**How Schedule is defined**

텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

JSON에 익숙하다면 아! availableTime은 (아래는 예시 시각)  
'[[{"from":"09:00","to":"18:00", “memo”: “빈 시간”, “isAvailableTime”: “true”}],[{"from":"09:00","to":"18:00", “memo”: “빈 시간”, “isAvailableTime”: “true”}],[{"from":"09:00","to":"18:00", “memo”: “빈 시간”, “isAvailableTime”: “true”}],[{"from":"09:00","to":"18:00", “memo”: “빈 시간”, “isAvailableTime”: “true”}],[{"from":"09:00","to":"18:00", “memo”: “빈 시간”, “isAvailableTime”: “true”}],[{"from":"09:00","to":"18:00", “memo”: “빈 시간”, “isAvailableTime”: “true”}],[{"from":"09:00","to":"18:00", “memo”: “빈 시간”, “isAvailableTime”: “true”}]]' 의 배열의 형태를 띠겠구나! 라는 것을 인지할 수 있을 것이다.

왜 굳이 저렇게 했냐면, 서버에서 환자와 의사를 매칭시켜 줄 때 가용 시간을 비교하는 냅색 문제 알고리즘으로 해결하기 때문이다.

대략적인 개요를 보자면, availableTime은 7칸짜리 배열이고, ***인덱스 순서대로 인덱스 0은 월요일, 1은 화요일…6은 일요일 같은 순서로 작동된다.***

그리고 그 배열 각 칸에는 또 배열이 존재하는데, 예를 들어 월요일에 내가 9시부터 12시, 14시부터 18시까지가 가능하다 하면

[{"from":"09:00","to":"12:00", “memo”: “빈 시간”, “isAvailableTime”: “true”}, {"from":"14:00","to":"18:00", “memo”: “빈 시간”, “isAvailableTime”: “true”}] 의 형태가 인덱스 0에 저장되겠구나! 라고 인지하면 된다. 참 쉽죠?

그리고 프론트에서 백엔드로 넘겨줄 때는 쟤네를 stringify 해서 넘겨주되,

'[[{"from":"09:00","to":"18:00", “memo”: “빈 시간”, “isAvailableTime”: “true”}],[{"from":"09:00","to":"18:00", “memo”: “빈 시간”, “isAvailableTime”: “true”}],[{"from":"09:00","to":"18:00", “memo”: “빈 시간”, “isAvailableTime”: “true”}],[{"from":"09:00","to":"18:00", “memo”: “빈 시간”, “isAvailableTime”: “true”}],[{"from":"09:00","to":"18:00", “memo”: “빈 시간”, “isAvailableTime”: “true”}],[{"from":"09:00","to":"18:00", “memo”: “빈 시간”, “isAvailableTime”: “true”}],[{"from":"09:00","to":"18:00", “memo”: “빈 시간”, “isAvailableTime”: “true”}]]'   
와 같은 형태는 인식이 안되고  
[[{"from":"09:00","to":"18:00", “memo”: “빈 시간”, “isAvailableTime”: “true”}],[{"from":"09:00","to":"18:00", “memo”: “빈 시간”, “isAvailableTime”: “true”}],[{"from":"09:00","to":"18:00", “memo”: “빈 시간”, “isAvailableTime”: “true”}],[{"from":"09:00","to":"18:00", “memo”: “빈 시간”, “isAvailableTime”: “true”}],[{"from":"09:00","to":"18:00", “memo”: “빈 시간”, “isAvailableTime”: “true”}],[{"from":"09:00","to":"18:00", “memo”: “빈 시간”, “isAvailableTime”: “true”}],[{"from":"09:00","to":"18:00", “memo”: “빈 시간”, “isAvailableTime”: “true”}]]'와 같은 형태는 인식이 됨을 유의하자. 따옴표 하나 차이가 크다. 이것과 관련된 API들은 Postman으로 다 정상 작동 테스트를 했기 때문에, 제대로 작동이 안된다면 프론트에서 잘 파싱해서 보내주는지 확인 해 보자.

또 하나 주의해야 할 점!!!!!! 반드시, 반드시, 반드시 시간 순서대로 와야 한다!!!! 이는 백엔드에서 알고리즘 문제해결을 할 때 진료 가능 시작 시각을 array의 맨 첫 원소의 from으로 계산하기 때문이다.

그런데 여기까지 읽으면 드는 의문. 약속은 서버에서 관리를 해 주냐? 인데, 핵심은 이건 스케줄 “관리용” 이 아니라 “언제가 가능한지” 확인하는 용도이다(그러면 isAvailableTime 속성은 왜 있나요?라고 할 수 있는데, 그건 나중에 서비스 확장을 위함이다). 약속은 appointment 계열 API로 달력화 시킬 수 있으니까 그걸 활용하자. 즉, 환자와 의사 각각이 “가능한 시간대” 가 언제인지 확인하기 위한 용도이다!

minimalAppointmentTime은 뭐냐하면, 환자-의사 큐레이팅 시스템에서 “최소한 이정도 시간이 나야 환자를 예약 받을 수 있다”라는 개념이다. 작성은 분 단위의 수로 한다. 예를 들어, 의사 A가 환자 한 명을 진료 보는데 최소 10분이 필요하다면, 이 값이 10이 되어야 하는 것이다.

이게 어디 쓰이냐 하면, 서버에서 환자와 의사를 매칭시켜 줄 때 쓰는 알고리즘에서 활용되기 때문이다.