# edge-book

后端服务低代码框架开发文档

> 文档完善中

## 代码示例

- 酒店系统后台完整项目

## 联系

huiloademail@163.com

核心功能

定义全局函数，用于接口之间复用的代码，一般在funcs目录下的lua文件，通过`funcs.call("checkbook", ...args)`进行调用，返回字符串格式

```
-- name=checkbook
-- entry=index
-- args=[
-- {"name": "start_date", "type": "string"},
-- {"name": "end_date", "type": "string"},
-- {"name": "roomid", "type": "number"}
-- ]

function index(start_date, end_date, roomid)
    res = state.orm()
        .table({"rooms"})
        .select({
            "rooms.*"
        })
        .joins({
            {"JOIN booking ON rooms.id = booking.roomid"}
        })
        .where({
            "rooms.id = ? AND booking.deleted_at IS NULL AND (\
                (booking.book_start < ? AND booking.book_end > ?)\
            )", roomid, end_date, start_date
        })
        .find({
            {"unique", "id"}
        })
        .exec("base", false)
    if res.err ~= nil or next(res.res) ~= nil then
        return "false"
    else
        return "true"
    end
end
```

数据层，支持sqlite、mysql、postgresql数据库

支持数据版本的前进后退

文件结构

- migrations
  - mysql
    - 1
      - xxx.aql
    - 2
      - xxx.sql
  - sqlite
    - 1
      - xxx.sql
    - 2
      - xxx.sql

# 简介

```
services:
  adduser:
    url: /api/user/add # api地址
    method: post # 接口方法，支持使用`,`分割多个方法
    middlewares: # 使用全局接口中定义的名字，在这里直接引入使用
      - local
      - ipLimit
    request:
      data:
        - name: name
          type: string
          mode: query
        - name: info
          type: string
          mode: query
    response:
      type: application/json
      mock:
        default:
          content: '{"code": 0, "msg": "ok"}'
      script: |
        state("base", "addone",
{name=req('name'),info=req('info')});
        resjson(200, "ok");
```

# request

定义请求参数

`mode`类型

- query
- header
- form
- uri

# response

`type`定义数据的返回类型

`mock`定义服务的mock数据

`script`

> script语法详情查看第5章介绍

定义请求处理的表达式

该语法提供了基础数据结构能力: number、string、array、dict以及一些操作方法

提供了对于state应用的操作能力

edge框架内置中间件及其用法

# 简介

使用预定义的中间件，初始化后，能够在其他的接口中使用

```yaml
middlewares:
  local: # 全局中间件的命名
    id: local # 使用什么中间件
  ipLimit:
    id: ipLimit
    opts: # 该中间件初始化时候的参数
      block_time: 5
```

# 中间件列表

## local

只允许本地网络才能连接

- `127.0.0.1`
- `192.168.x.x`
- `172.16.x.x ~ 172.31.x.x`
- `10.x.x.x`

## ipLimit

对于接口访问的限速

参数

- block_time: 每n秒访问一次

## local

仅限本机地址访问

```
middlewares:
  backproxy:
    id: local
```

## iplimit

## basicauth

## tokenlimit

## leaklimit

## 跨域请求

cors

## 反向代理

```yaml
middlewares:
  backproxy:
    id: backproxy
    opts:
      srv: edgetest
      key: query@key
```

# staticfile