

React Native 埋坑之路

自我介绍

- 刺杰
- 格图（北京）科技有限公司
- 平时使用的技术栈：Node.js
- React Native 开发原生应用

格图



主题

- React Native 介绍
- React Native 实战经验
- Redux 架构介绍
- 热更新部署 CodePush

React Native 是什么？

- React Native 让你可以使用JavaScript来构建移动应用程序
- 一次学习，随处编写
- 混合开发
- 高效的UI开发 UI调试

示例代码

```
import React, { Component } from 'react';
import { AppRegistry, Text } from 'react-native';

class HelloWorldApp extends Component {
  render() {
    return (
      <Text>Hello world!< /Text>
    );
  }
}

AppRegistry.registerComponent('HelloWorldApp', () => HelloWorldApp);
```

一个组件包括什么？

- 导入 react react-native 组件
- 导入自定义 组件
- 编写class 主体 render jsx
- 编写styles

```
import React, { Component } from 'react';
import { ...
} from 'react-native';

import TabBar from '../components/TabBar';
import util from '../util.js';
import ModalBox from 'react-native-modalbox';

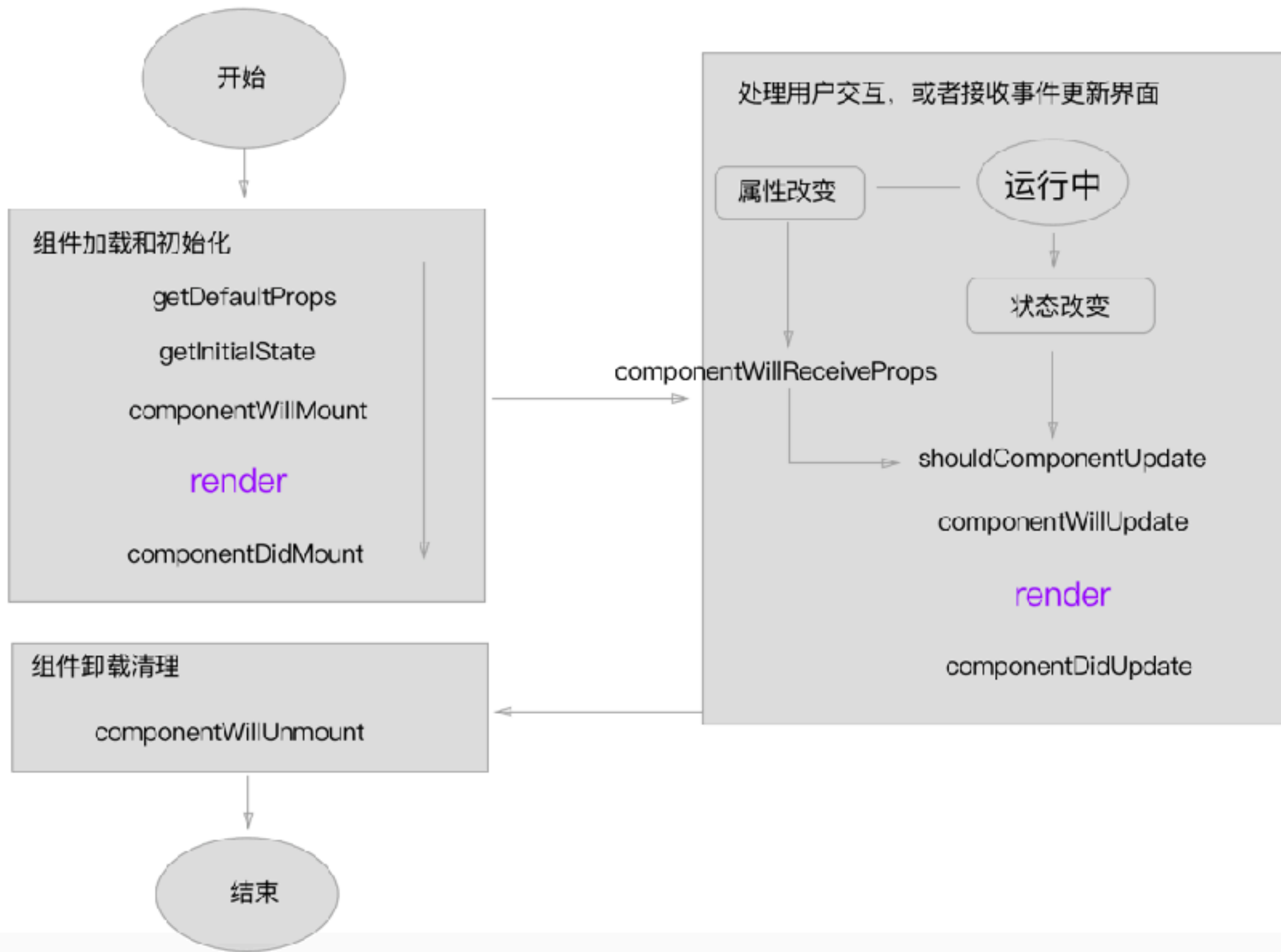
class Login extends Component {
  constructor(props) { ...
  }

  _login(phone, password) { ...
  }

  render() { ...
  }
}

let styles = StyleSheet.create({ ...
});

export default Login;
```



一个组件的生命周期

组件通信

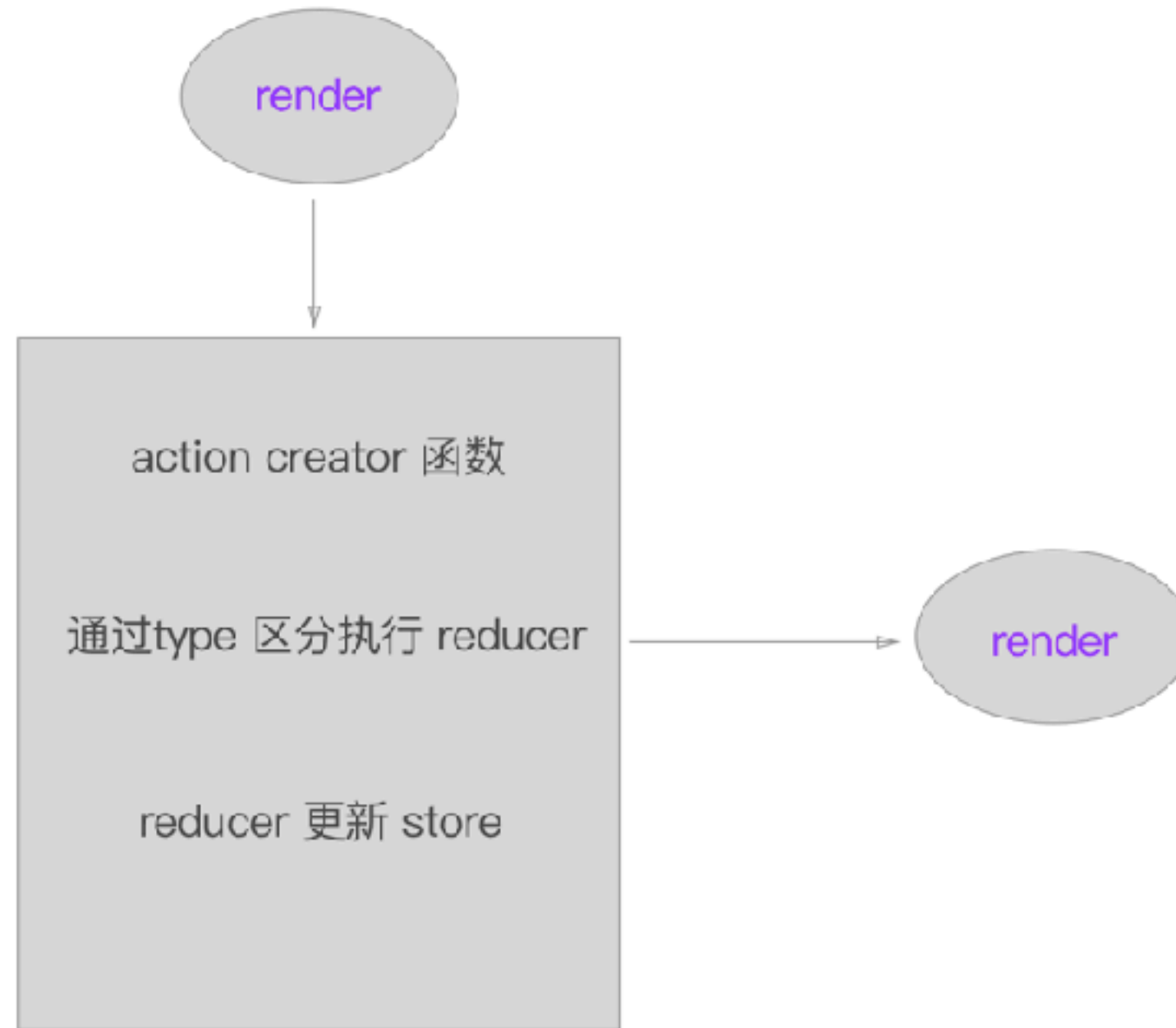
- 嵌套组件之间，父组件向子组件传递回调函数，子组件触发回调函数来更新父组件的数据
- 通过使用发布订阅的方式来通知数据更新
- Flux
- Redux

Redux

- Redux 是JavaScript 状态容器，提供可预测化的状态管理

原则

- 应用中所有的state 都以一个对象树的形式存储在一个单一的store中
- 唯一改变state的办法是触发action，一个描述发生什么的对象
- 为了描述action如何改变state树，需要编写reducers



Redux 执行逻辑

- 这是一个简单的目录结构

```
graph TD; actions --> A[业务逻辑]; common; components; containers; daily; feed; hot; images; post --> B[最终更新state]; reducers; store --> C[单一数据源]; styles; svg; user; app_js[app.js]; root_js[root.js]; util_js[util.js];
```

‣ actions 业务逻辑

‣ common

‣ components

‣ containers

‣ daily

‣ feed

‣ hot

‣ images

‣ post 最终更新state

‣ reducers

‣ store 单一数据源

‣ styles

‣ svg

‣ user

app.js

root.js

util.js

创建action

- 通过字符串type来表示将要执行的动作
- 通过dispatch把user数据传到reducer中
- 这里可以写通过网络传输获取数据的业务逻辑

```
'user strict';

import routes from '../routes';
import * as TYPES from './actionTypes';
import * as Storage from '../common/Storage';
import networkUtil from '../common/network';

export const userFromSync = (user) => {
  return (dispatch) => {
    dispatch({
      type: TYPES.USER_INFO_SYNC,
      user: user
    });
  }
};
```

创建reducer

- reducer是最终更新state的地方
- 会改变store中的值

```
import * as TYPES from '../actions/actionTypes';

const initialState = { ... };

let userReducer = (state = initialState, action) => {

  switch (action.type) {
    case TYPES.USER_LOGIN: ...
    case TYPES.USER_LOGIN_RECEIVED: ...
    case TYPES.USER_LOGOUT: ...
    case TYPES.USER_INFO_SYNC:
      return {
        ...state,
        ...action
      };
    case TYPES.USER_TOKEN: ...
    default:
      return state;
  }
};

export default userReducer;
```

创建store

store.js

×

⋮ || ↺

```
1  'use strict';
2
3  import { createStore, applyMiddleware } from 'redux';
4  import thunk from 'redux-thunk';
5  import rootReducer from '../reducers/rootReducer';
6
7  ⊕ const logger = store => next => action => { ...
12  }
13
14  ⊕ const crashReporter = store => next => action => { ...
27  }
28
29  // applyMiddleware 接收 createStore()
30  // 并返回一个包含兼容 API 的函数。
31  let createStoreWithMiddleware = applyMiddleware(thunk, logger, crashReporter)(createStore);
32  let store = createStoreWithMiddleware(rootReducer);
33
34  export default store;
```


热更新

- 热更新区别与我们去应用商店下载app安装更新
- 可以实时的更新和部署应用
- 可以加速app开发的迭代速度，方便活动上戏和bug调试
- 微软项目地址：<https://github.com/Microsoft/react-native-code-push>
- 不过现在应用商店审核周期已经缩短到24小时

怎么更新？

- 热更新服务器上有打包后的jsbundle文件
- 在应用启动位置配置热更新策略
- 如果有更新会自动下载
- 将服务器上的jsbundle文件替换本地文件
- jsbundle文件存放了rn源码，第三方库代码和业务逻辑代码

微信小程序对比

```
page {  
  background-color: #fbf9fe;  
  height: 100%;  
}  
.container {  
  display: flex;  
  flex-direction: column;  
  min-height: 100%;  
  justify-content: space-between;  
}
```

```
let styles = StyleSheet.create({  
  loginBox: {  
    flex: 1,  
    justifyContent: 'center',  
    alignItems: 'center',  
    backgroundColor: '#FFFFFF'  
  },  
  phoneNumber : {  
    marginTop: 35,  
    width: util.size.width - 100,  
    marginBottom: 20,  
    borderWidth: 1,  
    borderColor: '#E6E6FA',  
    fontSize: 14,  
    height: 40,  
    paddingLeft: 10  
  },  
  pwdNumber: {
```

```
<view class="page">  
  <view class="page_hd">  
    <text class="page_title">audio</text>  
    <text class="page_desc">音频</text>  
  </view>  
  <view class="page_bd">  
    <view class="section section_gap" style="text-align: center;">  
      <audio src="{{current.src}}" poster="{{current.poster}}" nam  
    </view>  
  </view>  
</view>
```

```
render() {  
  const { navigator } = this.props;  
  
  return (  
    <View style={styles.loginBox}>  
      <View style={styles.phoneNumberBox}>  
        <TextInput style={styles.phoneNumber}  
          placeholder={'手机号'}  
          onChangeText={({text}) => this.setState({phone: text})}/>  
      </View>  
      <View style={styles.phoneNumberBox}>  
        <TextInput style={styles.pwdNumber}  
          placeholder={'密码'}  
          password={true}  
          onChangeText={({text}) => this.setState({password: text})}  
        />  
      </View>  
      <TouchableOpacity onPress={() => this._login(this.state.phone,  
        <View style={styles.buttonBox}>  
          <Text style={styles.buttonText}>登录</Text>  
        </View>  
      </TouchableOpacity>  
    </View>  
  );  
}
```

React Native福利

- react native vip交流群
- 也有很多喜欢rn的实战分享群
- 此公众号上有免费的50集入门视频
- 还有收费的项目实战开发视频
- 报我名字有优惠!!!



“谢谢”

—刺杰