

02

OPEN ORIENTED

凹凸实验室

nextjs体验

袁龙

React应用的服务端渲染方案

默认服务端渲染

自动代码分割

基于page的客户端路由

Hot Module Replacement

可自己配置服务

可自定义webpack配置

install:

```
npm install next@beta react react-dom --save
```

package.json:

```
{  
  "scripts": {  
    "dev": "next",  
    "build": "next build",  
    "start": "next start"  
  }  
}
```

项目结构

```
.
├── components
│   ├── Header.js
│   └── MyLayout.js
├── package.json
├── pages
│   ├── about.js
│   ├── index.js
│   └── post.js
└── static
```

```
<style jsx>{`  
  p {  
    color: blue;  
  }  
  div {  
    background: red;  
  }  
  @media (max-width: 600px) {  
    div {  
      background: blue;  
    }  
  }  
`}</style>  
<style global jsx>{`  
  body {  
    background: black;  
  }  
`}</style>
```

```
export default () => (  
    
)
```

自定义head

```
import Head from 'next/head'
export default () => (
  <div>
    <Head>
      <title>My page title</title>
      <meta name="viewport" content="initial-scale=1.0, width=device-width" />
    </Head>
    <p>Hello world!</p>
  </div>
)
```

Link

Router


```
static async getInitialProps () {  
  const res = await fetch('https://api.github.com/repos/zeit/next.js')  
  const json = await res.json()  
  return { stars: json.stargazers_count }  
}
```

page load , 服务端执行, 路由跳转, 客户端执行

```
<Link prefetch href="/">  
  <a style={linkStyle}>Home</a>  
</Link>
```

```
Router.prefetch('/post')
```

预加载只会预加载页面的JS

自定义服务与路由

02

```
<Link as={` /p/${show.id}`} href={` /post2?id=${show.id}`}>
  <a>{show.name}</a>
</Link>
```

```
app.prepare()
  .then(() => {
    const server = express()

    server.get('/p/:id', (req, res) => {
      const actualPage = '/post2'
      const queryParams = { title: req.params.id }
      app.render(req, res, actualPage, queryParams)
    })

    server.get('*', (req, res) => {
      return handle(req, res)
    })

    server.listen(3002, (err) => {
      if (err) throw err
      console.log('> Ready on http://localhost:3002')
    })
  })
  .catch((ex) => {
    console.error(ex.stack)
    process.exit(1)
  })
```

next.config.js

```
module.exports = {  
  distDir: 'build',  
  assetPrefix: isProd ? 'http://test.jd.com' : '',  
  webpack: (config) => {}  
}
```

```
.babelrc    {  
  "presets": [  
    "next/babel",  
    "stage-0"  
  ],  
}
```

发布

2

next build
next start

更多

Dynamic Import

Custom error handling

Custom <Document>

Static HTML export

<https://github.com/zeit/next.js>

<https://github.com/zeit/next.js/tree/v3-beta/examples>