

프론트엔드 엔지니어 기술 과제 요구 사항

본 문서는 기술 과제에서 제출해야 할 페이지 구성 등을 설명합니다. 아래의 요구 사항에 만족하는 플레이리스트 페이지 구현 및 음악재생 플레이어를 구현해주세요.

과제 진행 안내

제출

- 제한 시간: **24시간**
- 제출 방법: **zip** 파일로 압축 후 메일로 제출
 - 제출 시, `node_modules`와 같이 설치 가능한 의존성 파일들은 제거 후에 제출해주세요.
 - 빌드된 결과물이 아닌 작업 가능한 상태의 코드를 제출을 제출해주세요.
 - 구현이 모두 완료되지 않아도 과제 제출이 가능합니다.
 - 과제 불합격 시에도 제출하신 과제에 대한 코드 리뷰를 전달드립니다.

프로젝트 구성 및 제한사항

- 프로젝트 구성
 - TypeScript 기반으로 구성된 간단한 CRA 입니다.
- 제한사항
 - 필수 사용 기술
 - TypeScript
 - 상태 라이브러리 또는 `fetching hooks library` (라이브러리 선택은 자유)
 - 모든 디자인은 자유롭게 구성 가능합니다.
 - CSS, SCSS, CSS-in-JS 등 스타일링 도구에 제한은 없습니다.
 - 단 **UI 라이브러리 (MUI, Ant Design 등)**는 사용할 수 없습니다.
 - 필요한 라이브러리는 자유롭게 설치 가능합니다.
 - 단 오디오 조작은 외부 라이브러리 (`react-audio-player` 등)를 사용할 수 없습니다.
 - Audio element 혹은 Web Audio API를 사용해야 합니다.

과제 요구사항

개요

음악을 들을 수 있는 플레이리스트 페이지를 구성해주세요. 플레이리스트 페이지는 음원 목록과 음원을 재생할 수 있는 오디오 플레이어로 구성됩니다.

구현 요구 사항

요구 사항은 필수/선택 사항으로 나뉩니다. 선택 구현 사항 구현시 가점이 부여됩니다. 선택 구현 사항을 구현하지 않으시더라도 불이익은 없습니다.

API 통신

- 필수 구현 사항
 - API 오류 발생시 UI 처리
 - 디자인은 자유롭게 구성 가능합니다.
- 선택 구현 사항
 - API 응답 대기 상태를 나타내는 로딩 UI
 - 로딩 중에는 다른 버튼을 클릭 할 수 없도록 디자인 혹은 기능이 구성되어야 합니다.

플레이리스트 페이지

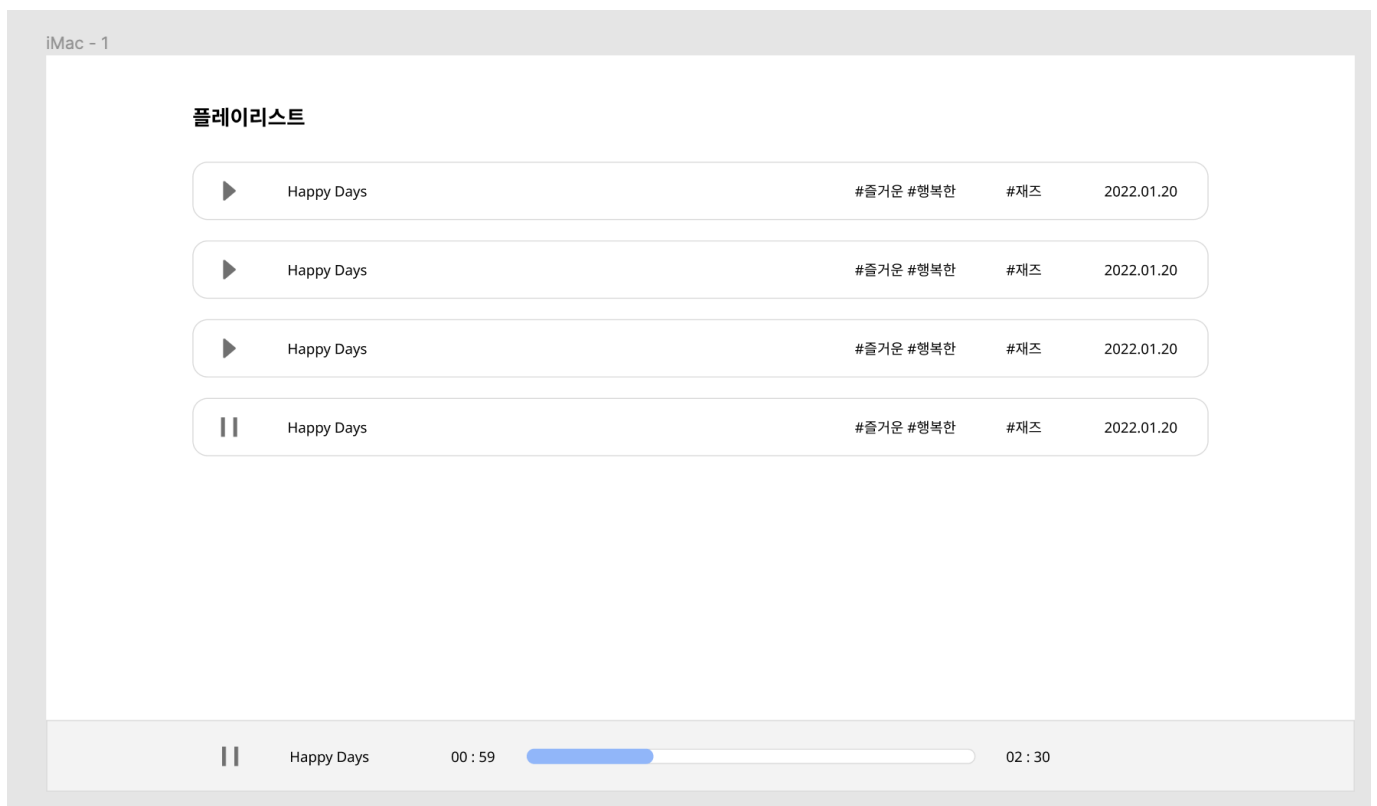
- 필수 구현 사항
 - 제공되는 API에서 음원 리스트를 받아 화면에 출력
 - 각 음원 항목은 아래 요구 사항을 만족해야 합니다.
 - 정보 표시
 - 음원 제목
 - 분위기
 - 장르
 - 발매일
 - 재생 버튼 구현
 - 음원이 재생중이면 일시 정지 아이콘, 재생중이 아니라면 재생 아이콘을 표시
 - 재생버튼 및 일시정지 아이콘은 `src/assets/images` 경로를 확인해주세요.
- 선택 구현 사항
 - 제공되는 데이터를 최신 발매일 순으로 정렬
 - 발매일을 `yyyy.MM.dd` 의 형태로 출력

오디오 플레이어

- 필수 구현 사항
 - 오디오 플레이어 렌더링 규칙
 - 페이지 최초 렌더링시에는 오디오 플레이어는 화면에 미표시
 - 음원에서 재생버튼을 클릭하면 오디오 플레이어가 화면에 나타나며 클릭한 음원을 재생
 - 오디오 플레이어 표시 정보
 - 현재 음원 (재생/일시정지)의 정보
 - 음원 제목

- 현재 재생 시간
- 전체 재생 시간 (API를 통해 받아온 정보에서 추출)
- 재생바
- 정보 표시 규칙
 - 현재 재생 시간, 전체 재생 시간은 00:00 의 형태로 분과 초를 표시
 - 음원이 재생 중이면 현재 재생 시간도 함께 업데이트
 - 재생바는 음원이 재생될때마다 색이 채워짐
 - 재생바 클릭시 클릭한 위치 부터 음원이 재생
- 재생 규칙
 - 음원 목록에서 다른 음원의 재생버튼을 클릭하면 해당 음원을 처음부터 재생
 - 일시정지 버튼을 클릭하면 음원 재생 정지. 다른 음원을 클릭하지 않고 동일한 음원의 재생 버튼을 다시 클릭하면 현재 재생 시간부터 이어서 음원을 재생
- 선택 구현 사항
 - 크롬, 사파리 등 브라우저 종류와 관계 없이 재생이 가능하도록 구현

(예시 화면)



API

- 함께 제공되는 Mock 서버를 사용해주세요.
- Mock 서버 URL: <http://localhost:8000>

음원 목록 조회

- 엔드포인트 : `GET /musics`

응답 스키마

```
interface MusicList {
  total: number;
  items: {
    id: string;
    title: string;
    moods: string[];
    genre: string;
    public_date: string;
  }[];
}
```

음원 재생 URL 조회

- 엔드포인트: `GET /musics/:musicId`
- 요청 후 2초 후에 응답이 리턴됩니다.

응답 스키마

```
interface MusicURL {
  url: string;
}
```

기타

- 필수 사항
 - 과제 해결 전략이나 구성 등을 `README.md` 에 명시해 주세요.
- 선택 사항
 - 필요한 라이브러리가 있는 경우 자유롭게 설치하실 수 있습니다. (오디오 조작 라이브러리 제외)
 - 예) styled-component, sass, tailwind, redux, recoil, react-query 등
 - 테스트 코드 작성

유의 사항

Mock 서버 관련 코드가 삭제되면 API 호출이 불가능합니다. 작업 혹은 제출 시 관련 폴더 및 파일 (mockServer, index.tsx의 createMockServer)의 삭제에 유의해주세요.

평가 기준

정상 동작 여부 및 필수 요구사항이 모두 만족했는지 평가를 진행합니다. 이후 확장성, 코드 스타일, 재사용성 등 코드 전반을 평가합니다. 코드 작성의 경우, 코드의 확장성 및 재사용성은 중요한 기준으로 평가됩니다.

- 정상 동작 여부
 - 결과물의 모든 기능이 정상 동작하는지
 - 모든 필수 요구사항이 구현되었는지
- 코드 작성
 - 확장성 및 재사용성이 고려되었는지
 - 최적화가 고려되었는지
 - 코드의 가독성이 높은지 (e.g. 변수 및 함수명이 명확한 네이밍으로 작성되어 있는지)
 - 일관된 코드 스타일이 적용되었는지

저작권

해당 문제에 대한 저작권은 주식회사 포자랩스(이하 '회사')에게 있으며 수령자는 오로지 채용을 위한 목적으로만 해당 문제를 활용할 수 있습니다. 사유를 불문하고 해당 문제의 전부 또는 일부를 공개, 게재, 배포, 제3자에게 제공 하는 등의 일체의 누설 행위에 대해서는 저작권법에 의한 민.형사상의 책임을 질 수 있습니다. 아울러 이러한 누설 금지 행위에는 문제의 문구를 변형하여 그 취지를 알 수 있도록 하는 경우 포함됩니다.