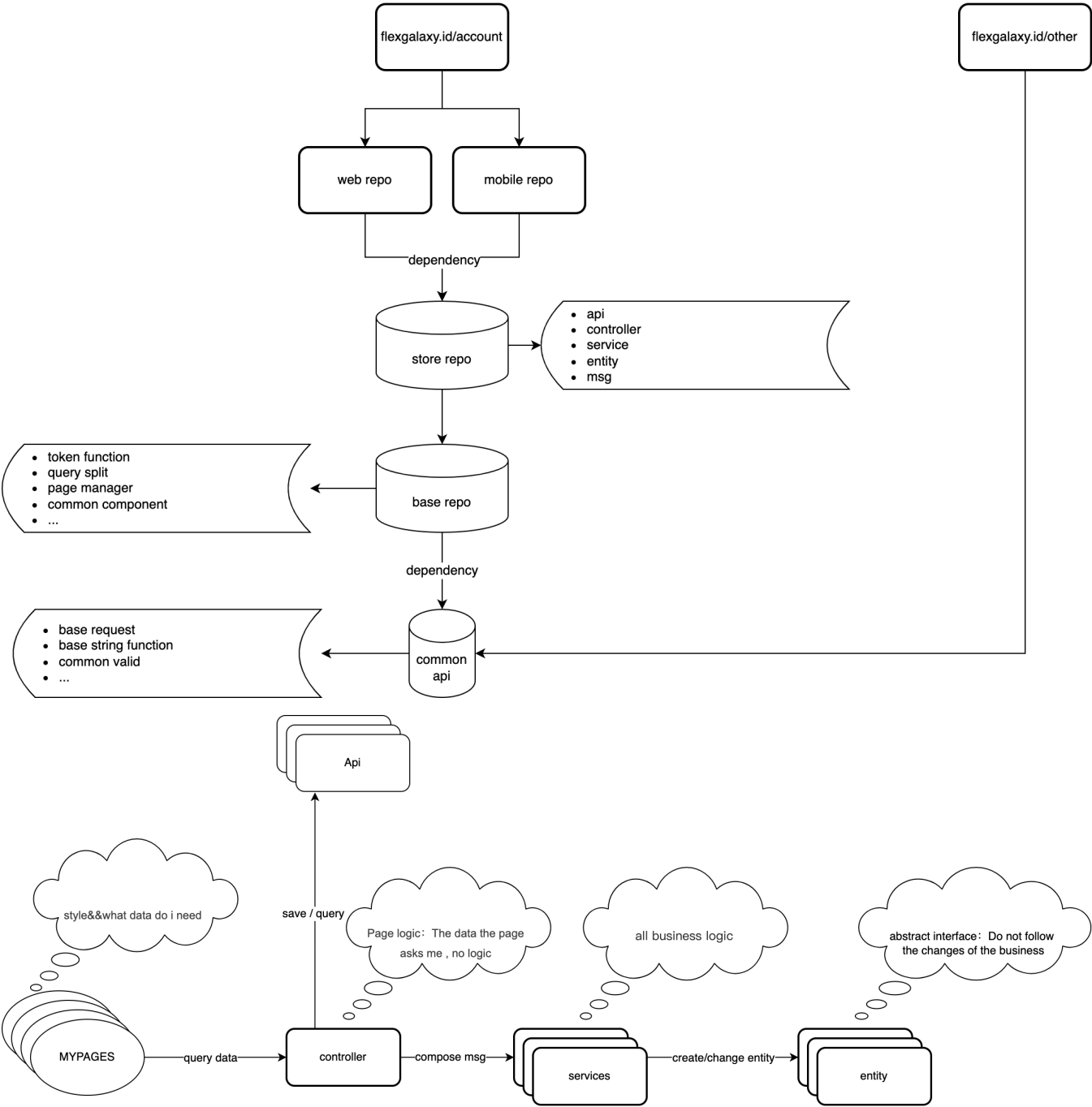


前端代码架构

- 前端代码整体结构以及数据流向
- flexgalaxy.id整体设计
 - 1.整体设计思路
 - 2.如何使用逻辑框架
 - 3.frontend business base 的工程设计
 - 4.服务的工程设计
- gRPC-web 下的单元测试
- gRPC-web 下的 流水线搭建
 - 本地环境搭建及构建脚本
- ReactHook下的单元测试的实现

前端代码整体结构以及数据流向



flexgalaxy.id整体设计

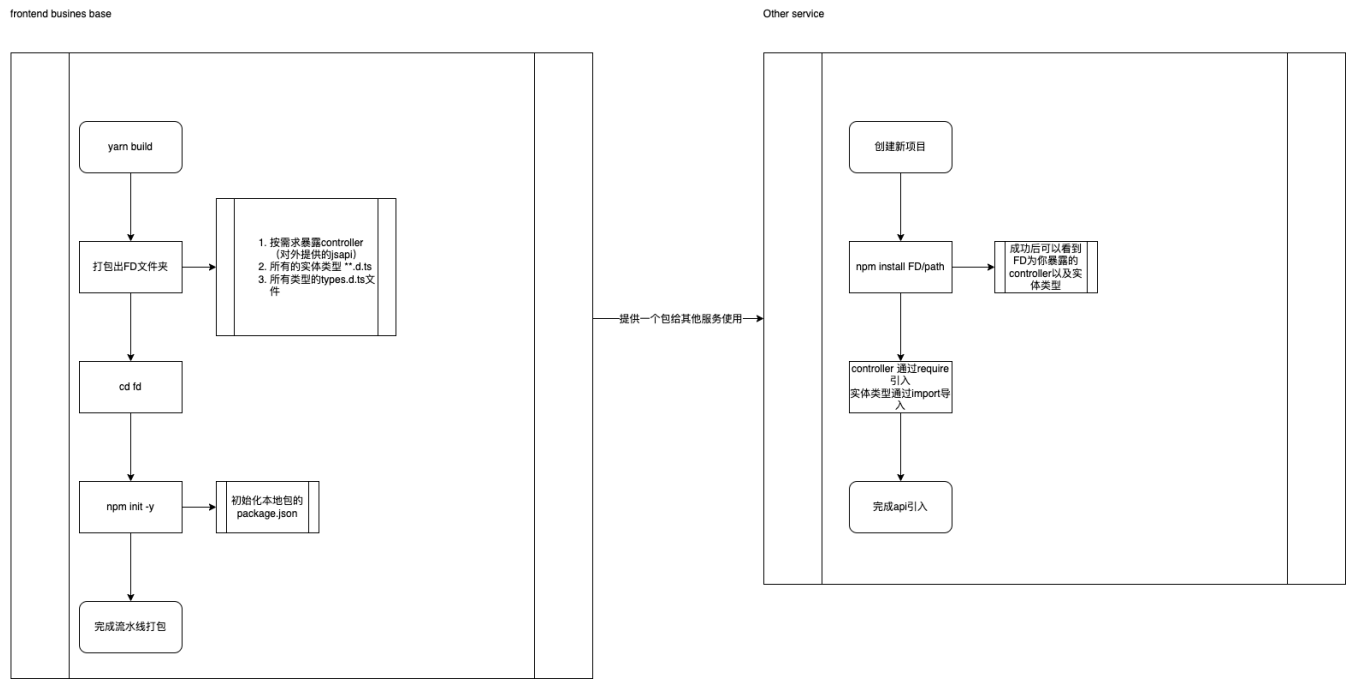
1.整体设计思路

2.如何使用逻辑框架

- 1. 是否搭载redux
 - a. redux的应用场景
 - b. redux的取舍
- 2. 如何搭载redux
- 3. redux在flexglaxy.id中的职责

3.frontend business base 的工程设计

- 1. 提供types.ts 文件导出所有的实体类型
- 2. 必须完全依赖以上的[前端代码整体结构以及数据流向](#)的设计



4.服务的工程设计

- 1. 在开发新的服务的时候通过脚本命令搭建主体
- 2. i.

gRPC-web 下的单元测试

gRPC-web 下的 流水线搭建

本地环境搭建及构建脚本

1. 将proto文件夹内容复制到本地
2. 将proto文件通过protoc解析到src/lib中
3. 脚本示例：

```
#!/bin/bash

rm -rf src/lib
mkdir -p src/lib
rm -rf com

cp -r ../../proto/src/main/proto/com .

proto_files=$(find . -name '*.proto')

for f in $proto_files;do
    echo "generating $f"
    docker run --rm \
    -v $(pwd):/app -w /app \
    --user $(id -u) \
    rvolosatovs/protoc -I=. $f \
    --js_out=import_style=commonjs:./src/lib \
    --grpc-web_out=import_style=commonjs,mode=grpcwebtext:./src/lib
done

rm -rf com
```

4. 在package.json中增加该脚本的执行

```

{
  "name": "flexgalaxy_id_portal",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@craco/craco": "^6.4.3",
    "@testing-library/jest-dom": "^5.14.1",
    "@testing-library/react": "^13.0.0",
    "@testing-library/user-event": "^13.2.1",
    "antd": "^4.20.4",
    "axios": "^0.27.2",
    "craco-less": "^2.0.0",
    "enzyme-adapter-react-16": "^1.15.6",
    "google-protobuf": "^3.20.1",
    "grpc-web": "^1.3.1",
    "react": "^18.1.0",
    "react-dom": "^18.1.0",
    "react-router-dom": "^6.3.0",
    "react-scripts": "5.0.1",
    "web-vitals": "^2.1.0"
  },
  "scripts": {
    "start": "craco start",
    "protoc-gen": "./protoc-gen.sh",
    "build": "craco build",
    "test": "jest --coverage",
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
    "extends": [
      "react-app",
      "react-app/jest"
    ]
  },
  "browserslist": {
    "production": [
      ">0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version",
      "last 1 firefox version",
      "last 1 safari version"
    ]
  },
  "devDependencies": {
    "@babel/preset-env": "^7.18.0",
    "enzyme": "^3.11.0"
  }
}

```

ReactHook下的单元测试的实现