1. 把当前歌曲列表的信息存储到本地(提高获取效率 )。当点击每一首歌，会传一个索引，通过索引取得对应的元素，再把元素的信息显示到界面上

* 在pages的musiclist中wx.setstorgesync()把musiclist数组放到缓存中
* 在component的musiclist里进到播放页面的方法中，传入当前这首歌的index(z在wxml中data-index来绑定当前index)，再把这index作为参数拼接到player路由的后面
* 去到pages的player里, options.index就是当前这首歌的index
* 从缓存中拿到数组，并传入拿到的index，从而拿到当前这首歌的对象
* 再把对象中的图片等等渲染的页面上

1. 去阿里矢量图库中拿到上一首，下一首，播发，暂停的矢量图并渲染
2. 音乐控制面板功能(1)

* 拿到每首歌音频的url

1. 去music云函数，定义musicURL路由
2. 路由中拼接url，注意要拿到当前player中options里的musicid
3. 去到前端player page中，onload中调用云函数，并把name和data传过去

* 用getBackgroundAudioManager()获取全局唯一的背景音频管理器(backgroundAudioManager)
* 拿到backgroundAudioManager的音频和标题，音频通过musicURL的res拿到，剩下的属性都从当前music对象中拿到
* 到全局json中配置requiredBackgroundModes:[“audio”], 代表可以在小程序之外也听到播放歌曲
* 再用backgroundAudioManager拿到封面图片，歌手名称，专辑名称
* 设个布尔值，来判定当前是否播放，默认为不播放，音乐载入好后也不播放，要当backgroundAudioManager各项拿到后播放，
* 在样式中，如果正在播放，显示暂停矢量图。如果没在播放，显示播放矢量图

(2)

- 给歌曲图片添加旋转效果(rotation)， 再给上面的指针添加下落效果(当音乐播放时才执行，旋转到0度)

- 播放暂停按钮: backgroundAudioManager.pause()暂停backgroundAudioManager.play()为播放

- 如果暂停，图片停止旋转(animation-play-state:paused)

- 上一首下一首:

- 绑定事件，上一首index++, 下一首index—

- 如果在第一首，上一首就是最后一首

- 如果在最后一首，下一首就回到第一首

- 拿到合适id再重新loadmusic

- 载入下一首之前，把上一首停掉

进度条

1. 页面布局

* 建立并调用progress-bar组件
* 显示当前时间和总时间
* 在当前时间和总时间之间，加入moveable-area(长条)和moveable-view(滚动的圈)
* 设置moveable-view的x位置和当前进度的百分比
* 获取当前整个进度条的宽度. 用this.createSelectorQuery拿到query, 再query.select(要选择的class).boundingClientReact(), query.exec(rect=>{}),拿到对应class的数组
* 把拿到数组的第一个和第二个的width分别赋值给movableAreaWidth和movableViewWidth
* 把backgroundAudioManager绑定上事件
* **在onCanplay()方法中(音乐加载完毕，可以播放了)**拿到backgroundAudioManager.duration(总时长)（小程序自带bug: 这里有几率会undefinded）
* 把duration转换成 分钟:秒的格式(如果分钟或秒<10, 要在前面加上0)
* 把处理好的分秒时间赋值给Time里的总时间

1. 进度条与时间联动

* 在onTimeUpdate()中更新当前时间。**用backgroundAudioManager.currentTime拿到当前时间**，并用之前的方法转换为分钟:秒格式
* **\*优化让onTimeUpdate()每秒更新一次即可, (之前是一秒4次)**

设一个当前时间默认-1，每次方法执行把它赋值为currentTime, 如果当前时间和currentTime的秒数不相等，才会做更新操作

* 拿到进度条中圆圈的x位置(和总宽度和它本身的宽度，当前时间有关)
* 拿到progress(白色已播放)进度条占比例的份数
* 把处理好的分秒时间赋值给当前时间

1. 拖拽进度条

* movableView绑定两个事件，一个是自己写的onchange函数，另一个是自带的bindtouchend=”onTouchEnd”(代表滑动后手指离开屏幕)
* Onchange函数中拿到事件源，只有当事件源的source为 touch 的时候才执行后面的赋值
* 分别把进度条百分比和滑动圈的位置进行赋值，这些值都和刚拿到的事件源的x有关系
* 到onTouchEnd()方法中，再用setData赋予刚刚这两个参数值，并且转换分钟:秒类型再赋值给当前时间，(backgroundAudioManage.currentTime拿到时间)
* backgroundAudioManage.seek(当前在多少秒)获取当前音乐文件的时间(拖拽到哪就播放到哪),

1. 自动播放下一首

* 当前这首结束时，会触发onEnded(), 在里面triggerEvent()
* 在page的进度条组件中bind里面定义的event， 给它绑定上之前写的onNext方法

1. 进度条性能优化

* 设一个值isMovig判断进度条有没有被拖动，默认是false，但onChange时就是true, onTouchEnd又是false, onPlay里也是false
* 在onTimeUpdate里，isMoving对当前取反的情况下，写入相应的逻辑。

歌词:

1. Lyric组件:

* 封面和它的显示是切换的，默认是封面显示
* 云函数中获取lyric接口的数据，并在前端被调用

1. 歌词数据解析:

* 在观察器中得到歌词
* 定义歌方法，把歌词传进去，拿 存着歌词每一行的数组
* 循环遍历每一行，正则表达式拿到时间
* 把时间和歌词分隔开，拿到每一行的歌词
* 用正则表达式把时间变成数组， 再把时间单位转换成秒
* 把每一行的时间和歌词作为一个对象push到一个数组，再把这个数组setdata到data的数组
* 在前端循环渲染data数组

1. 歌词与时间联动
2. 相应的歌词行高亮显示

* 在progress-bar的onTimeUpdate中用this.trigger()把一个自定义事件和currentTime抛出去
* 在player page中，定义timeUpdate方法，把事件传进来，用this.selectComponent(lyric的类名)来获取lyric，在调用lyric中相关方法，并传入event.detail.currentTime作为参数, 这样在lyric组件中就可以获取到currentTime
* 在这个方法中，拿到存放歌词的数组，并遍历，设一个目前高亮选中的一行歌词索引nowIndex：

1. 如果**currentTime比当前索引的时间短或者一样**，那nowIndex就是当前索引-1, 再去到wxml高亮渲染当前行的歌词(如果index和nowIndex相等，就高亮选择该行歌词)
2. 高亮显示后break这个循环

移动歌词

* 给需要滚动的地方加上 scroll-y scroll-top={{绑定的data}}
* 去到循环中，给scrollTop加上高度（就是比当前少一个的索引数）\* 每一行的高度
* 如何获取不同屏幕下每一行的高度
* 在自定义生命周期函数中，wx.getSystemInfo()，里面success后拿到的res.screenWidth / 750就是当前屏幕下1rpx的大小pixel值，让它乘以当前每一行的rpx数就把每一行高度转换成了px
* \*问题修复: 有些歌词的最后一句的时间距离整个歌曲结束还有一段时间

这样导致**currentTime比当前索引的时间短或者一样**的条件不成立，所以歌词高亮选中没有到最后一句

修复：设个条件，如果当前时间大于歌词最后一句的时间，那么就把nowIndex设为-1, 并把歌词滚动到最后一句的高度即可

* 去到观察器中，条件判断，如果当前的歌词内容为暂无歌词，setData手动构造歌词数组中的对象，传入设置的暂无歌词和0为时间，再把nowIndex设为-1。 有歌词的话，再解析对应的歌词

Bug修复:

1. 点击下一首或上一首，再返回，当前高亮还是之前那首

* 在app.js中用this.globalData设置全局变量playingMusicId
* 设置set和get方法，set是把playmusicid设置为之前的musicid
* 在player page中把musicid传进set方法(设为全局)
* 去到musiclist组件中，声明页面生命周期pageLifetimes的show方法，代表此页面展示时执行。 把这个组件中的playingid设为全局get的playingmusicid(注意要把playingmusicid的类型转换为数值，因为它本身是string类型，而playingid是数值类型)。实际上就相当于把playingId和当前播放的musicId进行绑定.

1. 让控制面板的暂停和播放也能控制当前歌曲的暂停播放按钮，以及图片旋转与否的界面

* 控制面板的暂停与否也能调用progress-bar组件中的onplay和onpause方法，所以在这两个方法中triggerEvent到父页面(player page)
* 在页面中调用该组件的位置bind播放与暂停事件，并在这两个事件中处理isPlaying为true还是false