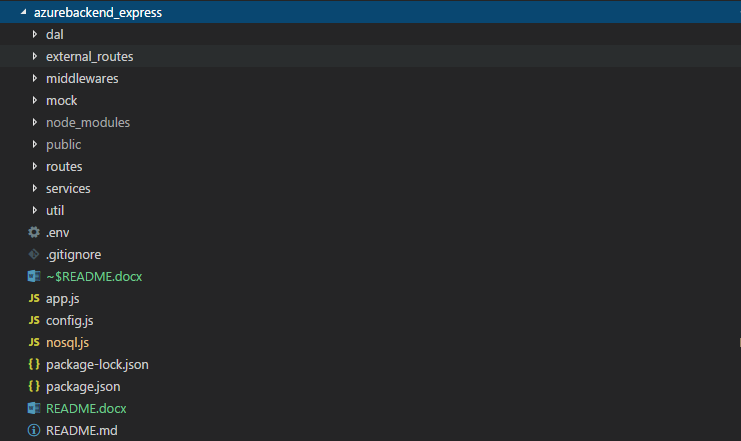
Azure Backend Project Documentation

**目录结构**



根目录

1. dal: CosmosDB数据访问层
2. external\_routes: 对外配置信息获取接口，均已ext\_api开头
3. middlewares: 自定义express中间件，包括用户认证， 单页面路由，
4. mock: 所有AD用户属性
5. public：azurefront\_react所有输出文件（gitignore）
6. routes: 所有express的路由, almost restful，例如login, signup, table & attributes …
7. service: 对dal层的业务封装，以及一些sql server的数据查询
8. util： 公用库
9. app.js: 项目的入口文件，http的监听以及中间件/路由的注册
10. config.js: 自定义的配置，例如表名，端口等

项目使用

环境：

1. node.js v8.9.4及以上

启动：

1. cmd切换到根目录下
2. 运行npm install，自动安装项目所有依赖
3. 运行npm start启动http服务。如果启动时不带端口参数，默认以4000端口启动*（iisnode下会在process.env.PORT变量中存储端口，必须以该端口启动）*

部署（azure）：

1. 克隆仓库：

git clone <https://nextlabs@cloudproxy.scm.azurewebsites.net:443/cloudproxy.git>

1. 添加远程仓库并起名azure：

git remote add azure <https://nextlabs@cloudproxy.scm.azurewebsites.net/cloudproxy.git>

1. 复制项目根目录下除.git/, node\_modules, .gitignore以外的文件到克隆仓库根目录
2. 类似git提交修改并上传：

* 暂存所有修改和未跟踪文件： git add -- .
* 提交修改并添加信息： git commit –m ‘xxx’
* 上传并自动部署： git push azure

1. 提交时所需用户名和密码：

* 用户名：Nextlabs 密码：123Next!

参考资料

1. Express JS：<https://expressjs.com/>
2. Tedious JS：<http://tediousjs.github.io/tedious/>
3. DocumentDB JS：[https://github.com/Azure/azure-documentdb-node](https://github.com/Azure/azure-documentdb-node%20)