



guess_music



~~로그인(유저) → 게임하러가기 클릭 → 방 목록 및 방 생성 화면 → (방 생성 클릭 시 게임 목록 중 하나 선택(추후 비번방)) → 방 목록 선택 시 방 입장하기 버튼 노출 및 클릭 시 해당 게임 대기방으로 페이지 전환 → 방장이 게임 시작 버튼 누를 시 게임 화면으로 전환~~



~~대기방은 해당 게임의 이름을 보여주고 참여자 목록을 보여줌, 그리고 intro에 저장한 글들을 불러 놓음 → 시작버튼을 방장이 누르면 게임 페이지로 참여자 모두 이동~~



~~게임 페이지에서 참여자들은 서로 실시간 채팅을 할 수 있고, 각자 노래를 맞출 수 있으며 자신이 맞춘 노래의 개수를 볼 수 있고 스킵 투표가 가능하다.~~



~~지금 인메모리로 게임 방이랑 User 저장 하긴 하는데, 비효율적 일 것 같음.. 객체 값 업데이트도 좀 문제가 있고..~~



~~어차피 JWT, OAuth로 바뀌어야 하니 일단 그냥 로그아웃해도 세션 유지시키는데.. 뭐.. 그리고 로그인 풀고 다른곳에서 로그인 하려 해도 그전에 이미 잘못로그인하면 맞게해도 로그인안됨..; JWT, OAuth쓰면 이것도 다 가능하게 수정 반드시 필요!~~



지금 vue를 쓰긴 하는데, script단에서 쓰는 거라.. 아무래도 좀 더럽기도 하고.. 여튼 애초에 thymeleaf를 쓰는 부분도 있고 하니 JWT는 다음 프로젝트 때 Vue를 직접 깔아서 따로 또 실행할 때 JWT이용해서 로그인 하고 guessmusic은 혼용하기도 했으니 여기서는 OAuth만 구현하기! 그리고 바로 위 콜아웃 내용 수정해야 함..



~~!개발 끝나면 최우선 실행! Spring Web Security 제대로 찾아보고 공부해서 페이지에 적용하고 OAuth2도 적용해보기~~



~~!추후 개발! 싱글플레이 모드(혼자서 해보고 저장후 종료 등 싱글용 여러 기능을 제공)~~

- 만들어야 할 기능
 - ~~오답 입력 시 현재 곡의 재생이 멈추면 안된다~~
 - ~~지금은 그냥 노래가 멈춰버림..~~
 - js로 해결(버튼,input에 onclick, onkeypress만들어서 해결)
 - fetch사용해서 api통신!
 - ~~첫번째곡을 맞추고 점수를 끊었다가 커도 그대로 맞춘 곡들이 유지됨..~~
 - ~~기능이라 봐도 될 듯~~
 - ~~추후에 세션? 혹은 member db에 게임별? 진행한 곡 저장하기??~~
 - ~~나중에 들어왔을 때 전에 하던 곡부터 계속 진행하는 방식~~
 - ~~이건 쿠키로 구현하면 되고~~
 - ~~홈으로 이동할때 세션에 seq값을 1로 초기화 하면 처음부터 다시 시작?~~
 - ~~세션에 seq값을 저장하고 사이트를 나가면 자동으로 expire처리, 같은 사이트 내에서는 세션으로 기존 정보 유지~~
 - ~~나중에 db에 게임별 진행도를 저장?(새로운 테이블?)~~
 - ~~게임 내 음악의 수를 알아야 하고 그걸 바탕으로 전체 곡수, 현재 진행곡수 등 표시~~

- 게임 종료조건을 만들기 위해 반드시 필요
- 해당 조건 만족시 게임 종료창?띄우기
 - 같은 html에서 hidden으로 바꾸고 결과 창만 띄우고 button 하든에서 reveal하는 방식
 - 그냥 새로운 페이지로 이동하고 결과를 보여주고 게임 선택창으로 이동하는 버튼을 두는 방식
- div자체의 hidden을 조절하며 구현 완료
- 정답 못 맞춘 상태로 skip누르는 경우 seq값 증가시켜서 다음 곡 재생되게끔 수정 필요
- 잔여시간 15초되면 힌트나오게
 - 초성,가수힌트 추가 완료
- 노래 못 맞추고 노래가 제한시간 초과 된 경우
 - 타이머기능 (setInterval)이용해서 해결
- 로그인 조건 걸기, 게임 선택 페이지, 시작 대기 페이지 구현하기
- 게임을 3곡이상으로 2개정도 만들고 페이지 정리, 기능 정리 하기
 - 지금 처럼 단순히 testGame에서 인덱스 1로만 주는게 아니라 뭐 함수 이름을 바꾸든 해서 ..해야 함..
 - 세션에 게임 인덱스도 저장?? → 게임 선택 시 갱신(생성), 게임 종료 시 삭제
- 게임추가, 게임내의 음악 추가,삭제,수정 관리자 기능 추가
 - 게임 추가 시 음악등록하는 기능
 - multipartfile로 파일 저장 완료
 - 근데,나중에 배포할 때는 다른 방식으로 해야 할 지도..?
 - 노래 추가 시 정답으로 인정할 애들을 여러개 적게 하자
 - 그리고 그 정답들에서 중복검사?
 - 스크립트에서 appendchild하고 form에서 정답을 리스트로 바꿈, 그리고 서비스에서 곡 저장은 정답갯수만큼 게임 업데이트는 한 번 수행
 - 근데, 좀 비효율적인 것 같음..
 - 파일 형식이 mp3가 맞는 지 확인 해야 함
 - split으로 간단하게 파싱해서 처리

- 일단 업로드 10메가 까지 가능하게 바꾸긴 했는데, 나중에는 앞 1분만 잘라서 저장하는 방식이 좋을 듯
- (나중에 game1의 노래를 game2의 노래로 변경하는 기능도 추가?)
- (초성,가수 수정하는 기능추가?)
 - 그걸려면, 노래 리스트를 정답 당 1개씩 나열하는게 아니라, 노래 하나당 1개씩 나열로 바뀌야 함(즉, 정답이 여러개면 여러개로 표시해주는 것,..)
- 페이지마다 title설정
- 정답에 가수도 표시
- 테스트 케이스 작성
- 새로그침시 음악 자동 재생이 안됨 → 프론트에서 처리할 부분이니.. 일단 keep
 - 이유는 모르겠지만 일단 해결됨
- ~~css~~간단하게 적용하기
 - 완료
- ~~로그인여부를 메인 페이지 제외하고 계속 확인하기??~~
 - 일단 다 체크 코드 추가하긴 했는데.. 통일하는 방법은 없을까..?
 - js로 만들어놓고 script를 추가한다
- ~~관리자 페이지 링크? or 관리자 아이디로 로그인 하면 바로 이동?~~
 - 관리자로 로그인시 바로 이동하는 거로 구현 하긴 함
 - 홈에 관리자 페이지로 이동 a태그 넣어놓음
- ~~Postgresql로 db변경~~
 - 완료
- ~~EC2 배포 방법 공부~~
 - 공부하고 현재 진행상태 프로젝트를 배포해보자
 - 계속진행
- ~~1-2에서 자꾸 unstoppable이 나와버림..귀신같은..무서움..~~
 - 이유는 파악안됨.. 근데 일단 지울거 다 지우니까 되긴 하는 듯..?
- 스킵하고 카운트 정지시켜야 함(정지 안하면 다음 노래 힌트 나옴)
 - game.js에서 skip song눌리면 clearInterval로 해결

- 세션에서 그 노래 번호 초기화 안됨
 - 세션에 seq에트리뷰트를 제거하는 방식으로 바꾸고, 다시 game에 들어왔을 때 seq이 설정되지 않아서 생겼던 오류
- js에서 redirect한 부분들 전부 변경..
- 노래 저장, 삭제시 경로가 로컬일때는 절대경로 느낌으로 저장해주니 ok됐는데.. 아 걸 ec2에 옮기려니 문제가 발생 함..
 - 우분투 절대경로로 지정해보기 → 실패, 서버에 저장한 파일을 웹에서 로드하면 접근 제한 오류 발생
 - 서버 그냥 메인에 저장하고 거기서 삭제하는 것 까지는 성공함
 - 근데, 오디오를 불러오질 못함.. 웹에서 서버 내부의 파일을 접근할 수는 없기 때문..
 - blob으로 뭐 해보려 했는데 안됨
 - 톰캣 수정 해보자..!
 - 차라리 Blob으로 데이터베이스에 저장??..?? 하고 파일 저장, 삭제 방식 변경 하고 blob으로 파일 전송하고 mp3로 웹에서 띄우는 방법 나중에는 구현해야 할 것 같음..
 - 일단 톰캣 수정으로 해결 함
 - 대신, 이제 노래 저장, 삭제, 그리고 tomcatconfig에 주소들을 서버 배포용이랑 로컬 테스트용이랑 다르게 설정해야 한다..그리고, DB세팅도..맞네.. 로컬에도 똑같은 DB만들어서 유지하자.(내용은 다르겠지만 ㅇㅈㅇ)
- upload페이지에서 finish uploading 명칭을 뭐 그냥 업로드 끝내기 이런식으로 변경하자.
- modifygame →
 - 돌아가기 버튼 추가
 - 추가로 관리자 페이지 메인에서도 홈으로 돌아가기 만들었음
- 노래나 게임 목록 여러개인 경우 스크롤 필요
 - overflow설정함
- modifygame → 노래 정답 수정 말고 정답 추가도 있어야 함.. 그리고 정답 삭제도 있어야 함
 - 정답추가,정답 삭제 기능 생성 완료

- 인텔리제이 커밋하는거 범위 설정하는 방법 찾아보기
- 리스트 불러올 때 노래 번호(seq)순으로 정렬해야 함
 - Answers에 Comparable 구현해서 해결 완료 (for Collections.sort())
- 노래 업로드시 정답 추가했다가 취소하는 버튼도 필요할 듯
 - element.remove()로 구현 완료
- 화면 크기에 따라 내용이 덜보이거나 함.. width(가로) 스크롤도 넣던가 뭐 css수정을 해야 할 듯..
- input도 기본 사이즈 좀 더 길게하고? 더 늘어나면 늘어난 만큼 증가 하게끔..
- 게임 제목 변경 기능 추가
- 로그인 실패 등 실패케이스도 테스트로 만들어 놓자.
- navbar..?
 - 로그인시 로그아웃버튼,홈버튼,관리자이동버튼? 정도
 - 비로그인시 로그인버튼 홈버튼 정도..
 - 추가하기에 모든 html에 다 추가해야 한다는 점, 로그인,아웃마다 구분해서 넣어야 할 텐데 이게 다 수작업이면 너무 하드 코딩이고 시간낭비라 생각 됨.. → 나중에 뭐 react나 vue로 프로젝트 프론트 만들면 그때 import해서 사용..
- 로그인시 이미 로그인되어 있는 계정이면 그거 핸들링 해야 함(중복로그인은 안되니까)
 - 스프링 웹 시큐리티로 중복로그인 제한 걸면 됨
- 멀티플레이
 - STOMP랑 vuejs로 채팅 기능 개발 완료
 - 웹소켓에 대해 공부해보자
 - waitingRoom.js부분 topic/ app/ 부분 부터 수정 하기
 - 방 목록은 서버 메모리에 저장하는 것으로 결정(휘발성이 맞으니까..)
 - 방 생성 클릭 시 게임 리스트 불러오고 그 옆에 생성하기 클릭 시 해당 방이 생성되고 방장이 선택한 사람으로 설정되며 방장은 바로 대기방으로 이동한다.
 - 멀티플레이를 하는 경우 여러 사람들이 한 방에 존재할 것이고,, 그렇다면 해당 방은 방 id, 방장 id를 가지게 됨

- ~~즉, 한 대기방에서 유저들의 세션을 관리할 수 있어야 하고 게임 시작버튼을 방장이 누르면 해당 대기방에 있던 유저들을 모두 게임 방으로 이동시켜야 함~~
- ~~클론해온 git 코드 살펴보고 그걸 바탕으로 멀티 플레이어시 세션을 어떻게 만지작 해야 하는 지 파악 해보자..(h2db커야함)~~
- 실시간 채팅
- 방 만들기, 방 찾기 뭐 이런 느낌으로 누가 만든 방이 있으면 그 방에 들어가고
 - 방장에게만 시작버튼 노출
 - 클릭 시 intro부분 텍스트 감추고 노래 틀어야 함
- 방장이 시작하면 방에 있는 모든이들 또한 시작해야 함
 - 메세지 객체 타입에 START추가하고 start면 구독자 전부 startgame호출하게 변경



여기까지 Ver0.1

- ~~세션목록을 받아오려면 non-anonymous 해야 한다 즉, spring-security를 활성화 해야 함..~~
 - ~~그리고 애는 모든 접속 시도를 인터셉트하고 로그인을 시키기에 내가 원하는 페이지에서 로그인하고 로그인 정보를 내 Db와 연동하기 위해서 config를 수정해야 함..~~
 - ~~근데 옛날 자료들은 이미 deprecated고 새로운 자료들은 뭔 말인지 잘 모르겠당 ㅜㅜ~~
 - 로컬 적용 완료
 - 배포 적용 테스트 필요
 - 방에 들어와 있는 유저의 수, 유저의 정보들을 공통의 보드에 표시할 수 있어야 한다.
 - 유저별 점수
 - 스킵 기준 선정
 - 방 폭파(유저0명)
 - 일단 정답 판별 기능 , 방 상태(대기,진행,종료) 구현 하고 입장 제한 걸고 하자.

- 종료는 방에 있는 인원을 알고 그게 0이되거나 게임이 끝나게 되면 설정
- 새로고침 시 다시 처음 인트로화면으로 로드됨 → 새로고침시 뭐 튕기게끔 하던지..
 - 노래 진행된 후 새로고침 박으면 해당 방에 정보가 이미 스킵된 노래부터 진행되니까 이것도 핸들링
 - 버튼으로 새로고침하는 경우 제외하고 전부 막음, 그리고 게임 페이지 이탈 시 경고 창 띄워주고 Home버튼이나 게임 종료 후 자동이동 하는 경우 경고 삭제
- 방에서 나가는 경우 핸들링 필요
 - 시작 전
 - 방장이 나가면 방장 변경
 - 유저가 나가면 참가자 수 감소 뭐 이 정도?
 - 시작 후
 - 방장개념 삭제
 - 스킵 유저 제한 변경
- 지금은 서버측에서 유저의 게임, 노래번호를 컨트롤해서 다수 유저가 들어오는 경우 1번 게임에서 2번노래 듣는 사람 때문에 2번 게임에서 1번노래듣는 사람의 힌트가 제대로 안나오기도 하고 반대로 1번에서 스킵이 안되기도 함(표시할 목록이 없어지기 때문)
 - 즉, gameId, seq 세션으로 컨트롤 해야 할 듯
 - 방 도메인에 gameId, seq, roomStatus 등 추가해서 관리
 - 대기실 입장 전에 방 목록에서 진행 여부 보고 시작 전이면 들어갈 수 있고 진행 중이면 들어갈 수 없고 종료되면 방 삭제
- 스킵투표기능
 - 1명이면 1명스킵시 스킵, 2,3명이면 2명 스킵시, 4~5인은 3인 스킵시 6~7인은 4인 스킵, 8~9 5인..
 - skipVote랑 skipSong이랑 timer에 노래 넘기는 부분 변경 필요
 - 구현 완료



여기까지 ver0.2

- 로그인 실패하는 경우 ~~핸들링 하는 custom handler~~필요함
 - 아이디 혹은 비밀번호 오류
 - 이미 로그인 중인 아이디 오류
 - 위 두 오류에 대해서 각각 alert를 보내든 될 하든 해주는게 좋을 듯..
- OAuth 공부해서 소셜 로그인 기능 추가
 - 아이디,비밀번호 찾기 기능 넣기
 - 그럴려면 이메일 인증이 필요함.. pass
- 전체적으로 리팩토링 필요..
- NavBar 만들기
 - 공통 : 홈
 - 익명 유저 : 로그인
 - 유저 : 로그아웃, 게임 방 입장
 - 관리자 : 로그아웃, 관리자 페이지 입장
- 대기실 새로고침 버튼 만들기



여기까지 ver 0.3

-
- !!알게 된 것!!
 - spring에서 세션을 저장하는 방법으로는
 - servlet context : 서버 시작시 생성, 서버 중지 시 제거
 - http session : client최초 접속시 생성, client 접속 종료 시 제거
 - 로그인,로그아웃, 장바구니 등등
 - httpServletRequest : client요청 시 생성, server응답 시 제거
 - 요청 재지정 : RequestDispatcher , HttpServletResponse
sendRedirect(URL)

- 쿠키는 Cookie객체로 간단히 생성 가능
 - setMaxAge(int) : 쿠키 지속 시간, 0이면 소멸, 1:1초
 - HttpServletResponse객체에 addCookie메소드로 쿠키 등록 가능
 - @CookieValue라는 어노테이션주고 params로 받으면 서블릿요청으로 쿠키 읽을 필요 없이 스프링 컨트롤러에서 파라미터로 바로 전달 받을 수 있음
 - 대신, @Cookie(value=전달받을인자,required=false)이런식으로 써야 함, 즉 required가 false가 아니면 IllegalStateException발생 가능, 그러므로 defaultValue나 required=false해주는 것이 좋다
- fetch()통신 !
 - 기존에 react와 node간에 사용하던 axios는 spring 내에서 사용하는 방법을 잘 모르겠음..
 - fetch도 거의 같은 형태인 것 같다..!
 - response를 json으로 처리하면 api에서 json형태(객체 형태)로 온 애들을 처리하고, text로 처리하면 String으로 온 애를 처리함
- gradle vs maven
 - maven이 오래된 친구고 애는 기본 의존성 설정파일이 pom.xml임
 - gradle로 변환하려면 그냥 ./gradlew init해주면 알아서 메이븐에서 그레이들로 변환함
 - 지금 단계에서는 그레이들을 안 쓸 이유가 없다!
- application.yml vs application.properties
 - 애는 버전관리랑은 상관없음
 - <https://recordsoflife.tistory.com/434> 참고
 - yml로 이름만 바꿔주면 알아서 바뀜
 - yml이 계층적 구조라 보기 편함
- EC2
 - nohup으로 서버 돌리는게 편ㅇ한듯? 로그도 남고..
 - 계속해서 사용하는 게 아니라면 freetier 한도 잘 보고 인스턴스 중지하고 rds도 꺼주는게 좋다
 - 특히, 탄력적 ip는 인스턴스에 연결되어 있지 않거나 연결된 인스턴스가 중지되어 있으면 요금이 발생하는 신기한 놈 이라 애도 한동안 안 쓸거면 같이 중지하

고 릴리스해서 풀어줘야 한다!

- 웹소켓

웹소켓 개념

stomp

- <https://velog.io/@rainbowweb/스프링부트-STOMP로-채팅방-구현하기>
- <https://brunch.co.kr/@springboot/695>

- Vue.js

- 스크립트 단에서 그냥 `var vm=new Vue({ el : , data : {}, ...}, {}, ...)` 이런 식으로 사용 가능
- localStorage라는 기능을 제공 쿠키보다 용량이 많은 장점이 있음
 - 세션처럼 그냥 `localStorage.setItem`으로 저장하고 `getItem`으로 값을 다시 얻을 수 있다!

- Spring Web Security

- <https://lotuus.tistory.com/78>
- 커스텀 UserDetails를 사용하면서 세션으로 max설정해서 중복로그인 방지하려면 반드시 같은 유저임이 체크 될수 있도록 equals , hashCode 메소드에 대한 처리를 고려 해야 한다.
- 중복 로그인 방지 하려고 `maxSessionPreventslogin true`로 주고 새롭게 로그인 하려는 사람을 막으니까 꽤나 오류가 많이 나옴
 - <https://www.infllearn.com/questions/40072/동시-세션-제어-동일-브라우저에서-로그아웃이-정책-미적용>
- Login Failure오류에서 UsernameNotFoundException은 아이디 오류 인데, Spring Security에서 아이디가 잘못된걸 알려주면 보안에 문제가 있을 것이라 판단해 기본적으로 해당 오류를 숨긴다. 즉, 비밀번호 오류도 아이디 오류도 같은 오류인 것 처럼 보이게 함
 - AuthenticationProvider 생성 시 `setHideUserNotFoundExceptions`를 false로 바꾼다.

- 공부할 사항

- entitymanager의 기능, 엔티티의 생명주기

- 프론트와 api통신할 때 param으로 소통하는 것 말고 body로 소통하는 방법(당연히 json으로 소통하는게 훨씬 대다수의 상황임.,.)
- EC2
 - <https://bcp0109.tistory.com/356>
- OAuth
- JWT 토큰 인증
 - <https://stir.tistory.com/266>
 - 여러명이 개발하는 환경에서 Thymeleaf + REST API 방식을 이용한다면 페이지 이동과 API 정확하게 분리해야해서 아키텍처 설계가 또 필요하다는 단점이 존재합니다.
 - 그래서 SSR 방식의 엔진을 사용하는 것 보다는 CSR(Client Side Rendering) 방식의 React, Vue를 사용하는 것이 JWT를 구축하기에는 더 깔끔합니다.
 - 또한 Thymeleaf로 REST API와 JWT를 구현하면 Thymeleaf 영역에서 Security 태그 기능을 이용할 수 없습니다.
- log4j2
 - 자바를 위한 로그
 - starter web안에 기본으로 logback이란 로깅 프레임워크가 있어서 애를 비활성화 시키고 추가해야 한다.
- 웹 소켓
 - 클라이언트와 서버가 tcp 핸드셰이크로 통신하는 것?
 - 나중에 정리 필요
- <https://tecoble.techcourse.co.kr/post/2020-09-29-spring-properties-binding/>
- 한계점
 - 정답을 정말 동시에 입력해서 api에서 처리되고 block되기 전에 들어오면 오류가 나지 않을까..?
 - 인 메모리 시스템 이라.. 불안정함
 -