Week 5：集成KOA后端框架

在很长一段时间里，基于Node.js的后端开发都是基于Express这个框架上的。Express是在Node原生的HTTP进行再封装库。

koa1

随着新版Node.js开始支持ES6，Express的团队又基于ES6的generator重新编写了下一代web框架koa。和Express相比，koa 1.0使用generator实现异步，代码看起来像同步的：

```
var koa = require('koa');
var app = koa();

app.use('/test', function *() {
    yield doReadFile1();
    var data = yield doReadFile2();
    this.body = data;
});

app.listen(3000);
```

用generator实现异步比回调简单了不少，但是generator的本意并不是异步。Promise才是为异步设计的，但是Promise的写法……想想就复杂。为了简化异步代码，ES7（目前是草案，还没有发布）引入了新的关键字async和await，可以轻松地把一个function变为异步模式：

```
async function () {
    var data = await fs.read('/file1');
}
```

这是JavaScript未来标准的异步代码，非常简洁，并且易于使用。

koa2

koa团队并没有止步于koa 1.0，他们非常超前地基于ES7开发了koa2，和koa 1相比，koa2完全使用Promise并配合async来实现异步。

koa2的代码看上去像这样：

```
app.use(async (ctx, next) => {
    await next();
    var data = await doReadFile();
    ctx.response.type = 'text/plain';
    ctx.response.body = data;
});
```

出于兼容性考虑，目前koa2仍支持generator的写法，但下一个版本将会去掉。

目前JavaScript处于高速进化中，ES7是大势所趋。为了紧跟时代潮流，我们将使用最新的koa2开发。

关于KOA2的官方文档和指南：https://github.com/demopark/koa-docs-Zh-CN

KOA2官方示范案例：

https://github.com/koajs/kick-off-koa

在开始任务之前，可以先通过kick-off-koa去了解最基础的中间件构建、HTTP请求响应流程。

KOA的中文手册《Koa2进阶学习笔记》：https://github.com/chenshenhai/koa2-note

KOA的原理探究《Koa.js设计模式》：https://chenshenhai.github.io/koajs-design-note/

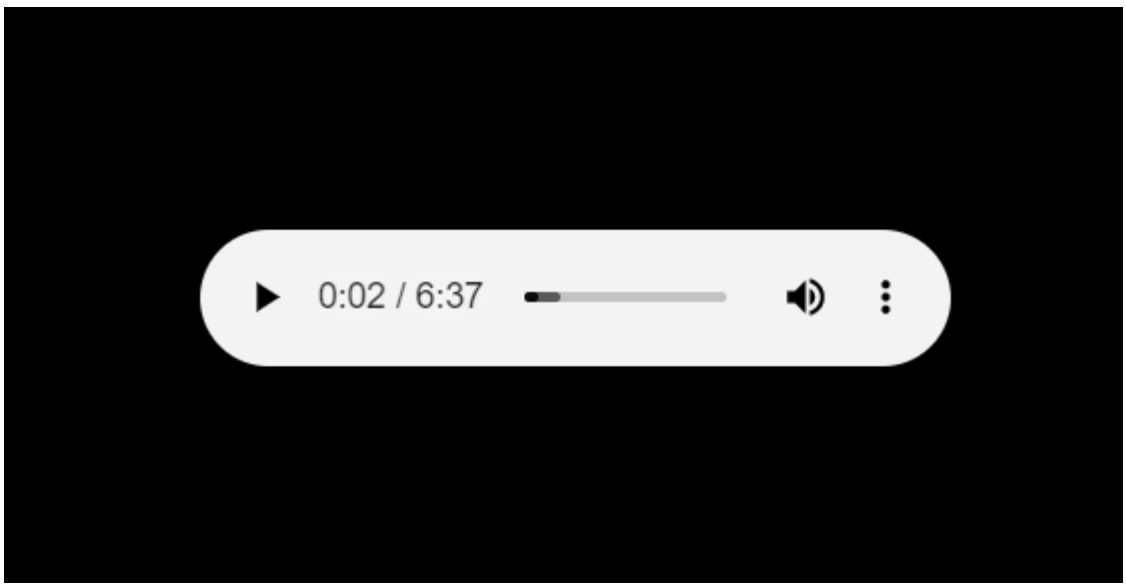
任务1

参考：https://juejin.cn/post/6844904007958265864

用KOA2实现通过get URL获取对应音频串流的的模块，例如：

http://localhost/stream/

返回音频串流。



任务2

现在，我们将开始着手处理在学习Axios的时候写的前端页面所发送的表单。

对于那个前端页面，在登陆时，发送POST请求，后端将信息与数据库中的users对比，如果正确，返回：

```
{
    "status": 0,
    "msg": "Success"
}
```

如果错误，返回：

```
{
    "status": 1,
    "msg": "Username or Password error."
}
```

在创建用户（注册）时，发送POST请求。

如果正确，返回：

```
{
    "status": 0,
    "msg": "Success"
}
```

如果错误，返回：

```
{
    "status": 1,
    "msg": "User Already Exist." // 或其它的错误信息
}
```

任务3

我们始终推荐，有现成的轮子就不要自己造。

所以，尝试集成koa-stream包以代替在任务1中写的逻辑完成串流。

https://github.com/claudetech/koa-stream

不要忘记在packages.json添加该依赖