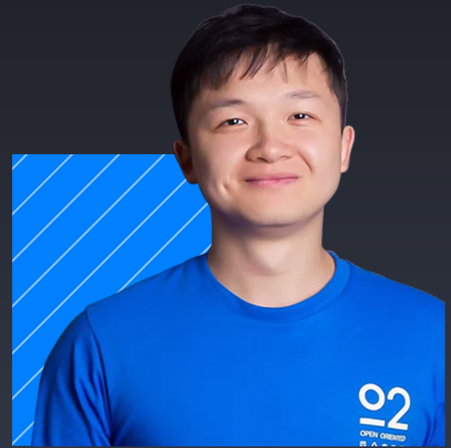
掘金

用 React 开发小程序的探索之路

李伟涛 | 高级工程师 @ 京东凹凸实验室



OPEN ORIENTED
凹 凸 实 验 室

掘金

1 原生开发之痛

2 如何使用 React 开发小程序

3 重生之路

4 开源探索

5 面向未来

02
OPEN ORIENTED
凹 凸 实 验 室

掘金

1 原生开发之痛

OPEN ORIENTED
凹 凸 实 验 室

原生开发之痛

项目入口

页面

组件

掘金

OPEN ORIENTED
凹 凸 实 验 室

原生开发之痛

掘金

项目入口

页面

组件

JS 逻辑交互

配置文件

WXML 模板

样式文件

02
OPEN ORIENTED
凹 凸 实 验 室

原生开发之痛

```
App({
  onLaunch: function () {
    wx.showToast({…})
  },
  onError: function () {},
  onShow: function () {},
  onHide: function () {}
})
})
```

```
Page({
  onLoad: function () {},
  onReady: function () {},
  onShow: function () {},
  onHide: function () {}
})
```

掘金

02
OPEN ORIENTED
凹 凸 实 验 室

原生开发之痛

掘金

```
<view bindtap=“clickHandler”>
 <text>{showText}</text>
</view>
```

OPEN ORIENTED
凹 凸 实 验 室

原生开发之痛

开发体验

性能瓶颈

掘金

OPEN ORIENTED
凹 凸 实 验 室

原生开发之痛

掘金

开发体验

OPEN ORIENTED
凹 凸 实 验 室

原生开发之痛 代码组织稍显复杂

掘金

JS 逻辑交互

配置文件

WXML 模板

样式文件

02
OPEN ORIENTED
凹 凸 实 验 室

原生开发之痛 编码体验不够顺畅

掘金

```
Page ({
  onLoad: function () {
    wx.showToast({…})
  },
  onReady: function () {},
  onShow: function () {},
  onHide: function () {}
})
```

02
OPEN ORIENTED
凹 凸 实 验 室

原生开发之痛 编码体验不够顺畅

掘金

页面/组件无法继承

```
Page({
  onLoad: function () {
    wx.showToast({…})
  },
  onReady: function () {},
  onShow: function () {},
  onHide: function () {}
})
```

OPEN ORIENTED
凹 凸 实 验 室

原生开发之痛 编码体验不够顺畅

掘金

```
Page({
  onLoad: function () {
    wx.showToast({…})
  },
  onReady: function () {},
  onShow: function () {},
  onHide: function () {}
})
```

页面/组件无法继承

缺乏智能提示

OPEN ORIENTED
凹 凸 实 验 室

原生开发之痛 字符串模板稍显孱弱

掘金

```
<view bindtap=“clickHandler”>
  {{formatDate(dateTime)}}
</view>
```

日期格式化

02
OPEN ORIENTED
凹 凸 实 验 室

原生开发之痛 代码规范不统一

session-from	String		会话来源	open-type="contact"	1.4.0
send-message-title	String	当前标题	会话内消息卡片标题	open-type="contact"	1.5.0
send-message-path	String	当前分享路径	会话内消息卡片点击跳转小程序路径	open-type="contact"	1.5.0
send-message-img	String	截图	会话内消息卡片图片	open-type="contact"	1.5.0
show-message-card	Boolean	false	显示会话内消息卡片	open-type="contact"	1.5.0
bindcontact	Handler		客服消息回调	open-type="contact"	1.5.0
bindgetphonenumber	Handler		获取用户手机号回调	open-type="getphonenumber"	1.2.0

原生开发之痛 缺乏统一的自动化编译处理

掘金

ECMAScript

Sass

{less}

OPEN ORIENTED
凹 凸 实 验 室

原生开发之痛

掘金

性能瓶颈

OPEN ORIENTED
凹 凸 实 验 室

原生开发之痛 性能瓶颈

掘金

原生开发之痛 性能瓶颈

掘金

避免频繁 setData

setData 避免传入大规模数据

OPEN ORIENTED
凹 凸 实 验 室

掘金

优秀的解决方案

02
OPEN ORIENTED
凹 凸 实 验 室

原生开发之痛

掘金

mpvue

wepy

OPEN ORIENTED
凹 凸 实 验 室

原生开发之痛

掘金

OPEN ORIENTED
凹 凸 实 验 室

原生开发之痛

掘金

2 如何使用 React 开发小程序

OPEN ORIENTED
凹 凸 实 验 室

如何使用 React 开发小程序

掘金

OPEN ORIENTED
凹 凸 实 验 室

如何使用 React 开发小程序

掘金

代码规范

OPEN ORIENTED
凹 凸 实 验 室

如何使用 React 开发小程序

掘金

代码规范

JSX 书写方便

OPEN ORIENTED
凹 凸 实 验 室

如何使用 React 开发小程序

掘金

代码规范

JSX 书写方便

天生组件化

OPEN ORIENTED
凹 凸 实 验 室

如何使用 React 开发小程序

掘金

```
View = F(Data)
```

OPEN ORIENTED
凹 凸 实 验 室

如何使用 React 开发小程序

掘金

`this.setState()`

`this.setData()`

OPEN ORIENTED
凹 凸 实 验 室

如何使用 React 开发小程序

掘金

JS 代码

生命周期

模板

OPEN ORIENTED
凹 凸 实 验 室

如何使用 React 开发小程序 JS 代码对比

掘金

```jsx
import React, { Component } from 'react'
class MyComponent extends Component {
  state = { title: '这是一个标题' }
  render () {
    return (
      <div className='my-component'>
        <div>{this.state.title}</div>
      </div>
    )
  }
}
```

React

```jsx
Component({
  data: { title: '这是一个标题' },
  methods: {
    handler () {
      this.setData({ title: '接受点击' })
    }
  }
})
```

小程序

如何使用 React 开发小程序 生命周期对比

小程序	React
created	componentWillMount
attached	componentDidMount
ready	shouldComponentUpdate
detached	componentWillUpdate
moved	componentDidUpdate
	componentWillRecieveProps

掘金

02
OPEN ORIENTED
凹 凸 实 验 室

如何使用 React 开发小程序 模板对比

掘金

JSX

```jsx
return (
  <div className='my-component'>
    <div>{this.state.title}</div>
  </div>
)
```

WXML

```wxml
<view class="my-component">
  <text>{title}</text>
  <view bindtap="handler">测试</view>
</view>
```

OPEN ORIENTED
凹 凸 实 验 室

掘金

使用 React 来开发小程序

OPEN ORIENTED
凹 凸 实 验 室

如何使用 React 开发小程序

掘金

使用 React 来开发小程序？

OPEN ORIENTED
凹 凸 实 验 室

如何使用 React 开发小程序

编译原理

掘金

源代码

OPEN ORIENTED
凹 凸 实 验 室

如何使用 React 开发小程序

编译原理

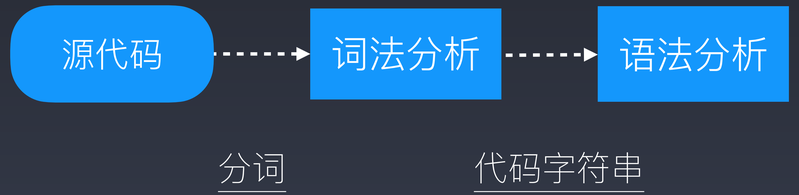
掘金

分词

OPEN ORIENTED
凹 凸 实 验 室

如何使用 React 开发小程序

编译原理



掘金

OPEN ORIENTED
凹 凸 实 验 室

如何使用 React 开发小程序

掘金

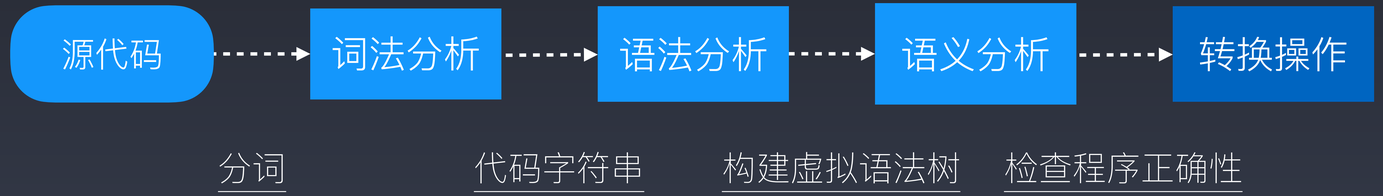
编译原理

OPEN ORIENTED
凹 凸 实 验 室

如何使用 React 开发小程序

掘金

编译原理

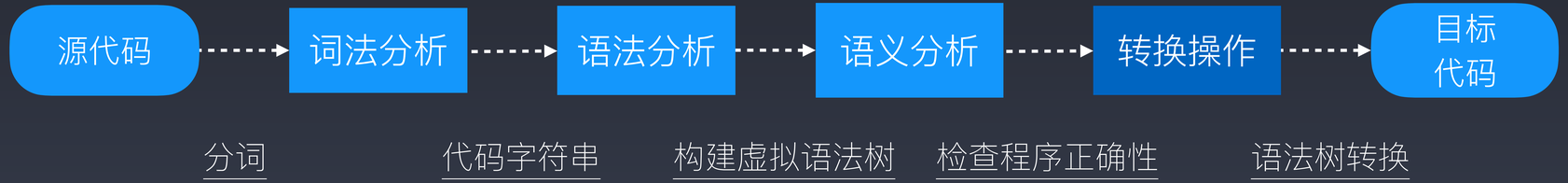


OPEN ORIENTED
凹 凸 实 验 室

如何使用 React 开发小程序

掘金

编译原理



OPEN ORIENTED
凹 凸 实 验 室

如何使用 React 开发小程序

掘金

编译原理

OPEN ORIENTED
凹 凸 实 验 室

如何使用 React 开发小程序

掘金

ESTree Spec

https://github.com/estree/estree

OPEN ORIENTED
凹 凸 实 验 室

如何使用 React 开发小程序

ESTree Spec

掘金

```
extend interface Program {
  sourceType: ‘script’ | ‘module’;
  body: [Statement | ModuleDeclaration]
}
```

OPEN ORIENTED
凹 凸 实 验 室

如何使用 React 开发小程序

ESTree Spec

Functions

Statements

Declarations

...

掘金

OPEN ORIENTED
凹 凸 实 验 室

如何使用 React 开发小程序

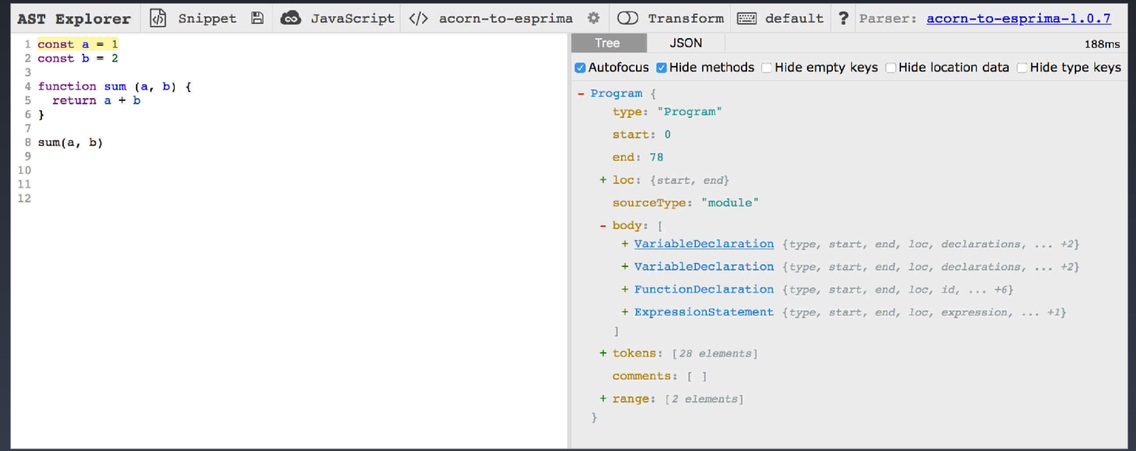
```
const a = 1
const b = 2

function sum (a, b) {
  return a + b
}

sum(a, b)
```

掘金

OPEN ORIENTED
凹 凸 实 验 室

如何使用 React 开发小程序

掘金



https://astexplorer.net/

02
OPEN ORIENTED
凹 凸 实 验 室

如何使用 React 开发小程序

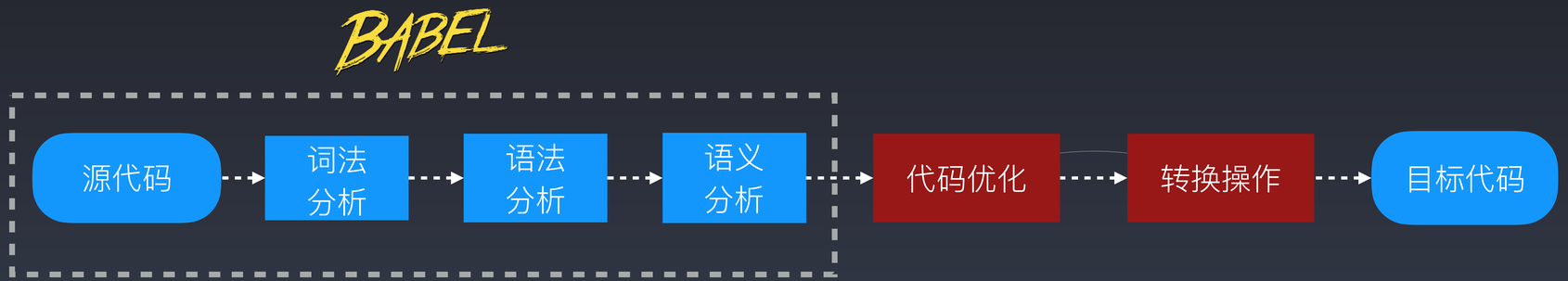
掘金

babel-core	解析代码获取 AST
babel-types	AST 节点定义，生成节点
babel-traverse	递归遍历操作 AST
babel-generator	AST 生成源码

02
OPEN ORIENTED
凹 凸 实 验 室

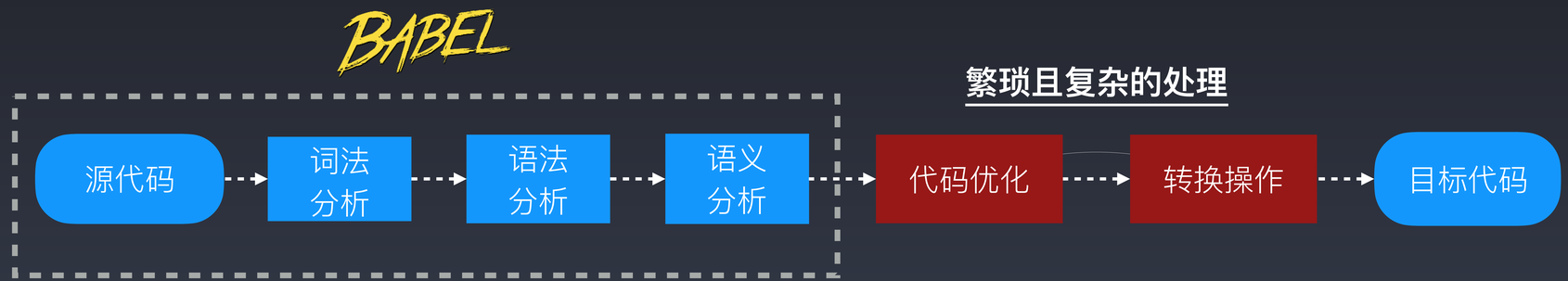
如何使用 React 开发小程序

掘金



OPEN ORIENTED
凹 凸 实 验 室

如何使用 React 开发小程序



掘金

OPEN ORIENTED
凹 凸 实 验 室

掘金

Taro 中具体实现

- 编译时处理
- 运行时适配

02
OPEN ORIENTED
凹 凸 实 验 室

掘金

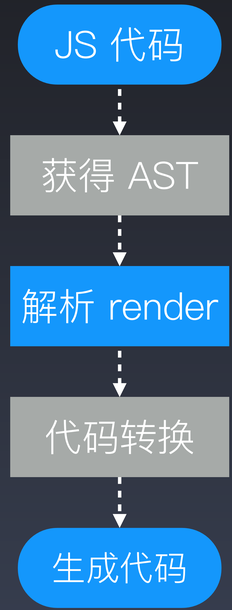
Taro 中具体实现

- 编译时处理
- 运行时适配

OPEN ORIENTED
凹 凸 实 验 室

如何使用 React 开发小程序 编译时处理

```
import Taro, { Component } from '@tarojs/taro'
class MyComponent extends Component {
  state = { title: '这是一个标题'}
  render () {
    return (
      <View className='my-component'>
        <View>{this.state.title}</View>
      </View>
    )
  }
}
```



掘金

02
OPEN ORIENTED
凹 凸 实 验 室

如何使用 React 开发小程序 编译时处理

掘金

解析 render

```
render () {
  const { a } = this.state
  const t = parseData(a)
  return (
    <View className='my-component'>
      <Text>{t}</Text>
      <View onClick={this.handler}>测试</View>
    </View>
  )
}
```

构建计算函数 _createData

生成 WXML 模板

02
OPEN ORIENTED
凹 凸 实 验 室

如何使用 React 开发小程序 编译时处理

掘金

解析 render

```
render () {
  const { a } = this.state
  const t = parseData(a)
  return (
    <View className='my-component'>
      <Text>{t}</Text>
      <View onClick={this.handler}>测试</View>
    </View>
  )
}
```

```
_createData () {
  this.__state = arguments[0] || this.state || {}
  var a = this.state.a
  var t = parseData(a)
  Object.assign(this.__state, {
    t
  })
  return this.__state
}
```

02 OPEN ORIENTED 凹凸实验室

如何使用 React 开发小程序 编译时处理

掘金

解析 render

```
render () {
  const { a } = this.state
  const t = parseData(a)
  return (
    <View className='my-component'>
      <Text>{t}</Text>
      <View onClick={this.handler}>测试</View>
    </View>
  )
}
```

→

```
<view class="my-component">
  <text>{title}</text>
  <view bindtap="handler">测试
</view>
</view>
```

02
OPEN ORIENTED
凹 凸 实 验 室

如何使用 React 开发小程序 编译时处理

解析 render

掘金

if…else…	wx:if wx:else
data.map(item => …)	wx:for=“data” wx:for–item=“item”
&&	wx:if
属性计算	复杂表达式

。。。

02
OPEN ORIENTED
凹 凸 实 验 室

如何使用 React 开发小程序 编译时处理

掘金

思考题

```
render () {
  const { a } = this.state
  const t = parseData(a)
  return (
    <View className='my-component' style={{color: 'red'}}>
      <Text>{t}</Text>
      <View onClick={this.handler}>测试</View>
    </View>
  )
}
```

实现 style 传入对象

02
OPEN ORIENTED
凹 凸 实 验 室

掘金

Taro 中具体实现

- 编译时处理
- 运行时适配

OPEN ORIENTED
凹 凸 实 验 室

如何使用 React 开发小程序 运行时适配

运行时框架

掘金

生命周期适配

事件处理

OPEN ORIENTED
凹 凸 实 验 室

掘金

开发工具配合

OPEN ORIENTED
凹 凸 实 验 室

如何使用 React 开发小程序

掘金

源代码

输入

Taro CLI

NPM 包管理

ES Next 支持

CSS 预处理器支持

代码压缩

……

输出代码

OPEN ORIENTED
凹 凸 实 验 室

掘金

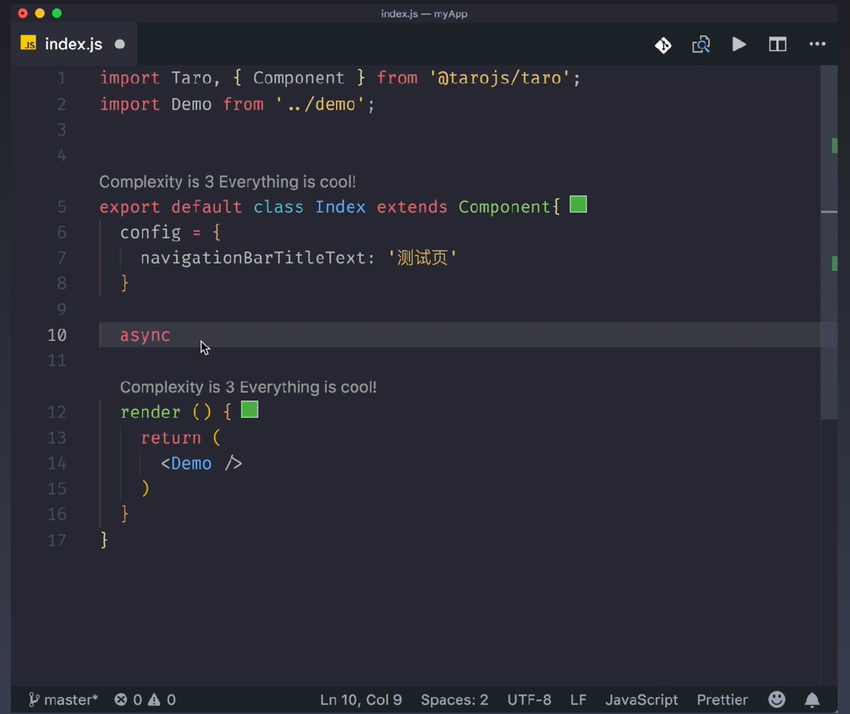
配合良好的开发体验

OPEN ORIENTED
凹 凸 实 验 室

如何使用 React 开发小程序

掘金

智能的代码提示

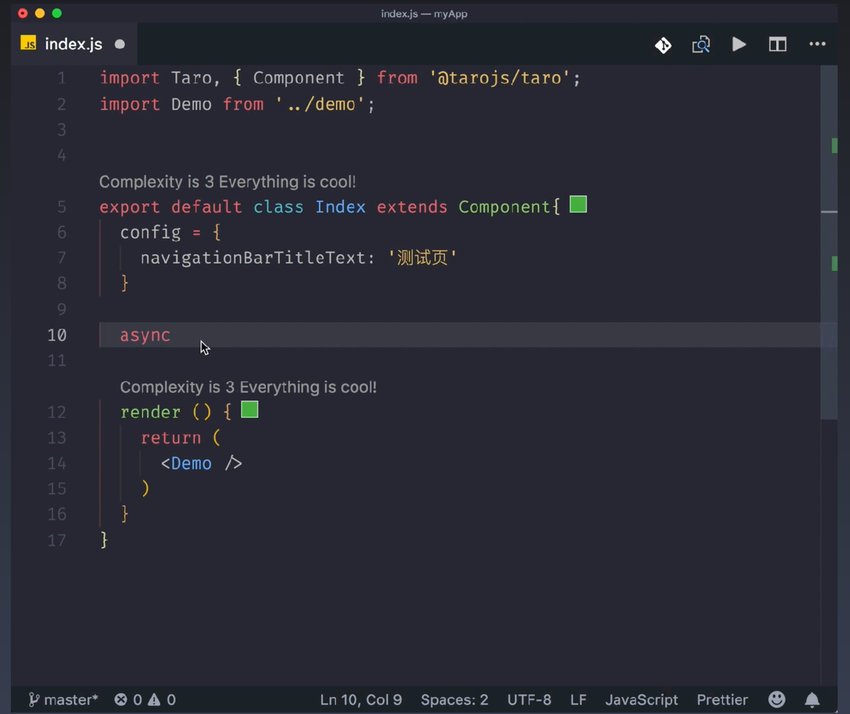


02
OPEN ORIENTED
凹 凸 实 验 室

如何使用 React 开发小程序

掘金

智能的代码提示

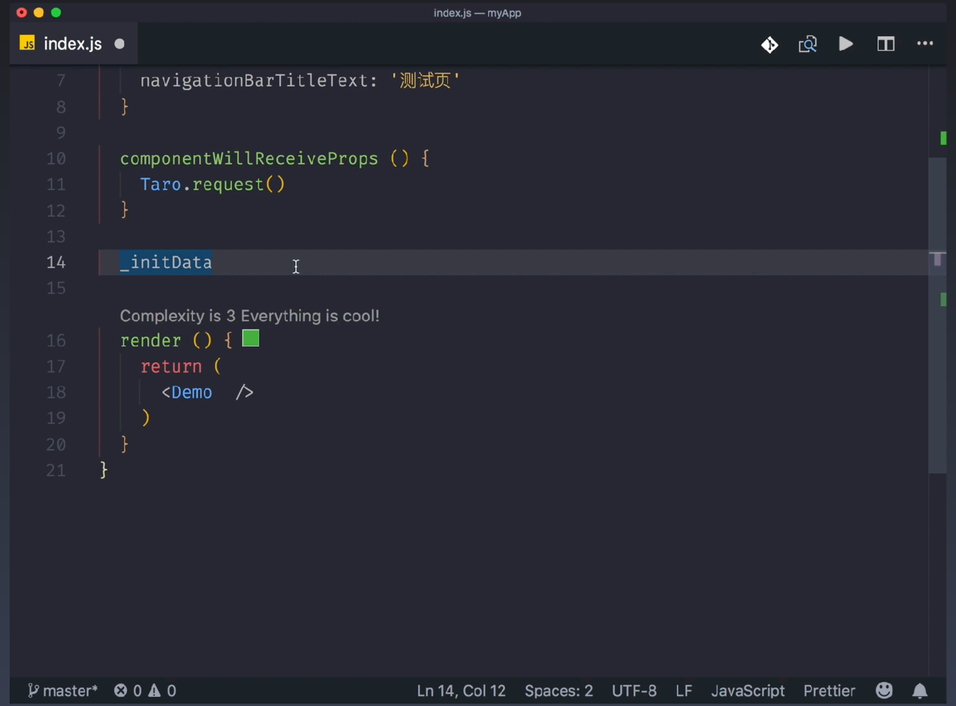


02
OPEN ORIENTED
凹 凸 实 验 室

如何使用 React 开发小程序

掘金

健全的错误代码检查

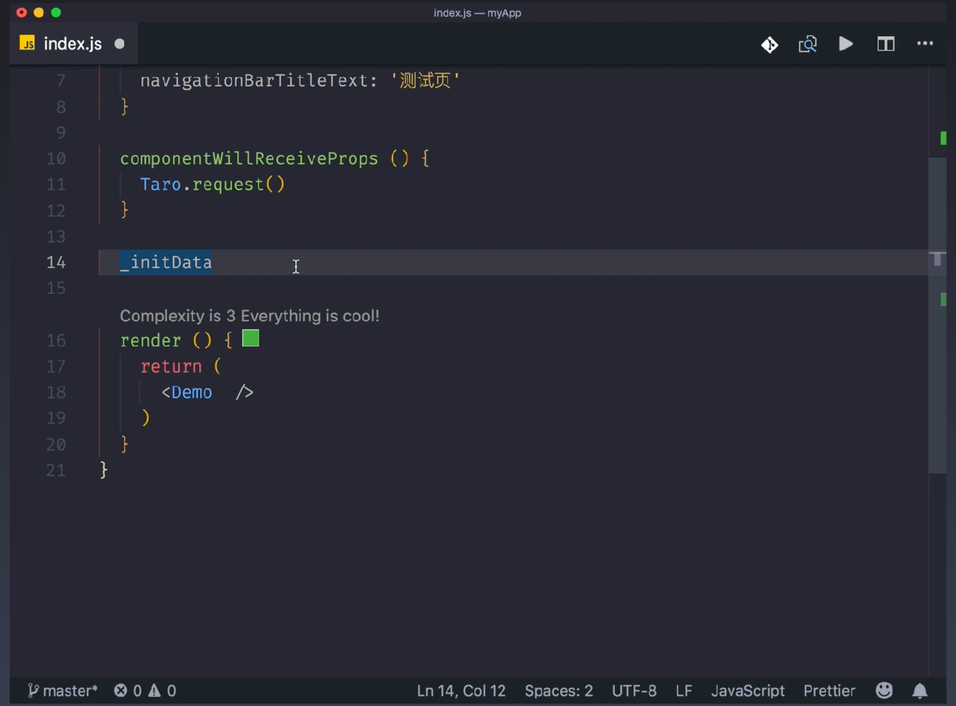


02
OPEN ORIENTED
凹 凸 实 验 室

如何使用 React 开发小程序

掘金

健全的错误代码检查



02
OPEN ORIENTED
凹 凸 实 验 室

掘金

3 重生之路

OPEN ORIENTED
凹 凸 实 验 室

掘金

问题一

丰富的 JSX 语法支持

重生之路

掘金

```
render () {
  const { a } = this.state
  const t = parseData(a)
  return (
    <View className='my-component'>
      <Text>{t}</Text>
      <View>测试</View>
    </View>
  )
}
```

OPEN ORIENTED
凹 凸 实 验 室

重生之路

```
render () {
  const { a } = this.state
  const t = parseData(a)
  return (
    <View className='my-component'>
      <Text>{t}</Text>
      <View>测试</View>
    </View>
  )
}
```

render 方法定义变量

掘金

OPEN ORIENTED
凹 凸 实 验 室

重生之路

```
render () {
  const { a } = this.state
  const time = new Date(a)
  return (
    <View className='my-component'>
      <Text>{parseDate(time)}</Text>
      <View className={classnames(styleOb)}>测试</View>
    </View>
  )
}
```

掘金

OPEN ORIENTED
凹 凸 实 验 室

重生之路

掘金

```
render () {
  const { a } = this.state
  const time = new Date(a)
  return (
    <View className='my-component'>
      <Text>{parseDate(time)}</Text>
      <View className={classnames(styleOb)}>测试</View>
    </View>
  )
}
```

JSX 中调用方法处理数据

属性支持传入复杂表达式

02
OPEN ORIENTED
凹 凸 实 验 室

重生之路

```
render () {
  const { list } = this.state
  return (
    <View className='my-component'>
      {list.map(item => {
        const style = parseStyle(item)
        return <Text style={style}>item</Text>
      })}
      <View>测试</View>
    </View>
  )
}
```

掘金

02
OPEN ORIENTED
凹 凸 实 验 室

重生之路

掘金

JSX Map 循环中定义变量

```
render () {
  const { list } = this.state
  return (
    <View className='my-component'>
      {list.map(item => {
        const style = parseStyle(item)
        return <Text style={style}>item</Text>
      })}
      <View>测试</View>
    </View>
  )
}
```

02
OPEN ORIENTED
凹 凸 实 验 室

重生之路

```
render () {
  const { list } = this.state
  return (
    <View className='my-component'>
      <Text>{1}</Text>
      <View onClick={this.handler.bind(this, '参数')}>测试</View>
    </View>
  )
}
```

掘金

02
OPEN ORIENTED
凹 凸 实 验 室

重生之路

掘金

```
render () {
  const { list } = this.state
  return (
    <View className='my-component'>
      <Text>{1}</Text>
      <View onClick={this.handler.bind(this, '参数')}>测试</View>
    </View>
  )
}
```

JSX 元素绑定事件传参

02
OPEN ORIENTED
凹 凸 实 验 室

重生之路

掘金

OPEN ORIENTED
凹 凸 实 验 室

编译时处理

重生之路

编译时处理

运行时补充

掘金

OPEN ORIENTED
凹 凸 实 验 室

重生之路

掘金

Bug！

OPEN ORIENTED
凹 凸 实 验 室

重生之路

掘金

React 支持，你不支持

OPEN ORIENTED
凹 凸 实 验 室

重生之路

React 支持，你不支持

我觉得不行

掘金

OPEN ORIENTED
凹 凸 实 验 室

掘金

持续增加，持续完善

OPEN ORIENTED
凹 凸 实 验 室

掘金

问题二

组件化实现方案

02
OPEN ORIENTED
凹 凸 实 验 室

重生之路

掘金

<template />

OPEN ORIENTED
凹 凸 实 验 室

重生之路

```
import Taro, { Component } from '@tarojs/taro'
import { View } from '@tarojs/components'
import A from '../components/A/A'

class Test extends Component {
  render () {
    return (
      <View>
        <A />
      </View>
    )
  }
}
```

组件 Test 中引用子组件 A

掘金

重生之路

```
<template src="../components/A/A.wxml">

<view>
  <template is="A" data="{{...A}}">
</view>
```

template 模板引用

掘金

OPEN ORIENTED
凹 凸 实 验 室

重生之路

组件化问题

掘金

- JS 逻辑与模板分离，组件传参非常麻烦
- 循环组件、组件嵌套各种 Bug
- 自定义组件无法嵌套子元素

重生之路

掘金

修 Bug 修到怀疑人生

OPEN ORIENTED
凹 凸 实 验 室

重生之路

掘金

Commit **900** 余次

OPEN ORIENTED
凹 凸 实 验 室

掘金

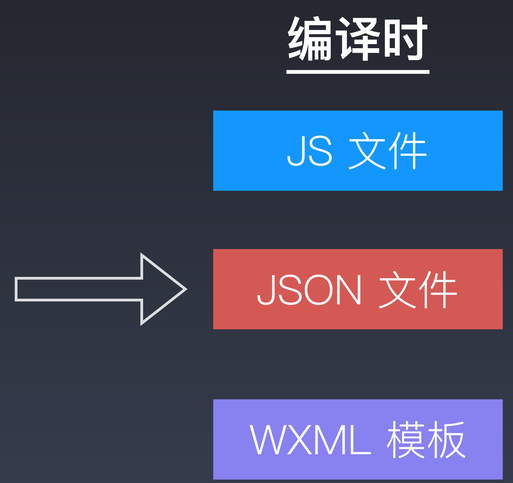
组件化推倒重构

02
OPEN ORIENTED
凹 凸 实 验 室

掘金

重生之路

```
import React, { Component } from 'react'
export default class MyComponent extends Component {
  state = { title: '这是一个标题'}
  render () {
    return (
      <div className='my-component'>
        <div>{this.state.title}</div>
      </div>
    )
  }
}
```



02
OPEN ORIENTED
凹 凸 实 验 室

重生之路

掘金

运行时

Taro 组件 → 小程序原生组件

组件 data 初始化

生命周期适配

组件传入函数

。。。

OPEN ORIENTED
凹 凸 实 验 室

重生之路

掘金

- 小程序端组件化更加健壮
- 尽可能减少由于框架带来的性能问题
- 依托官方组件化，方便以后解锁更多可能

02
OPEN ORIENTED
凹 凸 实 验 室

掘金

重要改进

小程序端性能优化

OPEN ORIENTED
凹 凸 实 验 室

重生之路

setData 性能问题

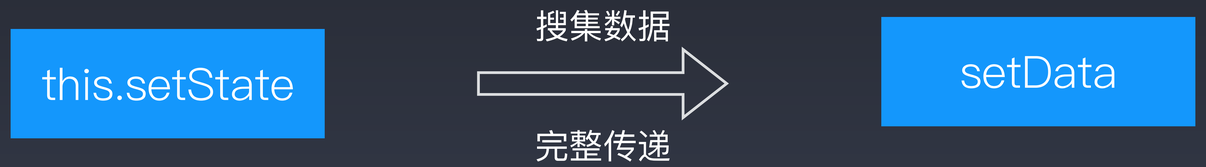
掘金

02
OPEN ORIENTED
凹 凸 实 验 室

重生之路

setState 性能升级

掘金



OPEN ORIENTED
凹 凸 实 验 室

重生之路

setState 性能升级

掘金

原始数据

```
this.state = {
  a: [1, 1],
  b: {
    x: 1
    y: 2
  }
}
```

OPEN ORIENTED
凹 凸 实 验 室

重生之路

setState 性能升级

掘金

原始数据

```
this.state = {
  a: [1, 1],
  b: {
    x: 1
    y: 2
  }
}
```

```
this.setState({
  a: [1, 2, 3],
  b: {
    x: 10
  }
})
```

OPEN ORIENTED
凹 凸 实 验 室

重生之路

setState 性能升级

掘金

原始数据

```
this.state = {
  a: [1, 1],
  b: {
    x: 1
    y: 2
  }
}
```

```
this.setState({
  a: [1, 2, 3],
  b: {
    x: 10
  }
})
```

OPEN ORIENTED
凹 凸 实 验 室

重生之路

setState 性能升级

掘金

原始数据

```
this.state = {
  a: [1, 1],
  b: {
    x: 1
    y: 2
  }
}
```

```
this.setState({
  a: [1, 2, 3],
  b: {
    x: 10
  }
})
```


```
this.setData({
  a: [1, 2, 3],
  b: {
    x: 10
  }
})
```

OPEN ORIENTED
凹 凸 实 验 室

重生之路

setState 性能升级

掘金

避免频繁 setData

避免 setData 传入大量数据

OPEN ORIENTED
凹 凸 实 验 室

重生之路

setState 性能升级

掘金

OPEN ORIENTED
凹 凸 实 验 室

重生之路

setState 性能升级

掘金

```
this.setState({
  a: [1, 2, 3],
  b: {
    x: 10
  }
})
```

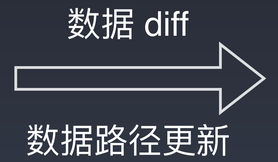
OPEN ORIENTED
凹 凸 实 验 室

重生之路

setState 性能升级

掘金

```
this.setState({
  a: [1, 2, 3],
  b: {
    x: 10
  }
})
```



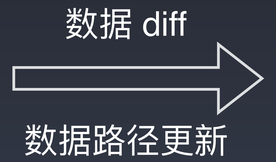
OPEN ORIENTED
凹 凸 实 验 室

重生之路

setState 性能升级

掘金

```
this.setState({
  a: [1, 2, 3],
  b: {
    x: 10
  }
})
```




```
this.setData({
  'a[2]': 2,
  ‘a[3]': 3,
  'b.x': 10
})
```

OPEN ORIENTED
凹 凸 实 验 室

我们还做了更多

- 使用小程序第三方组件
- 与原生小程序进行混写
- 更加健全的 TypeScript 支持

掘金

4 开源探索

OPEN ORIENTED
凹 凸 实 验 室

开源探索

掘金

NervJS/taro

https://github.com/NervJS/taro

6月7日正式对外开源

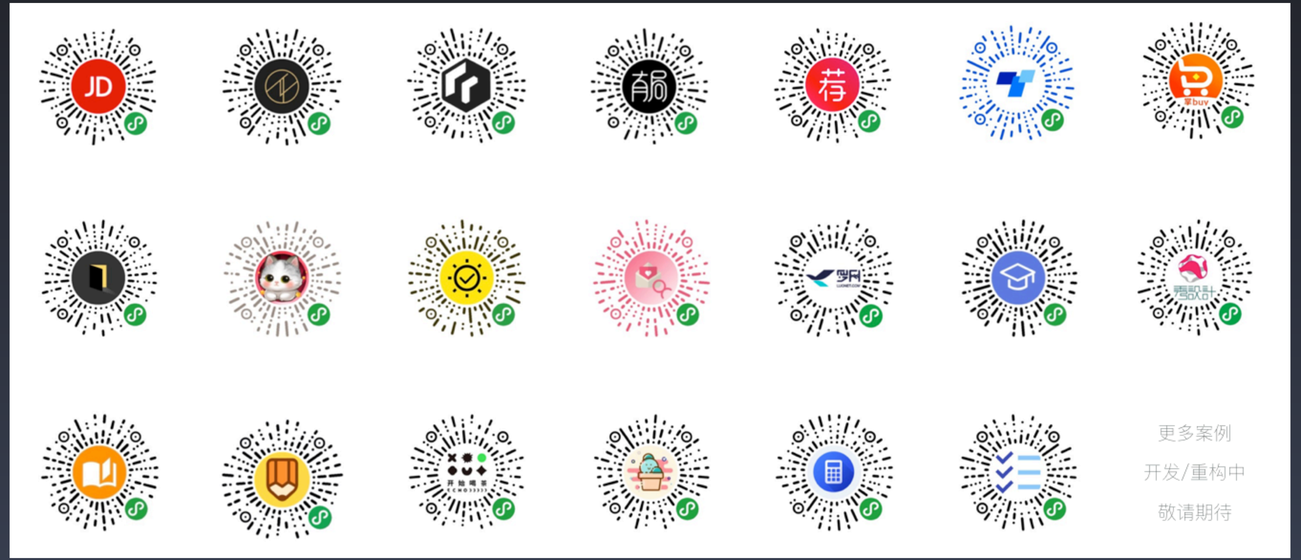
issue 393 / 500

PR 121 / 123

OPEN ORIENTED
凹 凸 实 验 室

开源探索

掘金



02
OPEN ORIENTED
凹 凸 实 验 室

开源探索

掘金

微信交流群

线下交流会

OPEN ORIENTED
凹 凸 实 验 室

开源探索

Taro 开发者计划

掘金

OPEN ORIENTED
凹 凸 实 验 室

开源探索

Taro 开发者计划

掘金

核心开发者

OPEN ORIENTED
凹 凸 实 验 室

开源探索

Taro 开发者计划

掘金

核心开发者

战略合作伙伴

OPEN ORIENTED
凹 凸 实 验 室

开源探索

Taro 开发者计划

掘金

核心开发者

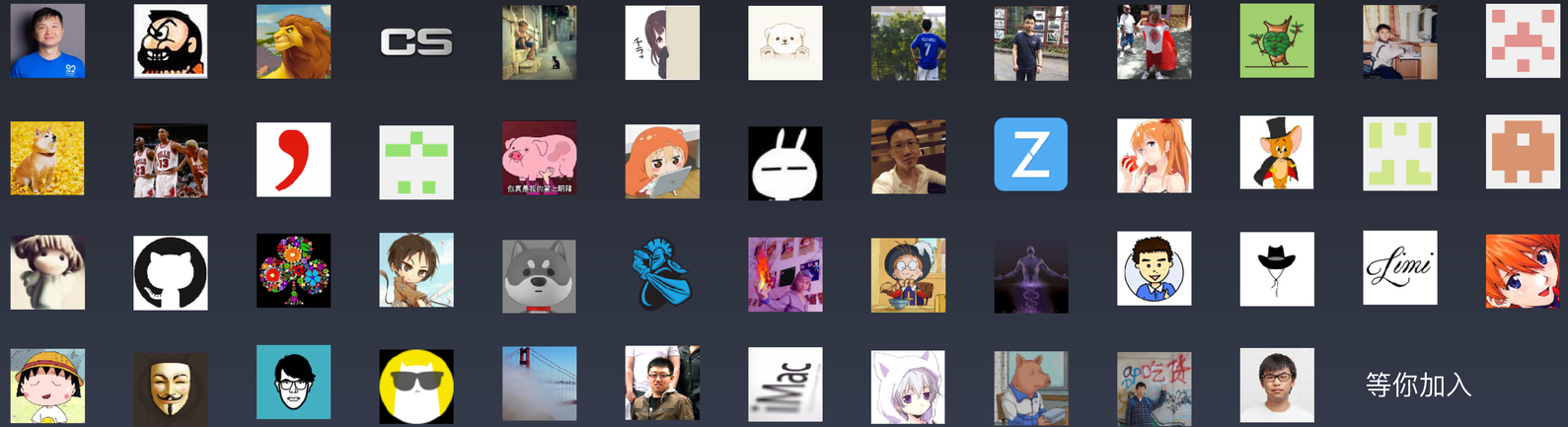
战略合作伙伴

优秀贡献者

OPEN ORIENTED
凹 凸 实 验 室

开源探索 贡献者们

掘金



OPEN ORIENTED
凹 凸 实 验 室

掘金

5 面向未来

OPEN ORIENTED
凹 凸 实 验 室

面向未来

掘金

Taro doctor

便捷测试

```
class Test extends Component {
  componentwillupdate () { }
  render () {
    this.setState({ x: 1 })
    return (
      <View>
        <A />
      </View>
    )
  }
}
```

OPEN ORIENTED
凹 凸 实 验 室

面向未来

掘金

Taro doctor

便捷测试

```
class Test extends Component {
  componentwillupdate () { }
  render () {
    this.setState({ x: 1 })
    return (
      <View>
        <A />
      </View>
    )
  }
}
```

拼写错误

OPEN ORIENTED
凹 凸 实 验 室

面向未来

掘金

Taro doctor

便捷测试

```
class Test extends Component {
  componentwillupdate () { }
  render () {
    this.setState({ x: 1 })
    return (
      <View>
        <A />
      </View>
    )
  }
}
```

拼写错误

不允许这么用

OPEN ORIENTED
凹 凸 实 验 室

面向未来

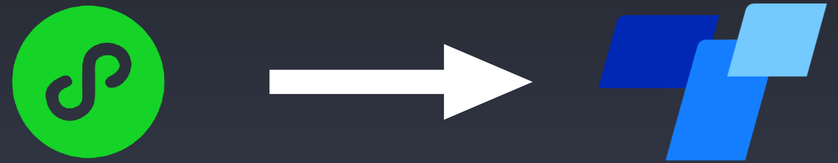
多端同步调试

掘金

OPEN ORIENTED
凹 凸 实 验 室

面向未来

小程序代码转 Taro 代码



掘金

OPEN ORIENTED
凹 凸 实 验 室

面向未来

与 React 新特性保持同步

掘金

生命周期

API

OPEN ORIENTED
凹 凸 实 验 室

面向未来

多端支持

掘金

React Native

即将发布

快应用

正在开发中

百度智能小程序

计划中

支付宝小程序

计划中

OPEN ORIENTED
凹 凸 实 验 室

面向未来

完善生态

掘金

OPEN ORIENTED
凹 凸 实 验 室

掘金

6 One More Thing

OPEN ORIENTED
凹 凸 实 验 室

掘金

正式发布 1.0 版本

OPEN ORIENTED
凹 凸 实 验 室

掘金

Taro

正式发布 1.0 版本

- 更健壮的组件化
- 支持引用小程序第三方组件库
- 与原生小程序代码混写
- 极致的性能优化
- 更加丰富的 JSX 语法支持
- 全面支持 TypeScript
- React Native 转换支持
- ……

OPEN ORIENTED
凹 凸 实 验 室

掘金

Thank you !

跟我在沸点 AMA 交流

OPEN ORIENTED
凹 凸 实 验 室