스케일러블 DM 시스템 구현을 위한 Documentation

참고: 기존 Documentation에 있던 채팅 관련 API는 전부 Deprecated.

Chat 스키마:  
텍스트, 스크린샷, 소프트웨어, 운영 체제이(가) 표시된 사진

자동 생성된 설명 isBanned는 차단 여부를 확인하는 용도이다.

Socket에서 의사는 /dm\_doctor 네임스페이스를 활용하고  
유저는 /dm\_user 네임스페이스를 활용한다.

1. RESTful API
   1. User용
      1. /mapp/dm/user/curateScreen
         1. POST
         2. 큐레이팅 화면에서 상담 요청 버튼을 클릭했을 때 반응하기 위한 API이다.
         3. 요청 인자: 
         4. 반환 인자: 채팅방이 기존에 없었으면 새로 만들어서, 기존에 있었다면 기존의 Chat의 \_id 속성을 반환한다.
         5. 오류 코드
            1. 403: 서버 내부 알고리즘 오류
         6. 사용 방법
            1. 큐레이팅 화면에서 상담 요청 버튼을 클릭했을 때, 해당 의사(또는 상담사)와의 채팅창으로 바로 이동하기 위해 사용된다.
      2. /mapp/dm/user/list
         1. GET
         2. 채팅방 리스트를 보기 위한 API이다.
         3. 요청 인자: 없음(토큰으로 본인인증)
         4. 반환 인자: {date, cid, isBanned, doctorId, doctorName, recentChat, unreadChat} 객체의 배열을 반환한다.
         5. 오류 코드
            1. 403: 서버 내부 알고리즘 오류
         6. 사용 방법
            1. 채팅 리스트 화면에서 아직 안읽은 채팅의 개수와 최근 메시지, 채팅방 업데이트 시간 순에 따른 정렬, 의사 이름 표시, 그리고 추후 추가될 프로필 사진불러오기에 대비해서 의사의 \_id 속성을 불러오는 용도로 쓰인다.
      3. /mapp/dm/user/delete/:cid
         1. DELETE
         2. 채팅방을 삭제하기 위한 API이다.
         3. 요청 인자: Chat의 \_id 속성을 파라미터 인자로 넘겨줘야 한다.
         4. 반환 인자: 없음
         5. 오류 코드
            1. 401: 그런 채팅이 없음
            2. 403: 서버 내부 알고리즘 오류
         6. 사용 방법
            1. 채팅을 삭제할 때 사용한다.
      4. /mapp/dm/user/ban
         1. POST
         2. 채팅방을 차단하기 위한 API이다.
         3. 요청 인자: Chat의 \_id 속성을 body에 chatId라는 이름으로 넘겨줘야 한다.  
            
         4. 반환 인자: 없음
         5. 오류 코드
            1. 401: 그런 채팅이 없음
            2. 403: 서버 내부 알고리즘 오류
         6. 사용 방법
            1. 채팅을 차단할 때 사용한다.
      5. /mapp/dm/user/appointmentStatus/:cid
         1. GET
         2. 약속의 수락 여부를 확인하기 위한 API이다.
         3. 요청 인자: Chat의 \_id 속성을 cid 자리에 파라미터로 넘겨줘야 한다.
         4. 반환 인자: Boolean
         5. 오류 코드
            1. 401: 아직 약속이 없음
            2. 403: 서버 내부 알고리즘 오류
         6. 사용 방법
            1. setInterval등으로 비동기적으로 몇 밀리초 마다 서버로 요청을 보내서 약속의 체결 여부를 확인한다.
   2. Doctor용
      1. /mapp/dm/doctor/list
         1. GET
         2. 채팅방 리스트를 보기 위한 API이다.
         3. 요청 인자: 없음(토큰으로 본인인증)
         4. 반환 인자: {date, cid, isBanned, userId, userName, recentChat, unreadChat} 객체의 배열을 반환한다.
         5. 오류 코드
            1. 403: 서버 내부 알고리즘 오류
         6. 사용 방법
            1. 채팅 리스트 화면에서 아직 안읽은 채팅의 개수와 최근 메시지, 채팅방 업데이트 시간 순에 따른 정렬, 의사 이름 표시, 그리고 추후 추가될 프로필 사진불러오기에 대비해서 환자의 \_id 속성을 불러오는 용도로 쓰인다.
      2. /mapp/dm/doctor/delete/:cid
         1. DELETE
         2. 채팅방을 삭제하기 위한 API이다.
         3. 요청 인자: Chat의 \_id 속성을 파라미터 인자로 넘겨줘야 한다.
         4. 반환 인자: 없음
         5. 오류 코드
            1. 401: 그런 채팅이 없음
            2. 403: 서버 내부 알고리즘 오류
         6. 사용 방법
            1. 채팅을 삭제할 때 사용한다.
      3. /mapp/dm/doctor/ban
         1. POST
         2. 채팅방을 차단하기 위한 API이다.
         3. 요청 인자: Chat의 \_id 속성을 body에 chatId라는 이름으로 넘겨줘야 한다.  
            
         4. 반환 인자: 없음
         5. 오류 코드
            1. 401: 그런 채팅이 없음
            2. 403: 서버 내부 알고리즘 오류
         6. 사용 방법
            1. 채팅을 차단할 때 사용한다.
      4. /mapp/dm/user/appointmentStatus/:appid
         1. GET
         2. 약속의 수락 여부를 확인하기 위한 API이다.
         3. 요청 인자: Appointment의 \_id 속성을 appid 자리에 파라미터로 넘겨줘야 한다.
         4. 반환 인자: Boolean
         5. 오류 코드
            1. 401: 그러한 코드를 가진 약속이 없음
            2. 403: 서버 내부 알고리즘 오류
         6. 사용 방법
            1. setInterval등으로 비동기적으로 몇 밀리초 마다 서버로 요청을 보내서 약속의 체결 여부를 확인한다.
2. Socket API
   1. Socket에서 의사는 /dm\_doctor 네임스페이스를 활용하고  
      유저는 /dm\_user 네임스페이스를 활용한다.
   2.   
      Handshake시에 query에 토큰 뿐 만 아니라 chat의 \_id 속성인 방 번호까지 주어야 한다.
   3. 공통 오류
      1. notYourChatorBannedChat 이벤트
         1. 본인이 속한 채팅이 아니거나 밴 당한 채팅일 때 서버측에서 emit 된다.
      2. tokenExpiredError 이벤트
         1. 토큰이 만료되었을 때 서버측에서 emit 된다.
         2. 이 이벤트를 수신했다면 토큰을 리프레시 하면 된다.
   4. User용
      1. SendChat 이벤트: 유저측에서 emit
         1. socket의 data 영역에 plain text로 바로 메시지를 주면 된다.
         2. 이게 끝이다.
      2. chatReceived 이벤트: 서버측에서 emit
         1.  형태의 객체가 유저측에게 반환된다.
         2. 이게 끝이다.
   5. Doctor용
      1. SendChat 이벤트: 의사측에서 emit
         1. socket의 data 영역에 plain text로 바로 메시지를 주면 된다.
         2. 이게 끝이다.
      2. chatReceived 이벤트: 서버측에서 emit
         1.  형태의 객체가 의사측에게 반환된다.
         2. 이게 끝이다.