 설명
1	클라이언트 → 스프링	완료	일반 회원가입	ID 중복 확인	checkDuplicateId	ID가 DB에 있는지 확인	HTTP	POST	/v1/user/check-id	userT.select (id) /session.save (id)	{ "id": "사용자id" }	200, Ok - 해당 id 사용 가능 409, Conflict - DB에 중복 id가 있음 400, Bad Request	ID 중복확인
2	클라이언트 → 스프링	완료	일반 회원가입	이메일 인증 번호 전송	sendEmailCode	이메일 중복 확인 후 DB에 등록 정보 없으면 코드 전송	HTTP	POST	/v1/user/email/send-code	userT.select (phone_number) / create (code) / send (code) / session.save (이메일 인증 코드)	{ "email": "이메일" }	200, Ok 409, Conflict 400, Bad Request	제목 없음
3	클라이언트 → 스프링	완료	일반 회원가입	이메일 인증 번호 확인	verifyEmailCode	전송한 코드와 입력 값이 같은지 검증	HTTP	POST	/v1/user/email/verify-code	session.get (이메일 인증 코드) / verify (이메일 인증 코드) / session.save (phone_number)	{ "code": "입력 code" }	200, Ok 401, Unauthorized - 인증 코드가 일치하지 X 400, Bad Request	제목 없음
4	클라이언트 → 스프링	완료	일반 회원가입	회원가입	registerUser	사용자 입력 회원정보 등록	HTTP	POST	/v1/user/regist 초대링크로 들어와도 동일 메서드	session.get (id) / verify (id) / session.get (phone_number) / verify (phone_number) / make (salt) / encrypt (password) / userT.create (user) / session.remove	{ "id": "사용자id", "password": "사용자pw", "nickname": "닉네임", "email": "이메일", "imageNum": "이미지 번호", "socialType": 0, "gender": "사용자성별", "birth": "2025-01-31", "mainDeviceId": 0 }	200, Ok 401, Unauthorized - 세션 없음, 세션에 ID, 핸드폰 번호 속성 없음, 속성은 있으나 값 일치 안 함 409, Conflict - 초대링크로 들어온 경우 초대멤버가 다 차서 자리 없음 - 회원가입 로직은 수행됨 400, Bad Request	회원가입
5	클라이언트 → 스프링	완료	카카오 회원가입	카카오 로그인	loginWithKakao		HTTP	GET	/v1/auth/kakao/login	redirect		[카카오 서버로 리다이렉트] Location: " https://kauth.kakao.com/oauth/authorize" + "?client_id=" + kakaoApiKey + "&redirect_uri=" + kakaoRedirectUrl + "&response_type=code"	카카오 로그인
6		완료	카카오 회원가입	카카오 콜백	kakaoCallback		HTTP	GET	/v1/auth/kakao/call-back	userT.select (id & social_type==2) - create (token) / redis.save (token) - session.save (id)	/call-back?code={인증코드}	상황에 따라 클라이언트 애플리케이션으로 리다이렉트 - 로그인 성공 - 초대링크로 들어온 경우 초대멤버가 다 차서 자리 없음 - 로그인 로직은 수행됨 http://localhost:5173/login/social?social=true&status=login?group=false → 리다이렉션 시 헤더 이용 불가능 하기에 JWT 토큰 쿠키 전달 통일 - 회원가입 필요 ?email="사용자 카카오 이메일" - 오류 발생 ?social=false	제목 없음
71	클라이언트 → 스프링	완료	카카오 회원가입	카카오 토큰 발급	issueAccessToken	카카오 로그인 시 access 토큰 발급	HTTP	POST	/v1/auth/kakao/token/issue		쿠키로 리프레시 토큰 들고 와라	헤더에 access 토큰 발급	제목 없음
7	클라이언트 → 스프링	완료	카카오 회원가입	카카오 회원가입	registerKakaoUser		HTTP	POST	/v1/auth/kakao/regist	session.get (id) / session.get (phone_number) / verify (phone_number) / make (salt) / encrypt (password) / userT.create (user) / session.remove / create (token) / redis.save (token) / redirect	{ "password": "사용자pw", "nickname": "닉네임", "phoneNumber": "폰번호", "imageNum": "이미지 번호", "socialType": 2, "gender": "사용자성별", "birth": "2025-01-31" }	200, Ok 401, Unauthorized - 세션 없음, 세션에 ID, 소셜타입 속성 없음 409, Conflict - 초대링크로 들어온 경우 초대멤버가 다 차서 자리 없음 - 회원가입 로직은 수행됨 400, Bad Request	제목 없음
8	클라이언트 → 스프링	완료	구글 회원가입	구글 로그인	googleLogin		HTTP	GET	/v1/auth/google/login	redirect		[구글 서버로 리다이렉트] Location: " https://accounts.google.com/o/oauth2/v2/auth" + "?client_id=" + clientId + "&redirect_uri=" + redirectUri + "&response_type=code" + "&scope=openid email profile" + "&access_type=offline" + "&prompt=consent"	제목 없음
9		완료	구글 회원가입	구글 콜백	googleCallback		HTTP	GET	/v1/auth/google/call-back	userT.select (id & social_type==1) / session.save (id) - create (token) / redis.save (token) - session.save (id)	/call-back?code={인증코드}	상황에 따라 클라이언트 애플리케이션으로 리다이렉트 - 로그인 성공 - 회원가입 필요 - 오류 발생	제목 없음
77		완료	구글 회원가입	구글 토큰 발급	issueAccessToken	구글 로그인 시 access 토큰 발급	HTTP	POST	/v1/auth/google/token/issue		쿠키로 리프레시 토큰 들고 와라	헤더에 access 토큰 발급	제목 없음
10	클라이언트 → 스프링	완료	구글 회원가입	구글 회원가입	registerGoogleUser		HTTP	POST	/v1/auth/google/regist	session.get (id) / verify (id) / session.get (phone_number) / verify (phone_number) / make (salt) / encrypt (password) / userT.create (user) / session.remove / create (token)	{ "password": "사용자pw", "nickname": "닉네임", "phoneNumber": "폰번호", "imageNum": "이미지 번호", "socialType": 1, "gender": "사용자성별",	200, Ok 401, Unauthorized - 세션 없음, 세션에 ID, 소셜타입 속성 없음 400, Bad Request	제목 없음

# 열 번호	통신방향	개 발 연 호	관 련 기 능	세부 기능	Method Name	Description	다중 선택	HTTP Method	Endpoint URL	Event	Request	Response	Aa 기타 추가 설명
										/ redis.save (token) / redirect	" "birth": "2025-01-31"		
11	클라이언트 → 스프링	완	로그인	로그인	loginUser	토큰 발급	HTTP	POST	/v1/user/login 초대링크로 들어와도 동일 메서드	userT.select (id) / userSeculInfoT.select (id) / verify (password) / create (token) / redis.save (token)	{" id " : "입력 값", " "password " : "입력 값"}	200, Ok 401, Unauthorized - 비밀번호 틀림 403, Forbidden - 계정이 없음 409, Conflict - 초대링크로 들어온 경우 초대멤버가 다 차서 자리 없음 - 로그인 로직은 수행됨 400, Bad Request	jwt 토큰은 쿠키가 아닌 header에 담아주기 /
12	클라이언트 → 스프링	완	로그아웃	로그아웃	logoutUser	토큰 만료	HTTP	POST	/v1/user/logout	verify (token) / redis.delete (token)		200, Ok 400, Bad Request	제목 없음
13	클라이언트 → 스프링	완	기기등록	기기등록 정보 입력	registerDevice	serial, ip 주소 받음	HTTP	POST	/v1/device/add	verify (token) / deviceT.create (device)	{ " serial " : "입력값" " "ipAddress " : "입력값"}	200. Ok { " " id " : "디바이스 pk" } 409, Conflict - 이미 등록된 기기 403, Forbidden - 핸드셰이크 실패 400, Bad Request	제목 없음
14	RB → 스프링 스프링 → RB	완	기기연결	라즈베리파이 핸드셰이크	beforeHandshake	serial token 필요	HTTP TCP	GET	/v1/ws/device	verify (token) / deviceT.select (serial)	[Headers] GET /ws/device HTTP/1.1 Host: <백엔드주소 :8080> Upgrade: websocket Connection: Upgrade Sec-WebSocket-Key: <16바이트 숫자를 base64로 인코딩> Sec-WebSocket-Version: 13 Authorization: Bearer <serial을 SHA 256 암호화>	200, Ok (시리얼이 DB에 있음) HTTP/1.1 101 Switching Protocols Upgrade: websocket Connection: Upgrade Sec-WebSocket-Accept: <Base64-encoded value> 403, Forbidden - 어드민 권한없는데 요청	캐싱
73	RB → 스프링 스프링 → RB	완	기기연결	라즈베리파이 에 기기 아이디 전송	sendDeviceId	기기 serial에 해당 하는 id 를 전송하는 메서드	TCP			verify(token) / deviceT.select (serial)	[destination] "/app/DeviceInfo/Id" { "token": "id jwt" }	[구독 경로] "/topic/DeviceInfo/Id" + 시리얼 { " " id " : "기기 id" }	제목 없음
15	RB → 스프링	완	기기연결	라즈베리파이 기기 정보 업데이트	handleTempHumData	온/습도 정보 받기 (이후에는 30분 마다 up)	TCP			verify (token) / deviceT.update (temperature, humidity)	[destination] "/app/DeviceStatus/Sensor/TempHum" { "token": "id jwt", " "temperature " : 22.5 (float), " "humidity " : 60 (int) }	제목 없음	제목 없음
16	클라이언트 → 스프링	완	캡슐등록	캡슐 정보 입력	inputCapsuleInfo	슬롯에 장착된 캡슐 정보 캡슐 등록 뿐만 아니라 수정이 발생하면 16 ~ 22 반복	HTTP	POST	/v1/device/capsules	verify (token) /deviceT.update (name, slot)	{" id " : "pk 값", " "name " : "기기 이름", " "slot1 " : "0~2", " "slot2 " : "3~5", " "slot3 " : "0~8", " "slot4 " : "0~8"}	200, Ok 400, Bad Request	제목 없음
17	스프링 → RB	완	캡슐등록	라즈베리파이 에 캡슐 정보 보내기	sendCapsuleInfo		TCP			verify (token) /deviceT.update (slot_remainingRatio)	[destination] "/app/DeviceStatus/Capsule/Info" { " "slot1 " : "캡슐숫자", " "slot2 " : "캡슐숫자", " "slot3 " : "캡슐숫자", " "slot4 " : "캡슐숫자" }	제목 없음	제목 없음
18	RB → 스프링	완	캡슐잔여량조회	라즈베리파이 캡슐 잔여량 업데이트	handCapsuleRemainer		TCP			verify (token) /deviceT.update (slot_remainingRatio)	[destination] "/app/DeviceStatus/Capsule/Remainder" { " "token": "아이디 jwt", " "slot1RemainingRatio " : "90", " "slot2RemainingRatio " : "100", " "slot3RemainingRatio " : "80", " "slot4RemainingRatio " : "20" }	제목 없음	제목 없음
19	클라이언트 → 스프링	완	기본항등록	항 조합 등록	inputDefaultCombination	기본 항기 조합, 공간 크기 저장	HTTP	POST	/v1/device/set	verify (token) / combinationT.create / deviceT.update (default_combination, room_type)	{ " " id " : "디바이스 pk", " "room_type " : 0, "combination" : { " "choice1 " : "첫 번째 항", " "choice1Count " : "분사 수", " "choice2 " : "두 번째 항", " "choice2Count " : "분사 수", " "choice3 " : "세 번째 항", " "choice3Count " : "분사 수", " "choice4 " : "네 번째 항", " "choice4Count " : "분사 수" } }	200, Ok 400, Bad Request	제목 없음
20		완	기본항등록	탈취 모드 설정	setDeodorizationMode	디폴트 항과 인터벌 (15분) /off로 설정				autoScheduleT.create (sub_mode(2))			제목 없음
21		완	기본항	동작 모드 설정	setRestMode/ setExerciseMode	1번(운동), 2번(휴식), 인				combinationT.create (1 항) / combinationT.create (2			제목 없음

# 열 번호	통신방향	유기발현	관리기능	세부기능	Method Name	Description	다중선택	HTTP Method	Endpoint URL	Event	Request	Response	Aa 기타 추가 설명
			등록			터벌 (15분) / off로 설정				항) / autoScheduleT.create ((sub_mode(1)))			
22		완료	기본항등록	단순탐지모드 설정	setDetectionMode	디폴트항 설정				autoScheduleT.create ((sub_mode(0)))			제목 없음
75	클라이언트 → 스프링	완료	기본항수정	기본항수정	updateDefultCombination	기본항기조합 변경	HTTP	POST	/v1/device/set/update	verify(token) / combinationT.update	{ "id": "디바이스 id", "combination": { "id": "조합 아이디", "choice1": "첫 번째 항", "choice1Count": "분사 수", "choice2": "두 번째 항", "choice2Count": "분사 수", "choice3": "세 번째 항", "choice3Count": "분사 수", "choice4": "네 번째 항", "choice4Count": "분사 수" } }	200, Ok 400, Bad Request	제목 없음
84	클라이언트 → 스프링	완료	캡슐변경	캡슐변경	changeCapsuleInfo	캡슐을 변경	HTTP	POST	/v1/device/capsules/change		{ "id": "pk 값", "name": "기기 이름", "slot1": "0~2", "slot2": "3~5", "slot3": "0~8", "slot4": "0~8" }		제목 없음
83	클라이언트 → 스프링	완료	캡슐변경	캡슐변경에 따른 기본항 등록	changeDefaultCombination	캡슐을 등록한 이후 기본항 등록	HTTP	POST	/v1/device/set/change	공간 크기 같고 캡슐 정보 변경 안 하면 스케줄 삭제하지 않음	{ "id": "디바이스 pk", "room_type": 0, "combination": { "choice1": "첫 번째 항", "choice1Count": "분사 수", "choice2": "두 번째 항", "choice2Count": "분사 수", "choice3": "세 번째 항", "choice3Count": "분사 수", "choice4": "네 번째 항", "choice4Count": "분사 수" } }	200, Ok 400, Bad Request	제목 없음
23	클라이언트 → 스프링	완료	그룹기능	그룹초대	inviteToGroup	최초 그룹 생성자가 관리자 / 그룹 코드 및 URL 생성	HTTP	POST	/v1/group/invite	verify (token) / deviceT.select (id) / verify (admin_id == 요청 id) / make (invite_code) / redis.add (invite_code, group_id)	{ "deviceid": "디바이스 id" }	200, Ok { "inviteCode": "코드 8자리", "inviteLink": "초대링크" } 403, Forbidden - 해당 디바이스의 admin이 아니기에 초대 권한 없음 400, Bad Request	로그인 이후 토큰 만료로 인해 인증되지 않는 것은 401
24	클라이언트 → 스프링	완료	그룹기능	초대링크로 그룹 가입	joinGroupByLink	유효한 초대 코드인지 확인 / 초대 코드 검증 소셜로그인의 경우 ?state=초대code로 해서 파라미터 전달	HTTP	POST	/v1/group/verify-link	redis. get (invite_code)	{ "inviteCode": "코드 8자리" }	200, Ok { "adminNickname": "어드민 닉네임" } 410, Gone - 초대코드가 만료되어 레디스에 없음 400, Bad Request	초대링크로 그룹 가입
25	클라이언트 → 스프링	완료	그룹기능	초대 코드로 그룹 가입	joinGroupByCode	초대 코드 검증 및 그룹 등록 이미 가입한 사람만 허용	HTTP	POST	/v1/group/verify-code	verify (token) / redis. get (invite_code) / groupT.select(id) / 빈 멤버 자리 확인 / groupT.update	{ "inviteCode": "코드 8자리" }	200, Ok 403, Forbidden - 이미 어드민 혹은 멤버로 있는 회원 409, Conflict - 빈 멤버 자리 없음 410, Gone - 초대코드가 만료되어 레디스에 없음 400, Bad Request	403이 권한 문제로 접근 불가 422데이터는 맞지만 실행이 안됨 요청시
68	클라이언트 → 스프링	완료	그룹기능	디바이스별 그룹 조회	getGroupByDeviceId	디바이스당 그룹 정보 조회	HTTP	POST	/v1/group/info	verify (token) / deviceT.select(id) / groupT.select(id)	{ "deviceid": 2 }	200, Ok { "group": { "id": "그룹 id", "adminid": "어드민 id", "adminNickname": "이름", "member1id": "멤버 id", "member1Nickname": "이름", "member2id": "멤버 id", "member2Nickname": "이름", "member3id": "멤버 id", "member3Nickname": "이름", "member4id": "멤버 id", "member4Nickname": "이름" } }	제목 없음

# 열 번호	통신방향	유계발현	관리기능	세부기능	Method Name	Description	다중선택	HTTP Method	Endpoint URL	Event	Request	Response	Aa 기타 추가 설명
												} 403, Forbidden - 해당 그룹의 가입자가 아니기에 정보 권한 없음 400, Bad Request	
26	클라이언트 → 스프링	완료	그룹기능	그룹관리	deleteGroupMember	그룹 회원 삭제/ 관리자일 경우만 허용	HTTP	POST	/v1/group/member/delete	verify (token) / group T.select (그룹id) / 요청 id == admin_id? / groupT.delete(삭제 id)	{ " "groupid" : "그룹 id" "memberId" : "삭제하는 id" }	200, Ok 403, Forbidden - 해당 디바이스의 admin이 아니기에 삭제 권한 없음 404 Not Found - 삭제 요청 준 id가 그룹에 없음 - 그룹 정보를 찾을 수 없음 400, Bad Request	제목 없음
27	클라이언트 → 스프링	완료	그룹기능	그룹삭제	deleteGroup	어드민일 경우 그룹 자체 삭제	HTTP	POST	/v1/group/delete	verify (token) / group T.select (id) / 요청 id == admin_id? / groupT.update (멤버 null)	{ " "groupid" : "그룹id" }	200, Ok 403, Forbidden - 해당 디바이스의 admin이 아니기에 삭제 권한 없음 404 Not Found - 삭제 요청 준 그룹 id가 없음 400, Bad Request	제목 없음
28	클라이언트 → 스프링	완료	홈정보 반환	홈탭에 들어갔을 때 정보 반환	getHomeInfo	홈탭에 들어갈 경우 정보 반환	HTTP	POST	/v1/home/info	verify (token) / userT.select (id), / groupT.select (id), / deviceT.select (mainDeviceld), / autoScheduleT.select (mainDeviceld), / customScheduleT.select (mainDeviceld)		200, Ok { "user" : { " "nickname" : "닉네임", " "imgNum" : "프로사 번호", " "mainDeviceld" : "메인기기", }, "deviceIdsAndNames": { "deviceld": "기기이름" (map) }, "mainDevice" : { " "id" : "디바이스 id", " "name" : "디바이스 이름", " "groupid" ~1, " "roomType" : 0, " "slot1" ~1, " "slot1RemainingRatio" ~90, " "slot2" ~3, " "slot2RemainingRatio" ~90, " "slot3" ~1, " "slot3RemainingRatio" ~90, " "slot4" ~1, " "slot4RemainingRatio" ~90, " "mode" ~0~1, " "temperature" ~20, " "humidity" ~ 70, " "defaultCombination" ~1 }, "autoSchedules" : [{ " "id" : 2, " "combinationid" : 1, " "subMode" : 0, " "type" : null, " "interval" ~ 15 " "modeOn" : true }], "customSchedules" : [{ " "id" ~3, " "name" : "출근 전에" " "combinamtionid" : 2, " "combinationName" ~"상해", " "day" ~ 64, " "startTime" ~09:00:00, " "endTime" ~10:00:00 " "interval" ~ 15, " "modeOn" : true } }] "favorites" : [123, 234, 345] } 400, Bad Request	인터벌 받을
81	클라이언트 → 스프링	완료	토큰재발급	access 토큰 재발급	issueAccessToken	새로고침 시 토큰 재발급	HTTP	POST	/v1/user/token/issue	verify(access token) / verify(refresh token) / re-issue(access token)	쿠키로 리프레시 토큰, 헤더에 access 토큰 가져와	access 토큰 재발급 해서 줌	제목 없음
69	클라이언트 → 스프링	완료	대표기기 설정	대표기기 설정하기	setMainDevice	계정의 대표 기기를 설정	HTTP	POST	/v1/user/device/set	verify (token) / userT.update (main_device_id)	{ " "deviceld" : "설정 id" }	200, OK 400, Bad Request	제목 없음
70	클라이언트 → 스프링	완료	기기정보 반환	기기 정보 반환	getDeviceInfo	사용자 계정에 등록된 모든 기기 정보 반환	HTTP	POST	/v1/device/info	verify (token) /deviceT.select(id)	{ " "deviceIds" : [2, 3, 4 (난수임)	200, Ok { " "devices" : [제목 없음

# 열 번호	통신방향	호출발원	관련기능	세부기능	Method Name	Description	다중선택	HTTP Method	Endpoint URL	Event	Request	Response	Aa 기타 추가 설명
			반환			기기 id는 홈 탭에 진입할 때 이미 전달됨					<pre> } }</pre>	<pre>{ "id" : 2, "name" : "기기 이름", "groupId" ~: 1, "roomType" : 0, "slot" ~: 1, "slot1RemainingRatio" ~: 90, "slot2" ~: 3, "slot2RemainingRatio" ~: 90, "slot3" ~: 1, "slot3RemainingRatio" ~: 90, "slot4" ~: 1, "slot4RemainingRatio" ~: 90, "mode" ~: 0~1, "defaultCombination" ~: 44789384 } }</pre> 404, Not Found 400, Bad Request	
29	클라이언트 → 스프링	완료	모드 설정	스케줄/ 자동화 모드 설정	setDeviceMode	스케줄/ 자동화 모드 전환 시 호출됨	HTTP	POST	/v1/device/mode/set	verify (token) /deviceT.update (mode) if 스케줄 → 자동화 / autoScheduleT.select (device_id)	스케줄 → 자동화 { "deviceId" : 1, "mode" : true (1) } 자동화 → 스케줄 { "deviceId" : 1, "mode" : false (0) }	200, Ok 400, Bad Request	제목 없음
75	RB → 스프링 스프링 → RB	완료	모드 정보 반환	스케줄/ 자동화 모드 정보 전송	sendDeviceMode		TCP				[목적 경로] "/app/Mode" { "token": "아이디 jwt" }	[구독 경로] "/topic/Mode" + 디바이스 id 스케줄 { "mode" : "0" } 자동화 { "mode" : "1" }	제목 없음
30	스프링 → RB	완료	모드 설정	스케줄/ 자동화 모드가 수정 되었을 때 정보 전송	sendDeviceModeUpdate		TCP				[구독 경로] "/topic/Mode/Change" + deviceId 스케줄 { "mode" : "1" } 자동화 { "mode" : "0" }		제목 없음
72	RB → 스프링	완료	스케줄 모드	스케줄 정보 요청	sendCustomSchedules		TCP				[목적 경로] "/app/Schedule/Initial" { "token": "아이디 jwt" }	[구독 경로] "/topic/Schedule/Initial" + deviceId { "schedules": [{ "id" : "스케줄id" "deviceId" : "기기 id" "combination" : { "choice1" ~: 1, "choice1Count" ~: 3, "choice2" ~: 2, "choice2Count" ~: 2, "choice3" ~: null, "choice3Count" ~: null, "choice4" ~: null, "choice4Count" ~: null }, "startTime" ~: "09:00:00", "endTime" ~: "18:00:00", "interval" ~: 15 "modeOn" ~: true }] }	제목 없음
31	클라이언트 → 스프링	완료	스케줄 모드	시간 기반 예약 설정	setCustomSchedule	예약 이름, 기기 정보, 요일, 시작 시간, 종료 시간, 분사 주기, 커스텀 향 설정	HTTP	POST	/v1/custom/add	verify (token) / combinationT.create / customScheduleT.create	{ "name" : "예약 이름", "deviceId" : "기기 id", "day" : "비트마스킹" "combination" : { "name" : "시원 향" "choice1" ~: 1, "choice1Count" ~: 3, "choice2" ~: 2, "choice2Count" ~: 2, "choice3" ~: null, "choice3Count" ~: null, "choice4" ~: null, "choice4Count" ~: null }, "startTime" ~: "09:00:00", "endTime" ~: "18:00:00", "interval" ~: 15 "modeOn" ~: true }	200, Ok 403, Forbidden 이미 해당 시간대에 등록된 스케줄이 있음 400, Bad Request	combination → schedule

# 열 번호	통신방향	상태 발현	관련 기능	세부 기능	Method Name	Description	다중 선택	HTTP Method	Endpoint URL	Event	Request	Response	Aa 기타 추가 설명
											<pre>choice2Count": 2, " choice3": null, " choice3Count": null, " choice4": null, " choice4Count": null }, " startTime": "10:00:00", " endTime": "18:00:00", " interval": 15 " modeOn": true }</pre>		
78	스프링 → RB	완료	스케줄 모드	시간 기반 예약 추가 정보 전송	sendCustomScehdule	오늘에 해당할 경우 정보들 보냄	TCP				<p>[구독 경로] "/topic/Schedule/Add" + deviceId</p> <pre>{ "schedules": [{ " id": "스케줄id" " deviceId": "기기 id" "combination": { " choice1": 1, " choice1Count": 3, " choice2": 2, " choice2Count": 2, " choice3": null, " choice3Count": null, " choice4": null, " choice4Count": null }, " startTime": "09:00:00", " endTime": "18:00:00", " interval": 15 " modeOn": true } } }</pre>		제목 없음
32	클라이언트 → 스프링	완료	스케줄 모드	시간 기반 예약 개별 조회	getIndividualCustomSchedule	예약에 대한 기존 정보 조회 스케줄 id를 넘겨준다	HTTP	POST	/v1/custom/one	verify (token) / combinationT.select(id)	<pre>{ " combinationid": "조합 id" } }</pre>	<pre>200, Ok { "combination": { " id": 조합 id " name": 조합 이름, " choice1": 1, " choice1Count": 3, " choice2": 2, " choice2Count": 2, " choice3": null, " choice3Count": null, " choice4": null, " choice4Count": null } }</pre>	- 개별 조회는 언제 필요한지 / 컨트롤에 예약모드에서 각 각항 조합 <수 정> 누를 때 필 요 / combination 의 정보는 37번 API로 받은 데 이터 이용
33	클라이언트 → 스프링	완료	스케줄 모드	시간 기반 예약 수정	updateCustomSchedule	스케줄 id와 설정 정보를 넘김 수정된 정보를 저장하는 버튼을 누르면 호출	HTTP	POST	/v1/custom/update	verify (token) / combinationT.create / customScheduleT.update (id)	<pre>{ " id": 스케줄 id " name": "예약 이름", " deviceId": "기기 id", " day": "비트마스킹" "combination": { " name": 조합 이름, " choice1": 1, " choice1Count": 3, " choice2": 2, " choice2Count": 2, " choice3": null, " choice3Count": null, " choice4": null, " choice4Count": null }, " startTime": "10:00:00", " endTime": "18:00:00", " interval": 15 " modeOn": true } }</pre> <p>만약 조합이 바뀐 것이 없다면</p> <pre>{ " id": 스케줄 id " name": "예약 이름", " deviceId": "기기 id", " day": "비트마스킹" "combination": { " id": "조합 id"</pre>	<pre>200, OK 403, Forbidden 현재 실행 중인 스케줄임 409, Conflict 이미 해당 시간대에 등록된 스케줄이 있음 400, Bad Request</pre>	combination 등록 → id가지 고 custom schedule 변경 / 예약 수정하기 <완료> 버튼 누 를 때 호출

# 열 번호	통신방향	유계 발원 상황	≡ 관련 기능	≡ 세부 기능	Method Name	Description	≡ 다중 선택	⊙ HTTP Method	Endpoint URL	Event	Request	Response	Aa 기타 추가 설명
											<pre> }, " startTime ": "10:00:00", " endTime ": "18:00:00", " interval ": 15 " modeOn ": true } </pre>		
34	스프링 → RB	완료	스케줄 모드	라즈베리 파이 스케줄 수정	sendCustomScheduleUpdate		TCP				<p>[구독 경로] "/topic/Schedule/Change/" + id</p> <pre> { "schedule": { " id " : "수정 스케줄id" "combination" : { " choice1 ": 1, " choice1Count ": 3, choice2 ": 2, choice2Count ": 2, choice3 ": null, choice3Count ": null, choice4 ": null, choice4Count ": null, }, " startTime ": "10:00:00", endTime ": "18:00:00", interval ": 15 " modeOn ": true } } } </pre>		제목 없음
35	클라이언트 → 스프링	완료	스케줄 모드	시간 기반 예약 삭제	deleteCustomSchedule	시간 기반 예약 삭제 시 현재 실행 중인 예약은 삭제할 수 없음	HTTP	POST	/v1/custom/delete	verify (token) / customScheduleT.delete (id)	<pre> { " id " : "커스텀 스케줄 id" " deviceid " : 디바이스 id } </pre>	200, Ok 403, Forbidden 현재 실행 중인 스케줄임 400, Bad Request	제목 없음
36	스프링 → RB	완료	스케줄 모드	라즈베리 파이 스케줄 삭제	sendCustomScheduleDelete		TCP				<p>[구독 경로]</p> <p>"/topic/Schedule/Delete" + deviceid</p> <pre> { "scheduleid" : "삭제 스케줄id" } </pre>		제목 없음
37	클라이언트 → 스프링	완료	스케줄 모드	시간 기반 예약 전체 조회	getAllCustomSchedule	예약 모드 클릭 시 기기에 저장된 예약 정보 모두 반환	HTTP	POST	/v1/custom/all	verify (token) / customScheduleT.select (device_id)	<pre> { " deviceid " : "디바이스 id" } </pre>	<pre> { "customSchedules ": [{ " id ": 스케줄 id, name " : "출근 시", day " : "비트마스킹" " isFavorite " : true " combinationid " : 2, combinationName " : "시원", startTime ": "10:00:00", endTime ": "18:00:00", interval ": 15, modeOn ": true }] } </pre>	제목 없음
38	스프링 → RB	완료	스케줄 모드	라즈베리 파이 스케줄 통신	sendDailyCustomSchedules	매일 지정 스케줄 데이터 전송	TCP			customScheduleT.select ,groupBy(device_serial) /scheduler.send(00시)	<p>[구독 경로]</p> <p>"/topic/Schedule/Initial" + deviceid</p> <pre> { "schedules": [{ " id " : "스케줄id" deviceid " : "기기 id" "combination" : { " choice1 ": 1, choice1Count ": 3, choice2 ": 2, choice2Count ": 2, choice3 ": null, choice3Count ": null, choice4 ": null, choice4Count ": null }, " startTime ": "09:00:00", endTime ": "18:00:00", interval ": 15 " modeOn ": true } }] } </pre>		제목 없음
39	클라이언트 → 스프링	완료	자동화 모드	전체 자동화 모드 조회	getAllAutomationModes	기기별 자동화 예약 조회 (on, off 여부)	HTTP	POST	/v1/auto/all	verify (token) / autoScheduleT.select (device_id)	<pre> { " deviceid " : "디바이스 id" } </pre>	<pre> { "autoSchedules ": [{ " id " : "자동화 스케줄 pk" combinationid " : "조합id", subMode ": 0 ~ 2, " </pre>	여기서 interval은 받음

# 열 번호	통신방향	호출발원	관련기능	세부기능	Method Name	Description	다중선택	HTTP Method	Endpoint URL	Event	Request	Response	Aa 기타 추가 설명
												<pre> " type ": 0 ~ 1, " " interval ": 15, " " modeOn ": false/ true } } } </pre>	
74	RB → 스프링	완료	자동화 모드	자동화 모드 정보 요청	sendAutoSchedules		TCP			<p><서브모드/ type> 0 → 단순 탐지 모드 관련 (type - null) 1 → 행동 모드 관련 (type - 운동은 1/ 휴식은 2) 2 → 악취 모드 관련 (type, interval - null)</p>	<p>[destination] "/app/Auto/Schedule/Initial"</p> <pre> { "token": "아이디 jwt" } </pre>	<p>[구독 경로] "/topic/Auto/Schedule/Initial" + device Id</p> <pre> { "schedules": [{ "id " : 2, " combinationid " : 1, " subMode " : 0, " type " : null, " interval ": 15 " modeOn " : true }] } </pre>	제목 없음
41	RB → 스프링 스프링 → RB	완료	자동화 모드	라즈베리 파이 상태 통신	sendCombination	<p>자동화 모드 항기 정보 요청</p> <p>센서를 통해 감지되었고, 모드가 On일 때 호출</p>	TCP				<p>[destination] "/app/DeviceStatus/Sensor"</p> <pre> { "token": "시리얼 번호 jwt", " combinationid " : int } </pre>	<p>[구독 경로] "/topic/Auto/Operation" + 기기Id</p> <pre> { "combination" : { " " choice1 ": 1, " choice1Count ": 3, " choice2 ": 2, " choice2Count ": 2, " choice3 ": null, " choice3Count ": null, " choice4 ": null, " choice4Count ": null, } } } </pre>	제목 없음
45	RB → 스프링 스프링 → RB	완료	자동화 모드	라즈베리 파이 상태 통신	sendCombinationUpdate	<p>조합이 수정될 경우 정보 보내기</p>	TCP				<p>[구독 경로] "/topic/Combination/Change" + 기기Id</p> <pre> { "id " : 스케줄 id (int), " combinationid " : int } </pre>		제목 없음
50	RB → 스프링 스프링 → RB	완료	자동화 모드	라즈베리 파이 상태 통신	sendIntervalUpdate	<p>인터벌이 업데이트 된 경우 정보 보내기</p>	TCP				<p>[구독 경로] "/topic/Interval/Change" + 기기Id</p> <pre> { "id " : 스케줄 id (int), " interval " : int } </pre>		제목 없음
44	스프링 → RB	완료	자동화 모드	라즈베리 파이 상태 통신	sendUpdateModeOn	<p>모드 on/off 가 바뀔 경우 정보 보내기</p>	TCP				<p>[구독 경로] "/topic/Auto/Mode/Change" + 기기Id</p> <pre> { "id " : 스케줄 id, "modeOn" : true/ false, } </pre>		제목 없음
42	클라이언트 → 스프링	완료	자동화 모드	악취 모드 설정 수정	updateDeodorizationMode	<p>combination 을 새로 추가</p>	HTTP	POST	/v1/auto/deodorization/update	<p>verify (token) / combinationT.create / autoScheduleT.update (id)</p>	<p>[조합이 바뀐 경우] { "id " : "자동화 스케줄 pk" "deviceid " : "기기 id" "combination" : { " " choice1 ": 1, " choice1Count ": 3, " choice2 ": 2, " choice2Count ": 2, " choice3 ": null, " choice3Count ": null, " choice4 ": null, " choice4Count ": null, }, " interval ": 15, " intervalChange " : 1 " modeOn ": 0, " modeChange ": 1 (기존 대비 바뀌었냐 안바뀌었냐) } }</p> <hr/> <p>[조합이 안 바뀐 경우] { "id " : "자동화 스케줄 pk" "deviceid " : "기기 id" "combination" : { " "id " : "조합 id" }, " interval ": 15, " intervalChange " : 1 } }</p>	<p>200, OK</p> <p>400, Bad Request</p>	<p>설정부분 combination 은 새로 생성하는 것 / - 용어 탈취-악취 하나로 정하기</p>

# 열 번호	통신방향	상태 발 현 상	관 련 기 능	세부 기능	Method Name	Description	다중 선택	HTTP Method	Endpoint URL	Event	Request	Response	Aa 기타 추가 설명
											<pre>modeOn": 0, " modeChange": 1 (기존 대비 바뀌었나 안바뀌었나) }</pre>		
46	클라이언트 → 스프링	완료	자동화 모드	동작 모드 설정 수정	updateExerciseAndRestMode	스케줄 id, 인터벌 보내주기	HTTP	POST	/v1/auto/exercise/rest/update	verify (token) / autoScheduleT.update (id)	<pre>{ "exerciseSchedule ": { " id": "자동화 스케줄 pk", " deviceid": "기기 id" " interval ": 15 " modeOn": false / true }, "exerciseModeChange": 0, "exerciseIntervalChange": 0, "restSchedule ": { " id": "자동화 스케줄 pk", " deviceid": "기기 id" " interval ": 15 " modeOn": false / true. }, "restModeChange": 0, "restIntervalChange": 0 }</pre>	200, OK 400, Bad Request	제목 없음
51	클라이언트 → 스프링	완료	자동화 모드	단순 탐지 모드 수정	updateDetectionMode	combination을 새로 추가	HTTP	POST	/v1/auto/detection/update	verify (token) / autoScheduleT.update (id)	<pre>[조합이 바뀐 경우] { " id": "자동화 스케줄 pk" " deviceid": "기기 id" "combination": { " choice1": 1, " choice1Count": 3, " choice2": 2, " choice2Count": 2, " choice3": null, " choice3Count": null, " choice4": null, " choice4Count": null, }, " modeOn": 0, " modeChange": 1 (기존 대비 바뀌었나 안바뀌었나) } [조합이 안 바뀐 경우] { " id": "자동화 스케줄 pk" " deviceid": "기기 id" "combination": { " id": "조합 id" }, " modeOn": 0, " modeChange": 1 (기존 대비 바뀌었나 안바뀌었나) }</pre>	200, OK 400, Bad Request	제목 없음
54	스프링 → RB 클라이언트 → 스프링	완료	바로 분사	디플트항 라즈베리 파이 요청	sprayNow sendRemoteOperation	기기 id만 주면 됨	HTTP TCP	POST	/v1/device/spray	verify (token) /device.select (id) / combination.select (default_combination)	<pre>{ " deviceid": "기기 id" } [구독 경로] "/topic/Remote/Operation" + 기기 id { " id": combination id " name": 시원한 향, " choice1": 1, " choice1Count": 3, " choice2": 2, " choice2Count": 2, " choice3": null, " choice3Count": null, " choice4": null, " choice4Count": null }</pre>	200, Ok 400, Bad Request 아이디랑 이름은 활용하지 않지만 기존 메서드 활용을 위해 전달 (또한 공유 되어도 보안 이슈 없는 데이터)	제목 없음
55	클라이언트 → 스프링	완료	찜 기능	조합 찜하기	addCombinationToFavorites	combination id 보내주기	HTTP	POST	/v1/favorite/create	verify (token) / favoriteT.create	<pre>"combinationids": [2, 3, 4] }</pre>	200, Ok 400, Bad Request	제목 없음
56	클라이언트 → 스프링	완료	찜 기능	찜 리스트 전체 조회	getAllFavorites		HTTP	POST	/v1/favorite/all	verify (token) / favoriteT.select (user_id) / combinationT.select (id)	<pre>{ "favorites": [{ " id": "찜 id", "combination": { { " id": "조합 id", " name": 조합 이름, "</pre>	200, OK	토론 (클라에서 백으로 요청시 같이 보냄) → 토론 유효성 → user_id로 찜 불러오기 찜id → combinationid

# 열 번호	통신방향	유 계 발 연 황	≡ 관 련 기 능	≡ 세 부 기 능	Method Name	Description	≡ 다 중 선택	⦿ HTTP Method	≡ Endpoint URL	≡ Event	≡ Request	≡ Response	Aa 기타 추가 설명
												<pre> choice1 ": 1, " choice1Count ": 3, " choice2 ": 2, " choice2Count ": 2, " choice3 ": null, " choice3Count ": null, " choice4 ": null, " choice4Count ": null, } } } } </pre> 400, Bad Request	
57	클라이언트 → 스프링	완료	점 기 능	점 식 제	removeCombinationFromFavorites	점 id 보내주 기	HTTP	POST	/v1/favorite/delete	verify (token) / favoriteT.delete (조합 id) 조합 id와 토큰의 userId가 같아야 삭제해줌	<pre> { " combinationId " : "조합 id" } </pre>	200, Ok 400, Bad Request	제목 없음
58	클라이언트 → 스프링	완료	점 기 능	점 조 합 공 유	shareFavoriteCombination	Open AI 키 를 활 용 해 이 미 지 생 성	HTTP		/v1/favorite/share	verify (token) / makelmg / makeURL	<pre> { " combinationId " : "조합id" } </pre>	200, Ok { "combination" : { " id " : "조합 id", " name ": "시원 함", " choice1 ": 1, " choice1Count ": 3, " choice2 ": 2, " choice2Count ": 2, " choice3 ": null, " choice3Count ": null, " choice4 ": null, " choice4Count ": null, }, "S3Uri" : "이미지 링크", "shareUri" : " http://localhost:5170/favorite/share/read?combinationId=1449355375&imageName=9af866f4-bca0-451f-a025-7b62bd89b461.png } } 400, Bad Request	점 조 합 공 유 하 기
82	클라이언트 → 스프링	완료	점 기 능	공 유 된 점 조 합 조 희	readShareFavorite	쿼리 스트링으 로 정 보 보 내 주 기	HTTP	GET	/v1/favorite/share/read		?combinationId="아이 디"&"imageName"="이미지 네임"	200, Ok { "combination" : { " id " : "조합 id", " name ": "시원 함", " choice1 ": 1, " choice1Count ": 3, " choice2 ": 2, " choice2Count ": 2, " choice3 ": null, " choice3Count ": null, " choice4 ": null, " choice4Count ": null, }, "S3Uri" : "이미지 링크" } 400, Bad Request	제목 없음
59	클라이언트 → 스프링	완료	회 원 정 보 조 희	성 별, 생 년 월 일 조 희	getUserInfo		HTTP	POST	/v1/user/info/get	verify (token) / <u>userT.select</u> (id)		200, OK { " gender " : 0 ~ 2, " birth " : 2000-00-00 }	제목 없음
60	클라이언트 → 스프링	완료	회 원 정 보 수 정	닉네임 수정	updateUserNickname		HTTP	POST	/v1/user/nickname/update	verify (token) / userT.update (id)	<pre> { " nickname " : "수정 닉네임" } </pre>	200, Ok 400, Bad Request	제목 없음
61	클라이언트 → 스프링	완료	회 원 정 보 수 정	성 별, 생 년 월 일 수 정	updateUserInfo		HTTP	POST	/v1/user/info/update	verify (token) / userT.update (id)	<pre> { " gender " : 0 ~ 2, " birth " : 2000-00-00 } </pre>	200, Ok 400, Bad Request	제목 없음
62	클라이언트 → 스프링	완료	회 원 정 보 수 정	회 원 사 진 수 정	updateUserImg		HTTP	POST	/v1/user/img/update	verify (token) / userT.update (id)	<pre> { " imgNum " : 0 ~ 8 } </pre>	200, Ok 400, Bad Request	제목 없음
63	클라이언트 → 스프링	완료	회 원 정 보 수 정	비 밀 번 호 수 정	updateUserPassword		HTTP	POST	/v1/user/password/update	verify (token) / sesseion.get (비밀번호 검증 ok) / userT.update (id)	<pre> { " password " : "수정 비밀번호" } </pre>	200, Ok 401, Unauthorized - 세션 없음 400, Bad Request	제목 없음
64	클라이언트 → 스프링	완료	회 원 탈 퇴	비 밀 번 호 검 증	validatePassword		HTTP	POST	/v1/user/password/verify	verify (token) / <u>userSeculInfoT.select</u> (id) / userT.select (id) / <u>session.save</u> (비밀번호 검증 ok)	<pre> { " password " : "입력 값" } </pre>	200, Ok 401, Unauthorized 400, Bad Request	제목 없음
65	클라이언트 → 스프링	완료	회 원 탈 퇴	회 원 탈 퇴	deleteUserAccount		HTTP	POST	/v1/user/delete	verify (token) / sesseion.get (비밀번호 검증 ok) / userT.delete(id)		200, Ok 401, Unauthorized - 세션 없음 400, Bad Request	제목 없음
78	클라이언트 → 스프링	완료	비 밀	비 밀 번 호 수 정	sendEmailCodeForPassword		HTTP	POST	/v1/user/reset/password/send-code	userT.select(id) / 이메일이 같다면 전송	<pre> { " </pre>	200, Ok	제목 없음

# 열 번호	≡ 통신방향	⚙ 개 발 현 황	≡ 관 련 기 능	≡ 세부 기능	≡ Method Name	≡ Description	≡ 다중 선택	⊖ HTTP Method	≡ Endpoint URL	≡ Event	≡ Request	≡ Response	Aa 기타 추가 설명
			번호 찾기	을 위한 이메일 전송						/session.save(code)	<div>id " : "사용자 id",</div> <div>"</div> <div>email " : "사용자 이메일"</div> <div>}</div>	401, Unauthorized - DB에 등록된 이메일과 다름 400, Bad Request	
79	클라이언트 → 스프링	완료	비밀번호 찾기	비밀번호 수정을 위한 이 메일 코 드 확인	verifyEmailCodeForPassword		HTTP	POST	/v1/user/reset/password/verify-code	session.get(code) / verify.code	<div>{</div> <div>"</div> <div>code " : "이메일 전송 코드"</div> <div>}</div>	200, Ok 401, Unauthorized - 인증 코드가 일치하지 X 400, Bad Request	재목 없음
80	클라이언트 → 스프링	완료	비밀번호 찾기	비밀번호 재설정	resetUserPassword		HTTP	POST	/v1/user/reset/password	userSeculInfoT.update /userT.update	<div>{</div> <div>"</div> <div>id " : "사용자 id"</div> <div>"</div> <div>password " : "입력 비밀번호"</div> <div>}</div>	200, Ok 401, Unauthorized - 세션 없음 400, Bad Request	재목 없음
66	클라이언트 → 스프링	완료	기기 삭제	기기 삭제	deleteDevice		HTTP	POST	/v1/device/delete	verify (token) / deviceT.select (id) / admin_id == 요청 id? / deviceT.delete (id) / closeConnection (serial)	<div>{</div> <div>"</div> <div>id " : "디바이스 id"</div> <div>}</div>	200, Ok 401, Unauthorized - 해당 디바이스의 admin이 아니기에 삭제 권한 없음 404 Not Found - 삭제 요청 준 디바이스 id가 없음 400, Bad Request	기기 삭제할 때 매인기기인 경우 user의 메인 디바이스를 꼭 지정해줘야 함
67	스프링 → RB	완료	기기 삭제	웹소켓 통신 종료	closeConnection		TCP				[구독 경로] "/topic/Connection/Close/" + deviceId <div>{</div> <div>"</div> <div>Close"</div> <div>}</div>		재목 없음