

说明文档

实现功能

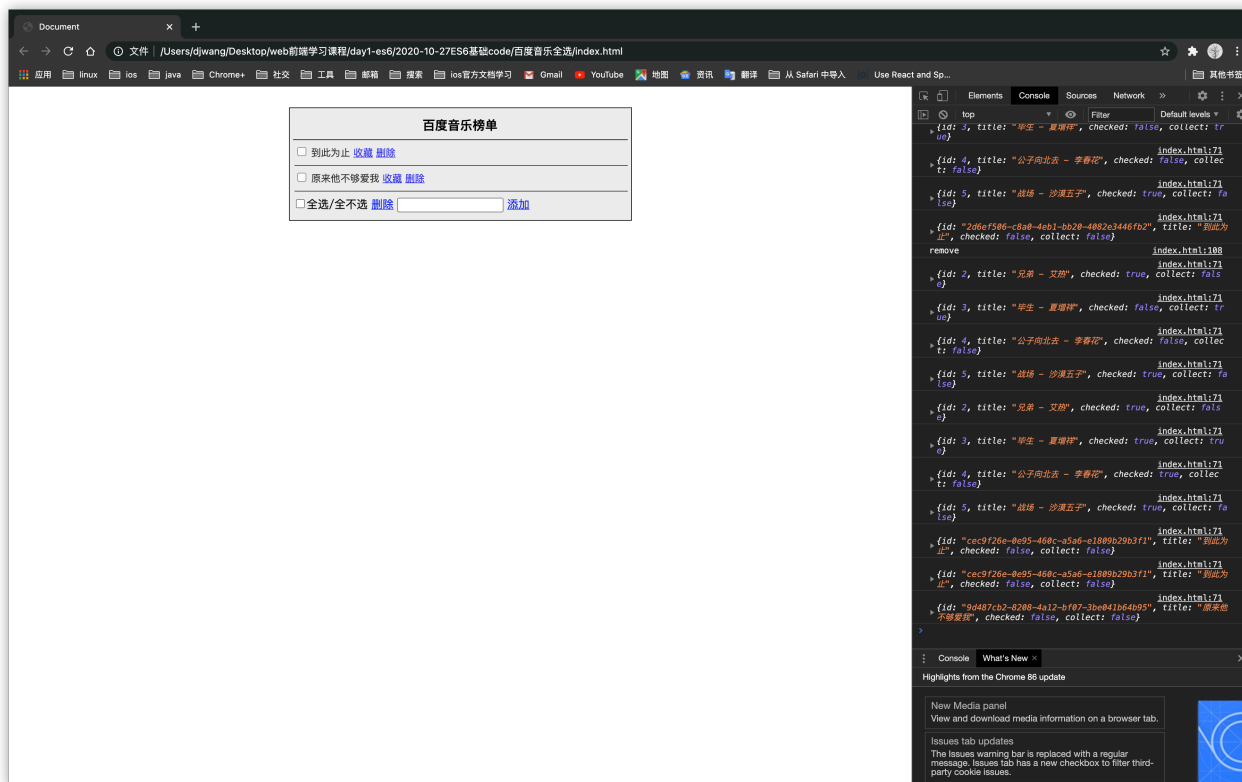
1. 点击“添加”或键盘回车时，若输入框中存在输入，则将歌曲添加到歌曲列表中显示
2. 前端生成uuid作为歌曲数据的id（该步骤实际上不是前端该考虑的，后端插入数据到数据库时会生成id）

修复功能：

1. 修复全选删除的逻辑错误

运行条件

1. 在chrome中测试运行正常



核心代码截图

```
addOtherEvent(); // 注册列表之外的其他事件
function addOtherEvent() {
  // 注册全选事件;
  document.querySelector("#checkAll").onclick = function () {
    data.forEach((item) => {
      item.checked = this.checked;
    });
    renderData(data);
  };
}
```

```

};

// 删除指定;
document.querySelector("#remove").onclick = function () {
  data = data.filter((item) => item.checked === false);
  renderData(data);
};

// 作业：在老师代码基础上实现添加功能。（先操作数据--》视图）；
// 1. 点击添加后添加新歌
document.querySelector("#add").onclick = addNewSong;
// 2. 输入后回车也可添加新歌
document
  .querySelector("#newInfo")
  .addEventListener("keydown", (e) => {
    if (e.key == "Enter") {
      addNewSong();
    }
  });
}

function addNewSong() {
  // 获取uuid
  function getUuid() {
    var temp_url = URL.createObjectURL(new Blob());
    var uuid = temp_url.toString();
    URL.revokeObjectURL(temp_url);
    return uuid.substr(uuid.lastIndexOf("/") + 1);
  }
  let inputElem = document.querySelector("#newInfo");
  let songName = inputElem.value;
  if (songName !== undefined && songName !== "") {
    let song = {
      id: getUuid(),
      title: songName,
      checked: false,
      collect: false,
    };
    data.push(song);
    renderData(data);
    // 添加成功后完后清空输入框
    inputElem.value = "";
  }
}

```

