Dva

React 应用框架在蚂蚁金服的实践

陈成 (花名：云谦)

 杭州

 蚂蚁金服

 @sorrycc

 @chenchengpro

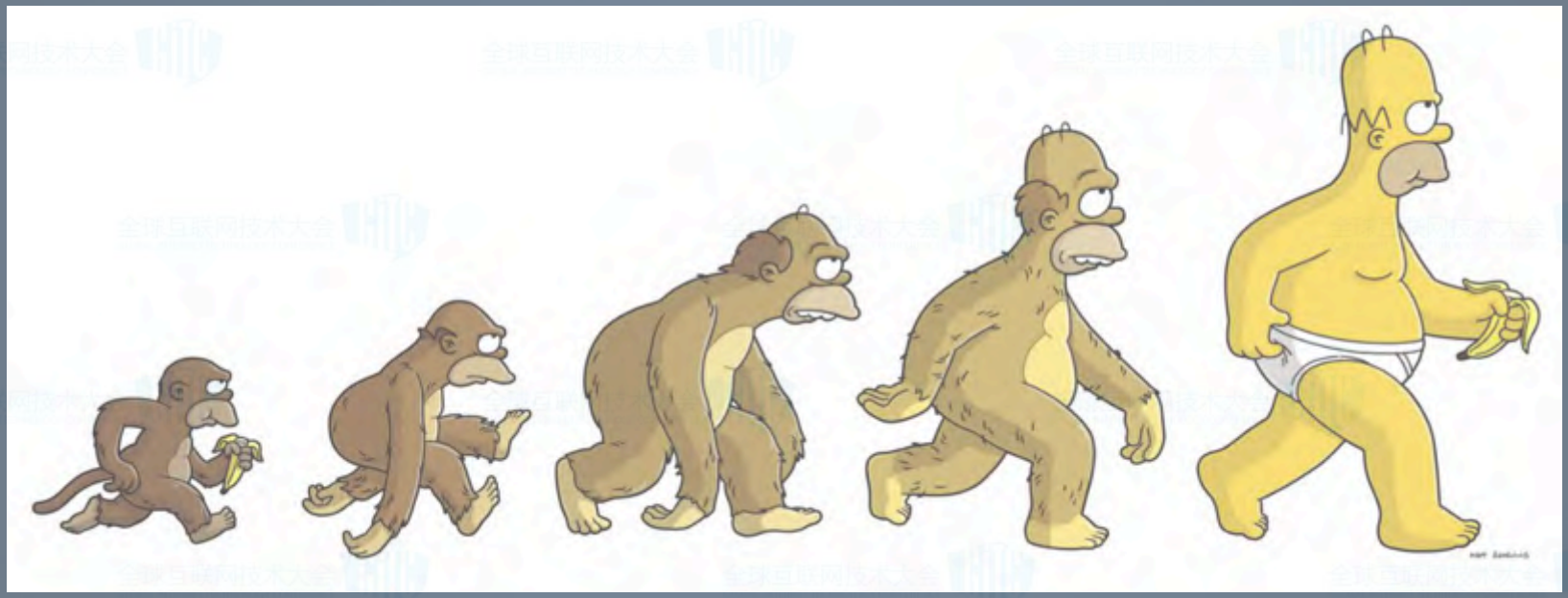
http://slides.com/sorrycc/dva

此次分享包含：

- 蚂蚁金服前端应用架构历程
- 介绍 Dva，基于 redux 的前端框架
- 我们的前端技术栈

Let's Get Started!

应用架构历程

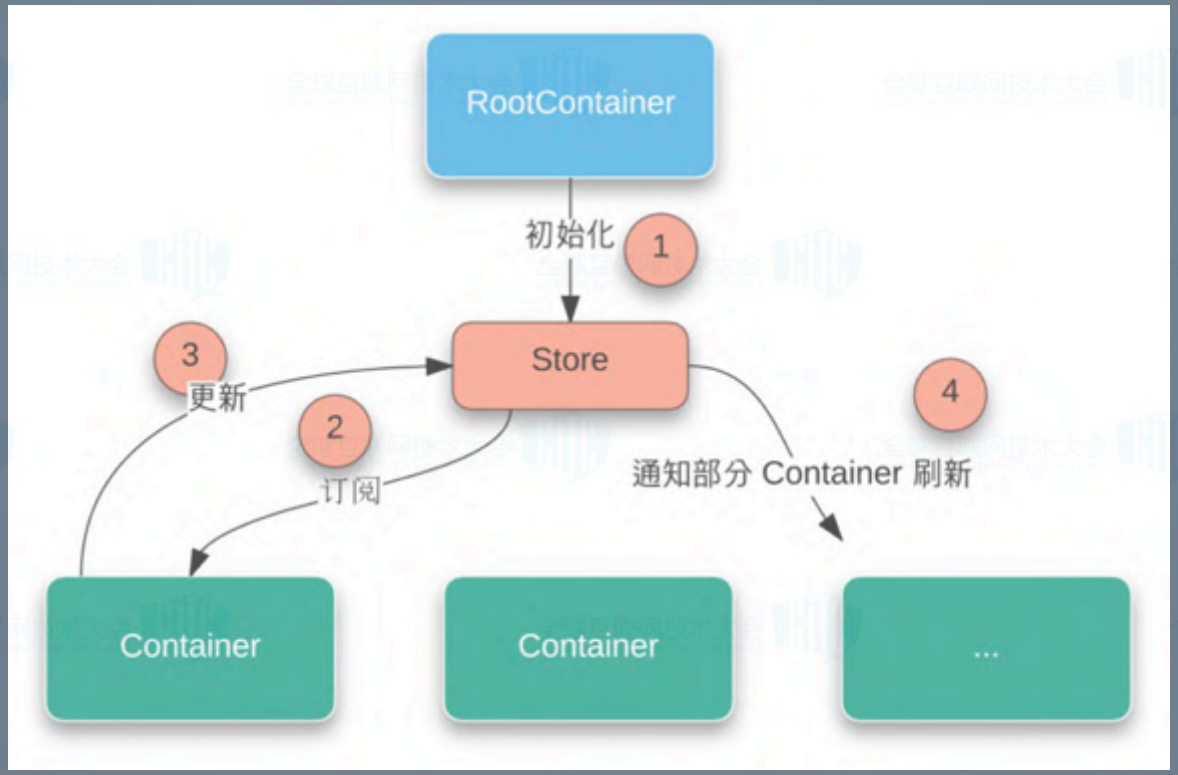


PlainReact Roof Roof@0.5 Redux Dva

Roof 0.4

- 第一个版本的应用架构
- 类似 baobab

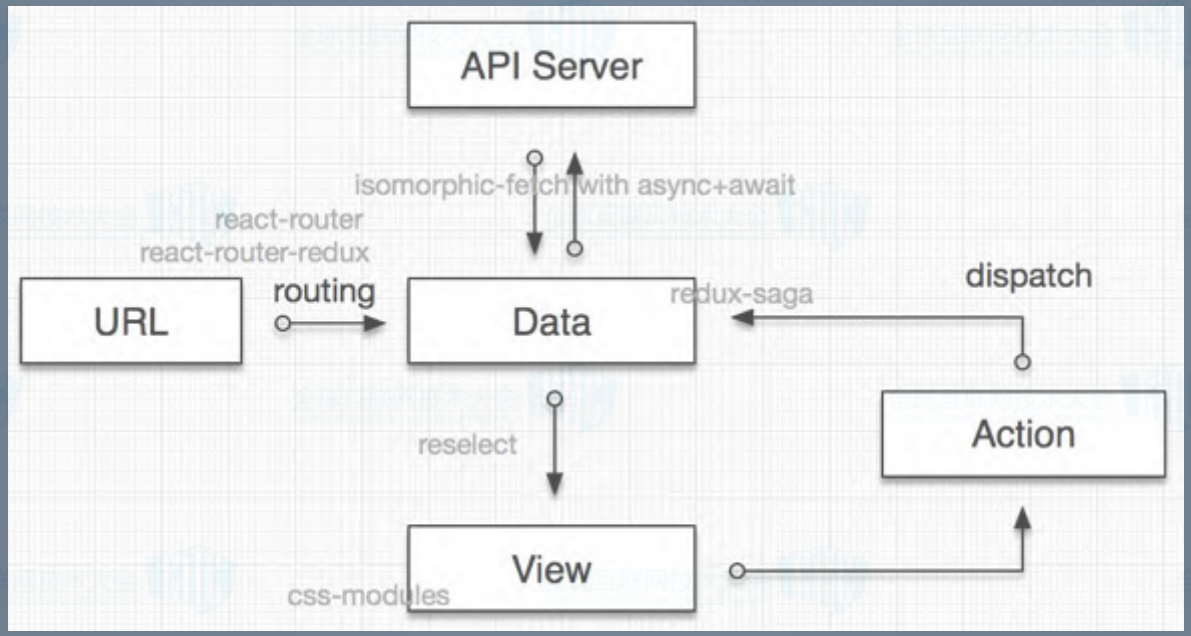
Roof 0.5


RootContainer
初始化
1
Store
3
更新
2
订阅
4
通知部分 Container 刷新
Container
Container
...

Redux

- 轻量级数据流方案，解决从 state 到 component 的单向数据流转问题
- 借助 Action 和 Reducer 实现可预测的 state 变更
- 社区活跃，丰富的扩展、调试方案
- ...

Redux 最佳实践



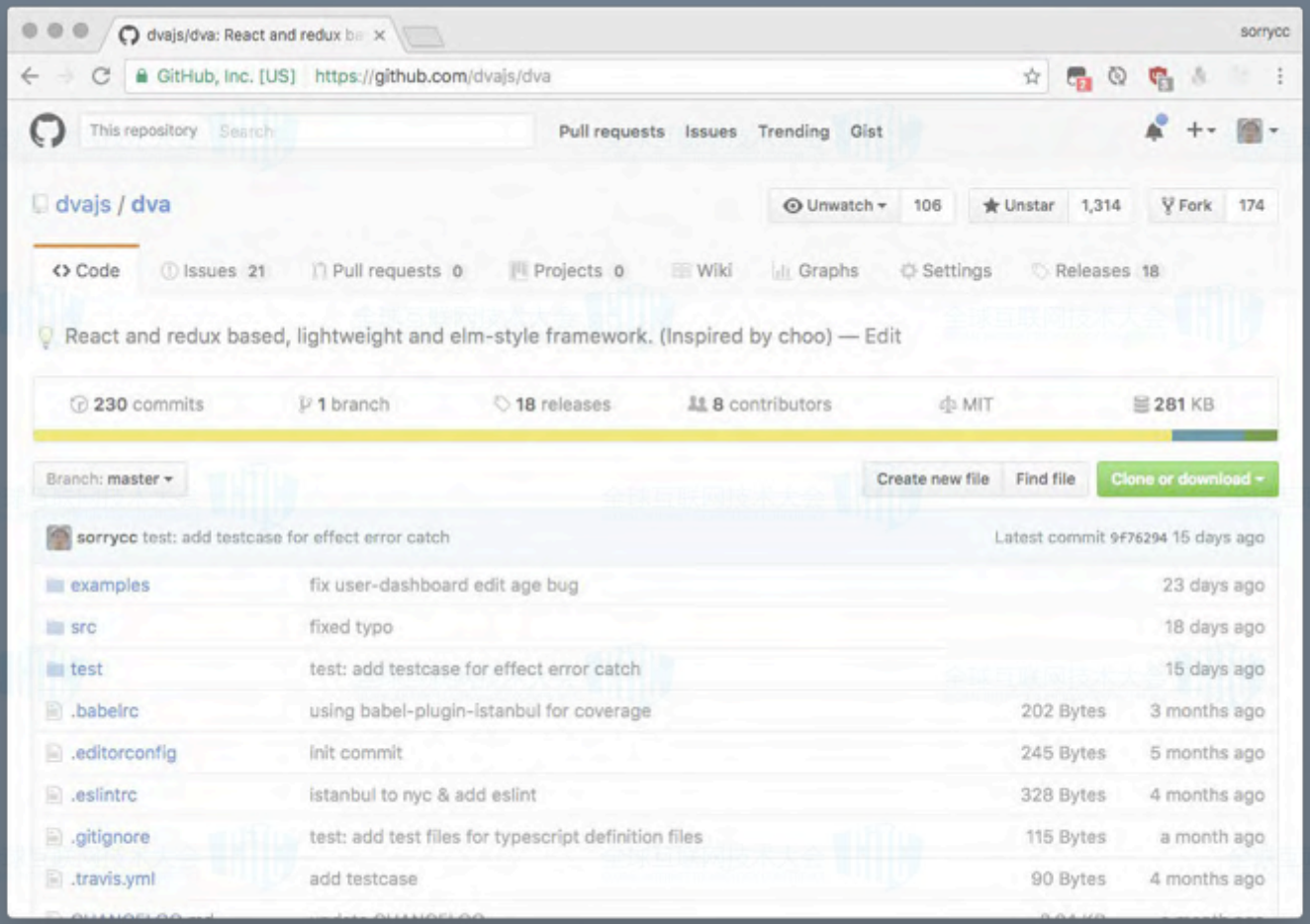
文：React + Redux 最佳实践

新的问题 😐

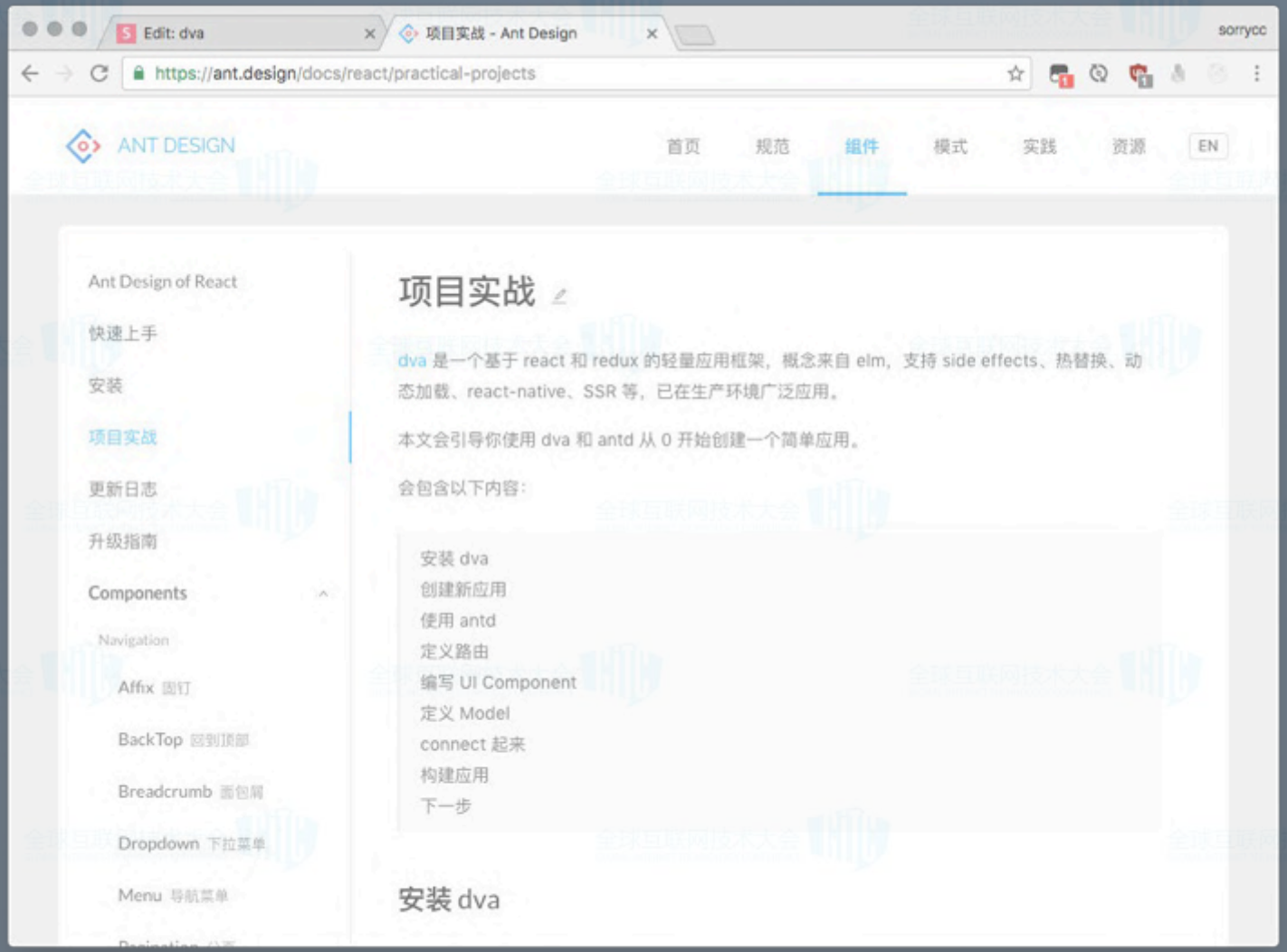
- 非常多的库供选择
- 代码写哪里？是个问题
- 代码分散，影响专注力
- 一遍遍重复地写 **showLoading** 和 **hideLoading**
- 出错处理太繁琐，每个异步 saga 都要 try .. catch
- 项目太大了，我需要动态加载方案
- ...

Dva

Build redux application easier and better.


dvajs/dva: React and redux
sorrycc
GitHub, Inc. [US] https://github.com/dvajs/dva
This repository Search
Pull requests Issues Trending Gist
dvajs / dva
Unwatch 106 Unstar 1,314 Fork 174
Code Issues 21 Pull requests 0 Projects 0 Wiki Graphs Settings Releases 18
React and redux based, lightweight and elm-style framework. (Inspired by choo) — Edit
230 commits 1 branch 18 releases 8 contributors MIT 281 KB
Branch: master
Create new file Find file Clone or download
sorrycc test: add testcase for effect error catch
Latest commit 9f76294 15 days ago
examples fix user-dashboard edit age bug 23 days ago
src fixed typo 18 days ago
test test: add testcase for effect error catch 15 days ago
.babelrc using babel-plugin-istanbul for coverage 202 Bytes 3 months ago
.editorconfig init commit 245 Bytes 5 months ago
.eslintrc istanbul to nyc & add eslint 328 Bytes 4 months ago
.gitignore test: add test files for typescript definition files 115 Bytes a month ago
.travis.yml add testcase 90 Bytes 4 months ago


Dan Abramov
@dan_abramov
Follow
Cool to see some higher level libraries built on top of Redux ecosystem
dvajs/dva
dva - :bulb: React and redux based, lightweight and elm-style framework. (Inspired by choo)
github.com
RETWEETS
9
LIKES
40
10:16 PM - 15 Sep 2016
9
40


Edit: dva
项目实战 - Ant Design
sorrycc
https://ant.design/docs/react/practical-projects
ANT DESIGN
首页
规范
组件
模式
实践
资源
EN
Ant Design of React
快速上手
安装
项目实战
更新日志
升级指南
Components
Navigation
Affix 固钉
BackTop 回到顶部
Breadcrumb 面包屑
Dropdown 下拉菜单
Menu 导航菜单
项目实战
dva 是一个基于 react 和 redux 的轻量应用框架，概念来自 elm，支持 side effects、热替换、动态加载、react-native、SSR 等，已在生产环境广泛应用。
本文会引导你使用 dva 和 antd 从 0 开始创建一个简单应用。
会包含以下内容：
安装 dva
创建新应用
使用 antd
定义路由
编写 UI Component
定义 Model
connect 起来
构建应用
下一步
安装 dva

Dva @ 蚂蚁金服


react 应用中，各应用架构分布：
plain react
222
51%
roof 0.4
7
2%
roof 0.5.0 - 0.5.4
10
2%
roof 0.5.5 - 0.5.6
69
16%
redux
74
17%
dva
56
13%
react
438

Dva 是什么

- **框架**，而非类库
- 基于 redux, react-router, redux-saga 的轻量级封装
- 借鉴 elm 的概念，Reducer, Effect 和 Subscription
- ...

特性

- 仅有 5 个 API
- 支持 HMR
- 支持 SSR (ServerSideRender)
- 支持 Mobile/ReactNative
- 支持 TypeScript
- 支持路由和 Model 的动态加载
- 完善的语法分析库 dva-ast
- ...

初印象 

```
import dva from'dva';

// 1. 创建 dva 实例
const app = dva();

// 2. 装载插件 (可选)
app.use(require('dva-loading')());

// 3. 注册 Model
app.model(require('./models/count'));

// 4. 配置路由
app.router(require('./router'));

// 5. 启动应用
app.start('#root');
```

DEMO

https://jsfiddle.net/puftw0ea/3/embedded/js,html,css,result/

5 个 API

- app = dva(Opts)
- app.use(Hooks)
- app.model(ModelObject)
- app.router(Function)
- app.start([HTMLElement], opts)

8 个概念

- State
- Action
- Model
  - Reducer
  - Effect
  - Subscription
- Router
  - RouteComponent

State

- State 表示应用的数据层，由 model 的 state 组成全局的 state

全局 State

```
{
  todos:[],
  visibilityFilter:'SHOW_ALL',
}
```

Model

```
app.model({namespace:'todos', state: []});
app.model({namespace:'visibilityFilter', state: 'SHOW_ALL'});
```

Action

- Action 表示操作事件，可以是同步，也可以是异步

格式

```
{
  type: String,
  payload: Any?,
  error? Error,
}
```

触发 action

```
dispatch(Action);
dispatch({ type: 'todos/add', payload: 'Learn Dva' });
```

Model

- Model 非 MVC 中的 M，而是领域模型，用于把数据相关的逻辑聚合到一起
- 包含 5 个 key
  - namespace
  - state
  - reducers
  - effects
  - subscriptions

Model 例子

```
export default {
  namespace: 'todos',
  state: [],

  reducers: {
    // 保存 todos
    save(state, { payload }) {
      return payload;
    },

    // 添加 todo
    add(state, { payload }) {
      return [...state, payload];
    },
  },

  effects: {
    // 异步获取 todos，然后通过 action 保存
    *fetch(action, { call, put }) {
      const todos = yield call(service.fetchTodos);
      yield put({ type: 'save', payload: todos });
    },
  },
}
```

Reducer

- Reducer 是唯一可以修改 state 的地方，接收 state 和 action，返回新的 state 。

格式

```
(state, action) => newState
```

例子

```
// good
function(state, action) {
  return state + action.payload;
}

// bad
const foo = 1;
function(state, action) {
  return state + action.payload + foo;
}
```

Effect

- `Effect` 用于处理异步逻辑，基于 `redux-saga` 实现
- 基于 Generator

格式

```
*(action, effects) => void
```

例子

```
*fetch(action, { call, put, select }) {
  const todos = yield call(fetchTodos);
  yield put({ type: 'save', payload: todos });
}
```

Subscription

- Subscription 表示订阅，用于订阅一个数据源，然后按需

格式

```
({ history, dispatch }) => Function|void
```

例子

```
function({ dispatch, history }) {
  history.listen(({ pathname }) => {
    if (pathname === '/users') {
      dispatch({
        type: 'users/fetch',
      });
    }
  });
},
```

Router

- Router 表示路由配置信息

格式

```
({ history, app }) => JSXElement | Object
```

例子

```
export default function({ history }){
  return(
    <Router history={history}>
      <Route path="/" component={App} />
    </Router>
  );
}
```

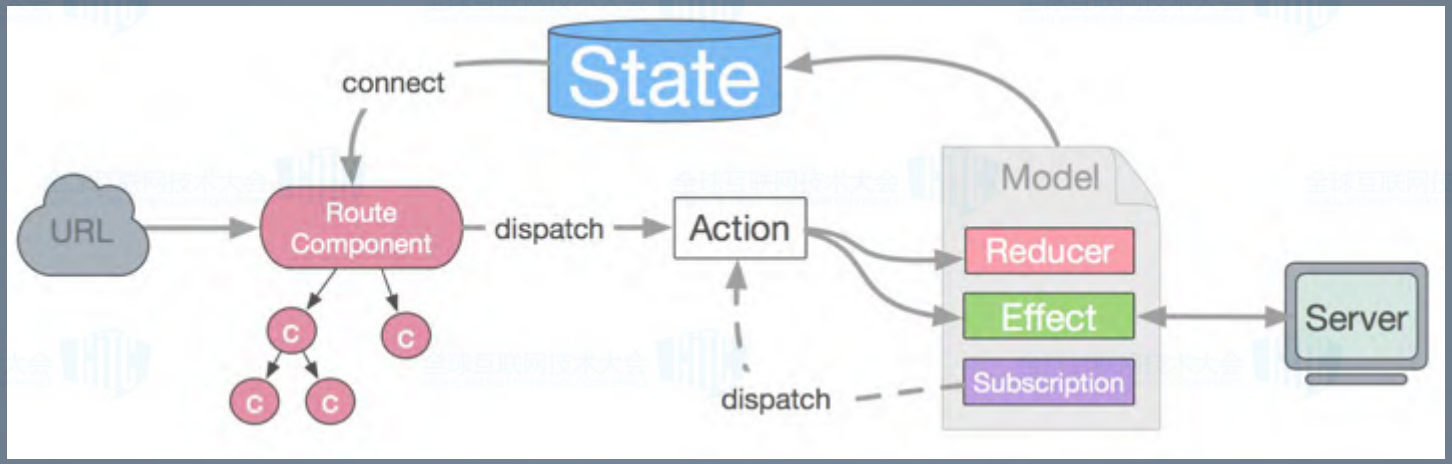
RouteComponent

- RouteComponent 表示 Router 里匹配路径的 Component，通常会绑定 model 的数据

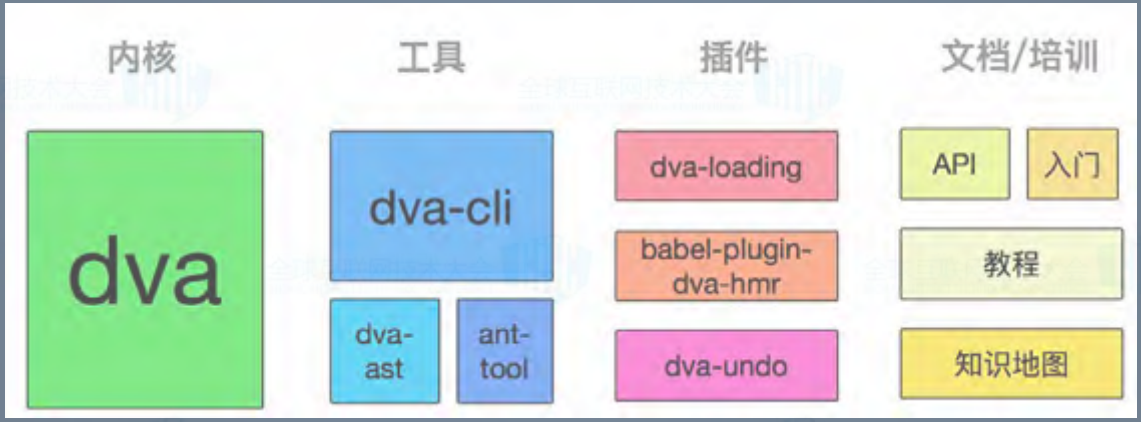
例子

```
import { connect } from 'dva';

function App() {
  return <div>App</div>;
}

function mapStateToProps(state) {
  return { todos: state.todos };
}

export default connect(mapStateToProps)(App);
```

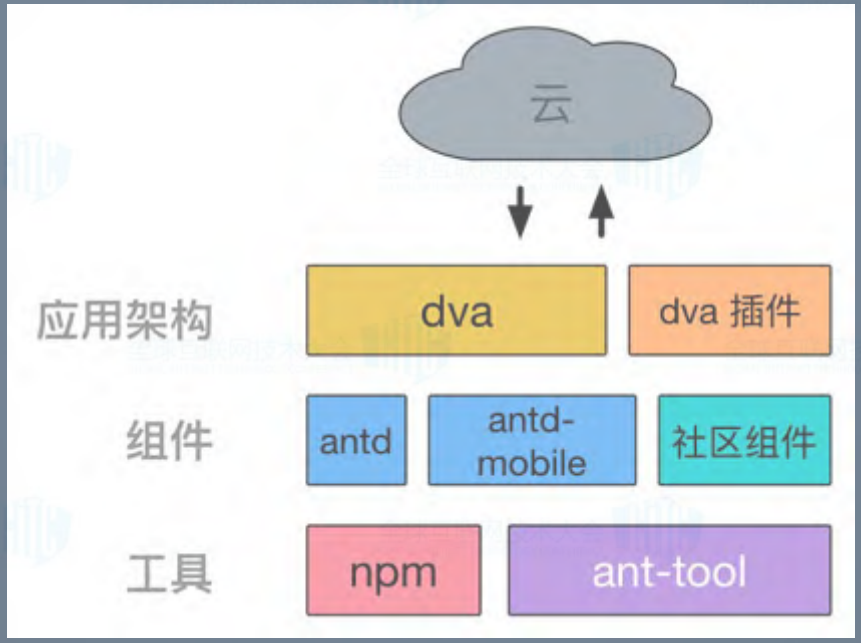
整体架构


State
connect
URL
Route
Component
dispatch
Action
Model
Reducer
Effect
Subscription
Server
dispatch
C
C
C
C

dva 生态


内核
工具
插件
文档/培训
dva
dva-cli
dva-ast
ant-tool
dva-loading
babel-plugin-dva-hmr
dva-undo
API
入门
教程
知识地图

蚂蚁金服前端技术栈



相关资源

- 蚂蚁金服前端应用架构的发展和选择
- dva 官网
- awesome-dva - dva 资源列表
- dva-knowledge - 包含使用 dva 所需的所有知识点
- 视频教程 - 一步步创建一个应用

Thank you!

@chenchengpro