

报修平台管理系统后端

接入项目：

```
yarn install
yarn global add nodemon ts-node typescript # 安装ts node环境
yarn global add prisma # 安装prisma
yarn add @prisma/client # 安装prisma client
npx prisma db pull
npx prisma generate
# nodemon index.ts
yarn dev
```

项目规范

- 数据库字段
 - 下划线命名
- 代码变量
 - 小驼峰命名
- 类型、类
 - 大驼峰命名

项目结构

```
.
├── Readme.md
├── package.json
├── prisma
│   └── schema.prisma
├── src
│   ├── api
│   │   ├── device.ts
│   │   ├── historicalRepairRecord.ts
│   │   ├── index.ts
│   │   ├── reimbursementRecord.ts
│   │   ├── repairForm.ts
│   │   └── user.ts
│   ├── index.ts
│   └── mock
│       ├── README.md
│       ├── device-mock.js
│       ├── device-mock.ts
│       ├── historicalRepairRecord-mock.js
│       ├── historicalRepairRecord-mock.ts
│       └── index.ts
```

```
├── reimbursemenrRecord-mock.js
├── reimbursemenrRecord-mock.ts
├── repairForm-mock.js
├── repairForm-mock.ts
├── user-mock.js
├── user-mock.ts
├── utils.js
├── utils.ts
├── tests
│   ├── device.spec.ts
│   ├── historicalRepairRecord.spec.ts
│   ├── reimbursementRecord.spec.ts
│   ├── repairForm.spec.ts
│   └── user.spec.ts
├── types
│   ├── faker.d.ts
│   ├── index.js
│   └── index.ts
├── utils
│   ├── database
│   │   ├── database.ts
│   │   └── index.ts
│   └── middleware
├── tsconfig.json
└── yarn.lock
```

使用技术栈

- koa
- Koa-router
- postgresql
- prisma

postgresql



本项目使用postgresql作为数据库，PostgreSQL是一种开源的关系型数据库管理系统（RDBMS），它提供了可扩展性、可靠性和数据完整性的功能。以下是对PostgreSQL的一些介绍：

1. **开源性和社区支持:** PostgreSQL是一种开源数据库，拥有庞大的活跃社区。这意味着它的源代码对所有人开放，并且有许多开发者贡献新功能、修复错误并提供支持。
2. **可扩展性:** PostgreSQL具有良好的可扩展性，可以处理大规模的数据和高并发访问。它支持水平和垂直扩展，可以通过添加更多的服务器节点或增加硬件资源来提高性能。
3. **丰富的功能集:** PostgreSQL提供了许多高级功能，例如事务支持、多版本并发控制（MVCC）、复杂查询、表间关联、触发器、存储过程、自定义函数等。它还支持各种数据类型（包括数组、JSON、XML等）和索引类型（B树、哈希、GIN、GiST等）。
4. **数据完整性:** PostgreSQL非常注重数据完整性和一致性。它支持各种约束（如主键、外键、唯一性约束、检查约束等），并提供了强大的数据校验和完整性保证机制。
5. **扩展性:** PostgreSQL支持扩展，允许开发者编写和安装自定义插件和扩展。这使得用户可以根据自己的需求添加额外的功能，扩展PostgreSQL的能力。
6. **安全性:** PostgreSQL提供了许多安全特性，如SSL加密连接、访问控制列表（ACL）、基于角色的访问控制（RBAC）等。它还支持数据备份和恢复，以及事务日志和崩溃恢复机制。
7. **跨平台性:** PostgreSQL可在多个操作系统上运行，包括Linux、Windows、macOS和其他类Unix系统。这使得它在不同环境中具有广泛的应用和部署灵活性。

总的来说，PostgreSQL是一种功能强大、可靠且高度可定制的开源关系型数据库管理系统。它适用于各种应用场景，从个人项目到大规模企业级应用都能发挥其优势。

prisma

本项目中，使用node作为后端语言，使用prisma框架进行数据库的关系映射。

Prisma 是一种现代化的数据库工具和 ORM（对象关系映射）框架，旨在简化和改进与数据库的交互。下面是对 Prisma 的一些介绍：

1. **类型安全的查询构建器:** Prisma 提供了一个类型安全的查询构建器，允许您使用编程语言的类型检查和自动完成来构建数据库查询。这样可以减少手写 SQL 查询的工作量，并且在编译时捕获潜在的查询错误。
2. **自动生成的模型:** Prisma 可以根据数据库架构自动生成模型。您可以使用 Prisma CLI 工具将数据库连接到 Prisma，并自动生成模型文件，使您可以轻松地在代码中使用数据库表和列的抽象表示。
3. **多数据库支持:** Prisma 支持多种主流数据库，包括 PostgreSQL、MySQL 和 SQLite。这使得您可以在不更改代码的情况下切换数据库，从而提供了灵活性和可扩展性。
4. **高性能:** Prisma 通过使用 Prisma Client，它是一个高性能的数据库访问库，将查询和数据库操作转化为高效的底层数据库语句，从而提供快速和高效的数据库访问。
5. **数据迁移:** Prisma 提供了强大的数据迁移工具，使您能够轻松地对数据库进行模式更改和迁移。您可以使用 Prisma 的数据迁移命令来创建、修改和应用数据库迁移，以保持数据库结构的一致性。
6. **实时数据库更新:** Prisma 还提供了实时数据库更新的功能，您可以订阅数据库中的变更，并通过 WebSocket 接收实时更新。这对于构建实时应用程序和实时通知功能非常有用。

总之，Prisma 是一个强大且易于使用的数据库工具和 ORM 框架，它提供了类型安全的查询构建器、自动生成的模型、多数据库支持、高性能访问、数据迁移和实时数据库更新等功能。它使得与数据库的交互更加简单和高效，并提供了更好的开发体验。

koa

本项目使用koa框架编写后端接口，与前端进行交互。

Koa 是一个现代、轻量级的 Node.js Web 应用框架，由 Express 团队开发。它基于异步函数（Async/Await）和中间件（Middleware）的概念，旨在提供更简洁、灵活和可扩展的方式来构建 Web 应用程序。以下是对 Koa 的一些介绍：

1. **轻量级和精简:** Koa 的设计理念是保持核心的代码量尽可能少，并鼓励使用中间件来实现更具可读性和可扩展性的应用。这使得 Koa 被认为是一个轻量级框架，同时它提供了丰富的扩展能力。
2. **基于异步函数:** Koa 利用了 ES2017/ES8 中引入的异步函数（Async/Await）特性。这使得编写异步代码变得更加直观和简单，而无需显式处理回调函数或使用额外的库。
3. **中间件架构:** Koa 的核心概念是中间件。中间件是一个函数，接收 `context` 对象和一个 `next` 函数作为参数，可以对请求和响应进行处理。Koa 应用程序由一系列中间件组成，形成一个处理请求的管道。这种中间件架构使得编写和组合中间件变得非常灵活。
4. **上下文（Context）对象:** Koa 在每个请求处理中引入了一个上下文（Context）对象，提供了对请求和响应的封装。上下文对象包含了请求和响应的相关信息，并提供了许多有用的方法和属性，使得在中间件和路由处理程序中更方便地操作请求和响应。
5. **错误处理:** Koa 提供了一个便捷的错误处理机制，通过 `try/catch` 来捕获和处理错误。这使得错误处理变得简单明了，并且可以更好地控制错误的传播和响应。
6. **可扩展性:** Koa 具有丰富的插件和中间件生态系统，使得开发者可以方便地扩展应用的功能。从身份验证和会话管理到日志记录和缓存，Koa 的插件生态系统提供了各种功能性扩展。

总而言之，Koa 是一个现代化的、轻量级的 Node.js Web 应用框架，借助于异步函数和中间件的概念，提供了简洁、灵活和可扩展的方式来构建 Web 应用程序。它具有简单的错误处理、中间件架构、上下文对象和丰富的插件生态系统等特点，使得开发者能够更轻松地构建高性能的 Web 应用。

ER图

设备模块

接口名称	请求方法	请求路径	请求参数	请求体	响应数据
创建设备	POST	/device/	无	deviceKind (number null) deviceName (string null) deviceAddress (string null) deviceQrcode (string null)	id (number) deviceKind (number null) deviceName (string null) deviceAddress (string null) deviceQrcode (string null)
删除设备	DELETE	/device/:id	id (number, path)	无	id (number) deviceKind (number null) deviceName (string null) deviceAddress (string null) deviceQrcode (string null)
更新设备	PUT	/device/:id	id (number, path)	deviceKind (number null) deviceName (string null) deviceAddress (string null) deviceQrcode (string null)	id (number) deviceKind (number null) deviceName (string null) deviceAddress (string null) deviceQrcode (string null)
获取设备列表	GET	/device/	无	无	数组，包含以下属性的设备对象： id (number) deviceKind (number null) deviceName (string null) deviceAddress (string null) deviceQrcode (string null)
获取单个设备	GET	/device/:id	id (number, path)	无	id (number) deviceKind (number null) deviceName (string null) deviceAddress (string null) deviceQrcode (string null)

用户模块

接口名称	请求方法	请求路径	请求参数	请求体	响应数据
创建用户	POST	/user/	无	idcardcheck (number null) username (string null) account (string null) idcard (string null) password (string null) phone (string null) role (string null)	id (number) idcardcheck (number
删除用户	DELETE	/user/:id	id (number, path)	无	id (number) idcardcheck (number null) username (string null) account (string null) idcard (string null) password (string null) phone (string null) role (string null)
更新用户	PUT	/user/:id	id (number, path)	idcardcheck (number null) username (string null) account (string null) idcard (string null) password (string null) phone (string null) role (string null)	id (number) idcardcheck (number null) username (string null) account (string null) idcard (string null) password (string null) phone (string null) role (string null)
获取用户列表	GET	/user/	无	无	数组，包含以下属性的设备对象： id (number) idcardcheck (number null) username (string null) account (string null) idcard (string null) password (string null) phone (string null) role (string null)
获取单个用户	GET	/user/:id	id (number, path)	无	id (number) idcardcheck (number null) username (string null) account (string null) idcard (string null) password (string null) phone (string null) role (string null)

工单模块

接口名称	请求方法	请求路径	请求参数	请求体	响应数据
创建工单	POST	/repair/	无	deviceId (number null) userId (number	null) faultDesc (string
删除工单	DELETE	/repair/:id	id (number, path)	无	id (number) deviceId (number null) userId (number
更新工单	PUT	/repair/:id	id (number, path)	deviceId (number	null) userId (number
获取工单列表	GET	/repair/	无	无	数组，包含以下属性的设备对象： id (number) deviceId (number
获取单个工单	GET	/repair/:id	id (number, path)	无	id (number) deviceId (number null) userId (number null) faultDesc (string null) faultPic (string null) doorTime (Timestamp null)

历史维修记录模块

接口名称	请求方法	请求路径	请求参数	请求体	响应数据
创建历史维修记录	POST	/historical-repair/	无	deviceName (string null) repairPerson (string null) repairAddress (string null) repairReason (string	null) commentStatus (string null)
删除历史维修记录	DELETE	/historical-repair/:id	id (number, path)	无	id (number) deviceName (string null) repairPerson (string null) repairAddress (string null) repairReason (string
更新历史维修记录	PUT	/historical-repair/:id	id (number, path)	deviceName (string	null) repairPerson (string
获取历史维修记录列表	GET	/historical-repair/	无	无	数组，包含以下属性的设备对象： id (number) deviceName (string
获取单个历史维修记录	GET	/historical-repair/:id	id (number, path)	无	id (number) deviceName (string

报销单模块

接口名称	请求方法	请求路径	请求参数	请求体	响应数据
创建报销单	POST	/reimburse/	无	<div>applicantName (string null)</div> <div>seller (string null)</div> <div>taxPrice (number null)</div> <div>billingDate (Timestamp null)</div> <div>status (boolean</div>	<div>null)</div> <div>approver (string null)</div>
删除报销单	DELETE	/reimburse/:id	id (number, path)	无	<div>id (number)</div> <div>applicantName (string null)</div> <div>seller (string null)</div> <div>taxPrice (number null)</div> <div>billingDate (Timestamp null)</div> <div>status (boolean</div>
更新报销单	PUT	/reimburse/:id	id (number, path)	<div>applicantName (string null)</div> <div>seller (string null)</div> <div>taxPrice (number null)</div> <div>billingDate (Timestamp null)</div> <div>status (boolean</div>	<div>null)</div> <div>approver (string null)</div>
获取报销单列表	GET	/reimburse/	无	无	数组，包含以下属性的设备对象： <div>id (number)</div> <div>applicantName (string null)</div> <div>seller (string null)</div> <div>taxPrice (number null)</div> <div>billingDate (Timestamp null)</div> <div>status (boolean</div>
获取单个报销单	GET	/reimburse/:id	id (number, path)	无	<div>id (number)</div> <div>applicantName (string null)</div> <div>seller (string null)</div> <div>taxPrice (number null)</div> <div>billingDate (Timestamp null)</div> <div>status (boolean</div>