# Next+Koa2开发github

## 项目准备

### 安装

### 手动搭建

yarn add react react-dom next

根目录创建pages文件夹

#### Package.json文件：

{

"name": "next-github",

"version": "1.0.0",

"description": "",

"main": "index.js",

"scripts": {

"test": "echo \"Error: no test specified\" && exit 1",

"dev": "next",

"build": "next build",

"start": "next start"

},

"author": "",

"license": "ISC",

"dependencies": {

"next": "^9.0.5",

"react": "^16.9.0",

"react-dom": "^16.9.0"

}

}

增加一下代码

"scripts": {

"dev": "next",

"build": "next build",

"start": "next start"

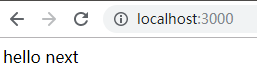
},

yarn dev 打开开发环境

#### pages目录下新建index.js文件

import React from 'react'

export default () => <span>hello next</span>



### 2. create-next-app

安装

npm i -g create-next-app

使用

npx create-next-app next-create 或者 yarn create next-app next-create 安装next项目

上面命令都创建了名为next-create的项目

### Next作为koa中间件使用

#### 安装koa

yarn add koa

根目录下创建server.js

const Koa = require('koa')

const next = require('next')

const dev = process.env.NODE\_ENV !== 'production'

const app = next({ dev })

const handle = app.getRequestHandler()

app.prepare().then(() => {

const server = new Koa()

server.use(async (ctx, next) => {

await handle(ctx.req, ctx.res)

ctx.respond = false

})

})

Package.json文件修改scripts

"dev": "node server.js",

yarn dev 启动

## koa的使用

### 基础用法

#### server.js

const Koa = require('koa')

const next = require('next')

const dev = process.env.NODE\_ENV !== 'production'

const app = next({ dev })

const handle = app.getRequestHandler()

const server = new Koa()

server.use(async (ctx, next) => {

const path = ctx.path

const method = ctx.method

ctx.body = `<span>Koa Render ${path} ${method}</span>`

await next()

})

server.use(async (ctx, next) => {

ctx.body = '<span>Koa Render2</span>'

})

server.listen(3000, () => {

console.log('koa server listening on 3000...')

})

### Koa-router

#### 安装koa-router

yarn add koa-router

#### koa-router简单使用

const Router = require('koa-router')

const router = new Router()

server.use(router.routes())

#### 示例：

const Koa = require('koa')

const Router = require('koa-router')

const next = require('next')

const dev = process.env.NODE\_ENV !== 'production'

const app = next({ dev })

const handle = app.getRequestHandler()

const server = new Koa()

const router = new Router()

router.get('/test', async (ctx) => {

ctx.body = `<p>this is /test </p>`

})

router.get('/test/:id', async (ctx) => {

ctx.body = `<p>this is /test ${ctx.params.id}</p>`

})

router.get('/json', async (ctx) => {

ctx.body = { success: true }

// ctx.set('Content-Type', 'application/json')

})

server.use(router.routes())

server.listen(3000, () => {

console.log('koa server listening on 3000...')

})

## Redis的安装使用

### 安装redise

cnpm i koa-generic-session koa-redis --save

### \* redis 远程连接方法

1、修改redis服务器的配置文件

vi redis.conf

注释以下绑定的主机地址

# bind 127.0.0.1

或

vim redis.conf

bind 0.0.0.0

protected-mode no

!!!!坑 修改配置文件启动时，记得指定配置文件

redis-server ./redis.conf

### 远程连接：

redis-cli -h 129.28.187.206 -p 6379

命令：

keys \* :反馈存储的所有key值

get key值 :获取指定key的数据

hget fix name :不经过sesson读取redis

### redis的基础使用

#### 连接redis

redis-cli

#### 获取一个值

get a

#### 设置一个值

set b 111

#### 配置密码

Redis.conf文件修改一下代码

requirepass foobared 修改密码为foobared

#### 连接：

Redis-cli –p 6379

#### 认证密码

auth foobared

#### 设置过期时间

setex c 10 1 设置一个a过期时间为10秒 值为1

#### 设置带前缀类的key

set session:sessionId 123

#### 获取

get session:sessionId

#### 查看所有key值

Keys \*

#### 删除某个key

del a

## nodejs连接redis数据库

### ioredis 模块

安装 yarn add ioredis

### 使用

#### test-redis.js

async function test() {

const Redis = require('ioredis')

const redis = new Redis({

port: 6379,

host: '129.28.187.206'

// password: '123456'

})

await redis.set('c', 123)

await redis.setex('d', 10 , 2345)

const keys = await redis.keys('\*')

console.log(keys, await redis.get('c'), await redis.get('d'))

setTimeout(async () => {

console.log(await redis.get('d'))

},13000)

}

test()

效果：



## 项目引入ant-design

### 安装

注意！ Nextjs 默认是不支持直接import引入css文件的， 需要第三方支持

yarn add @zeit/next-css

项目根目录新建文件

### next-config.js

const widthCss = require('@zeit/next-css')

if (typeof require !== undefined) {

require.extensions['.css'] = file => {}

}

module.exports = widthCss({})