Lesson 5

本节课的作业是编程题，并且也已经给出了参考答案地址，基本上只要认真听完课和看过文档的同学，做起来不会有太多的难度。这里没有什么特别可以说的地方，只点一下值得注意之处。

首先，从本课开始，各位正式进入了 Substrate 开发领域，看完视频之后，建议做以下两件事：

- 复习一下 Rust 知识点，尤其是工程的组织和构建，这属于基础中的基础。
- 重点参考 Substrate 的文档，地址：https://substrate.dev/ ，建议的顺序：KB > 教程 > Recipe。

重点看看教程 2 ：Build a PoE Decentralized Application，有助于对本课和下一次课的理解，它的难度小的多。

其次，请熟悉相关的工具和 IDE：

- IDE：vscode、clion、intelliJ idea rust 插件
- 包管理以及必要的镜像

第三，对于 Substrate 的关键对象理解清楚，这些内容我曾经写过一篇文章，里面已经写得比较清楚了：https://blog.dteam.top/posts/2020-06/substrate-step2.html 。

至于其他，没什么秘诀，一个字：练。

建议把官方教程中每个例子全部演练一下，然后把 recipe 仔细阅读。并且 recipe 本身也是例子驱动且也给出了代码示例和 github 仓库链接，可依次阅读和练习。

最后，检查作业时发现不少同学把题目理解错了，给出的是单 token 的实现。作业要求是一个多 token 的实现，请检查一下。

参考代码：https://github.com/substrate-developer-hub/substrate-erc20-multi/blob/master/runtime/src/erc20.rs

各位可以对比一下，如果没有以下一句话的作业，都做错了：

```
<Tokens>::insert(token_id, token);
```

Lesson 6

基本都是选择题，只需要通读文档并且安装好官方前端工程模板，自己运行一下基本就没有什么难度。并且，小鹅通中间提交作业时应该可以自己看到成绩，所以也就不重复了，这里单说重点。

1. 开发 Substrate 前端应用，建议的知识点：typescript + react + 官方包。
2. 对于有 UI 需求的应用，可以直接基于前端工程模板进行，由于基于 react，故 react 的背景知识是必要的。
3. 对于无 UI 需求或者是采用非 react 框架的前端队伍，可以直接使用前端 api 进行，也没有什么太多的难度。
4. 对于刚接触前端开发的后端工程师，需要注意的地方：
    - 了解前端应用的构建过程，因为经过多年发展，前端工程已经体系化，复杂度也高了很多，不是简单的一个 html 页面包含一个简单的 js 就完事。也已经形成了自己的包管理和构建机制。

    - 掌握 async/await 语法，对于大部分后端来讲，尤其是 java 出身的，这个语法还是比较陌生的，但相信你们会很快喜欢上它。对于 promise ，应该多少会看过类似的概念。
    - 若装了 hadoop，因为它也有 yarn 命令，注意不要跟前端常用的 yarn 冲突，运行时指明路径。
5. 同样需要注意包的镜像问题，对于 yarn 和 npm 都掌握一下。

最后，说一下前端 api 的注意事项：

- 注意它的 api 调用模式：api.<type>.<module>.<section>
- 由于依赖元数据，因此前后端的类型映射需要一致，否则，发起调用时会报错：

> Verification Error: Execution(ApiError("Could not convert parameter 'tx' between node and runtime

这种情形基本上都是类型没有映射正确，这里给出一个示范例子，对应上面教程 2 中后端的调用：

```
const api = await ApiPromise.create({
  provider: wsProvider,
  types: {
    Address: "AccountId",
    LookupSource: "AccountId",
  },
});

const content = Array.from(new Uint8Array(fs.readFileSync(name)))
  .map((b) => b.toString(16).padStart(2, "0"))
  .join("");

const hash = blake2AsHex(content, 256);
console.log(hash);

const keyring = new Keyring({ type: "sr25519" });
const alice = keyring.addFromUri("//Alice", { name: "Alice default" });

await api.tx["templateModule"]
  ["createClaim"](hash)
  .signAndSend(alice)
  .catch((e) => {
    console.log(e.toString());
    api.disconnect();
  });
```

注意上面的 type 映射、交易账户获取和交易签名部分，这些问题是新手最容易犯错的地方。

最后总结一下，掌握 Substrate 开发并不难，就两个秘诀：

- 熟读文档
- 动手练习

当然，有精力了还可以看源码，但我个人更倾向于先从文档入手，文档不足以支撑你的需求时，再深入源码去看。

附加一句：善用搜索引擎，并且只搜英文内容。