

Yo: 聪明你的移动前端开发

徐潇潇@qunar.com

Yo是什么？

聪明你的 移动 前端开发

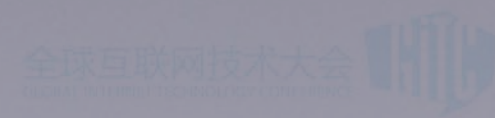
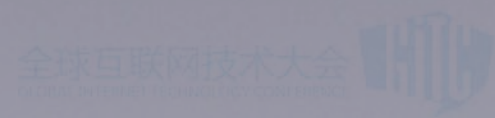
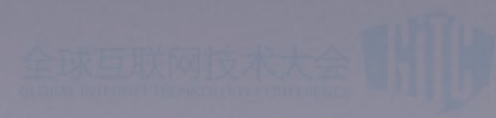
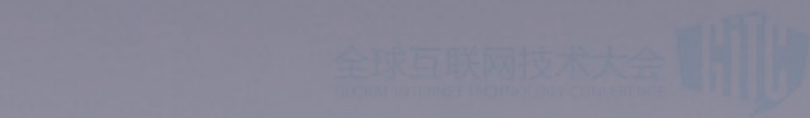
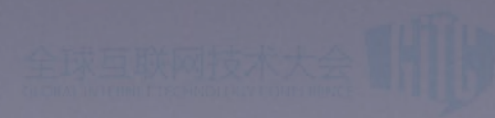
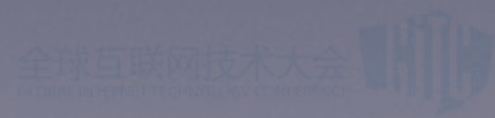
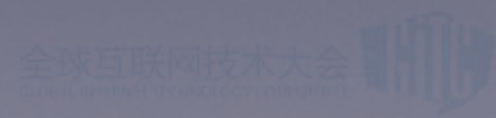
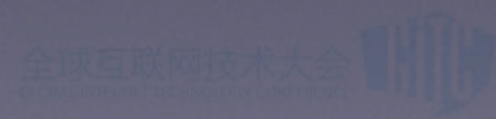
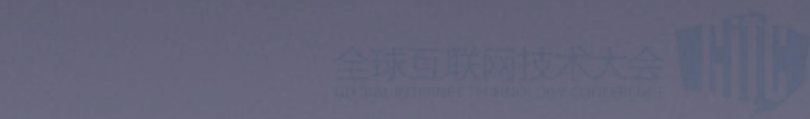
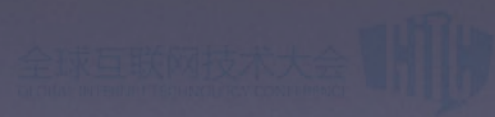
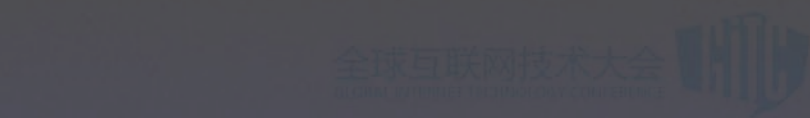
聪明的 移动端 UI 框架

• 背景

• Yo 的设计思路

• Yo 的特性

• Yo 的展望



背景

现状

- 近 100 个项目
- 易用性差、可维护性差、可扩展性差、性能不理想



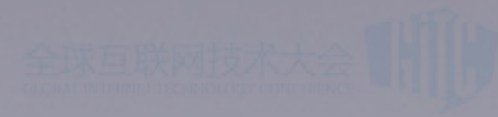
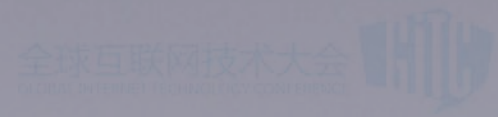
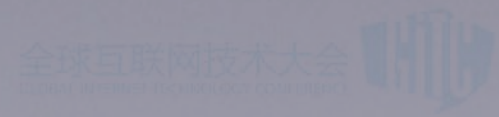
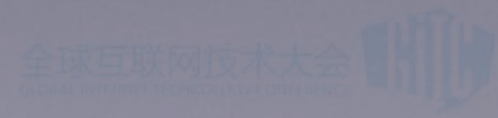
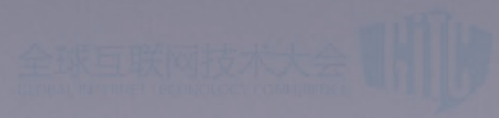
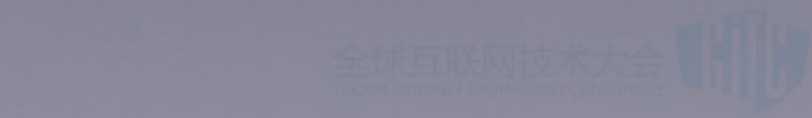
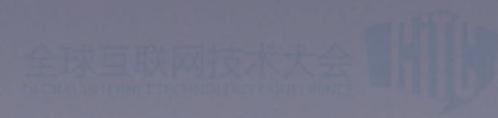
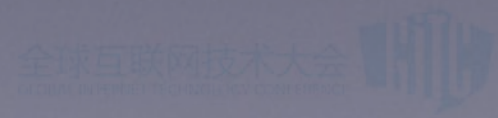
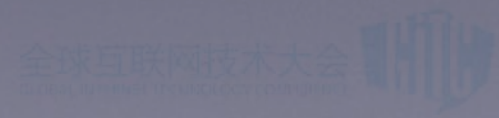
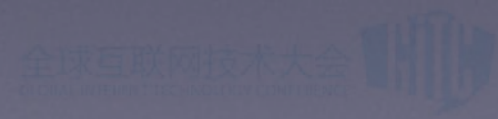
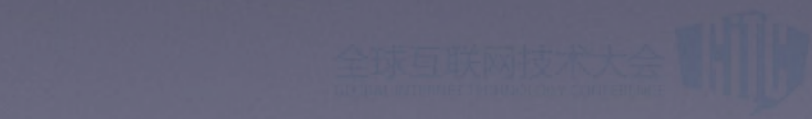
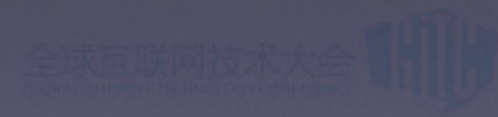
需求

- 交互体验好、体积小、性能好
- 简单易用、组件丰富、扩展性好



大而全？ 小而美？

Yo 的设计思路



用户体验

Mobile First

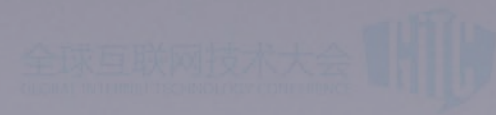
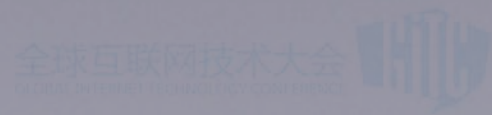
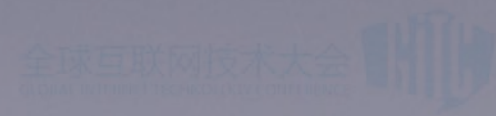
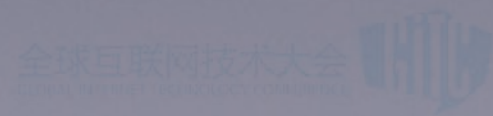
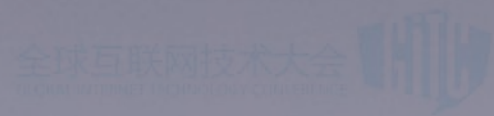
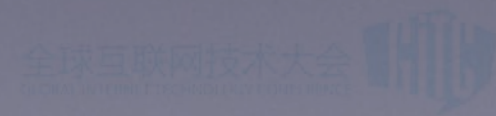
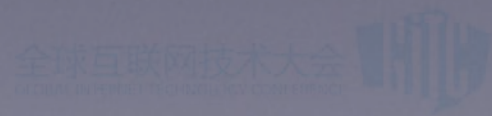
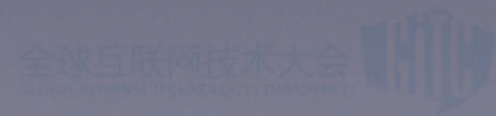
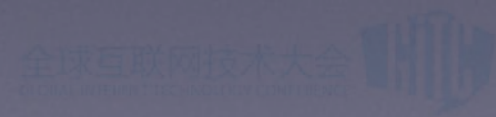
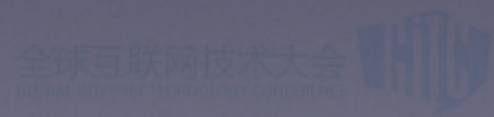
- 甩掉包袱，拥抱新特性
- 交互方式统一
- 更轻量



iOS 风格

- iOS 用户群体大
- 安卓碎片化太严重
- 普适性





开发体验

组件化

- CSS 提供可扩展的样式碎片
 - CSS 扩展：组件在样式层面的扩展
- JS 提供功能丰富的组件
 - JS 扩展：重新组合样式碎片和组件



+



按需加载

- 丰富的组件意味着框架体积不会小
- JS：按需引用
- CSS：保留核心功能和扩展能力

分层设计

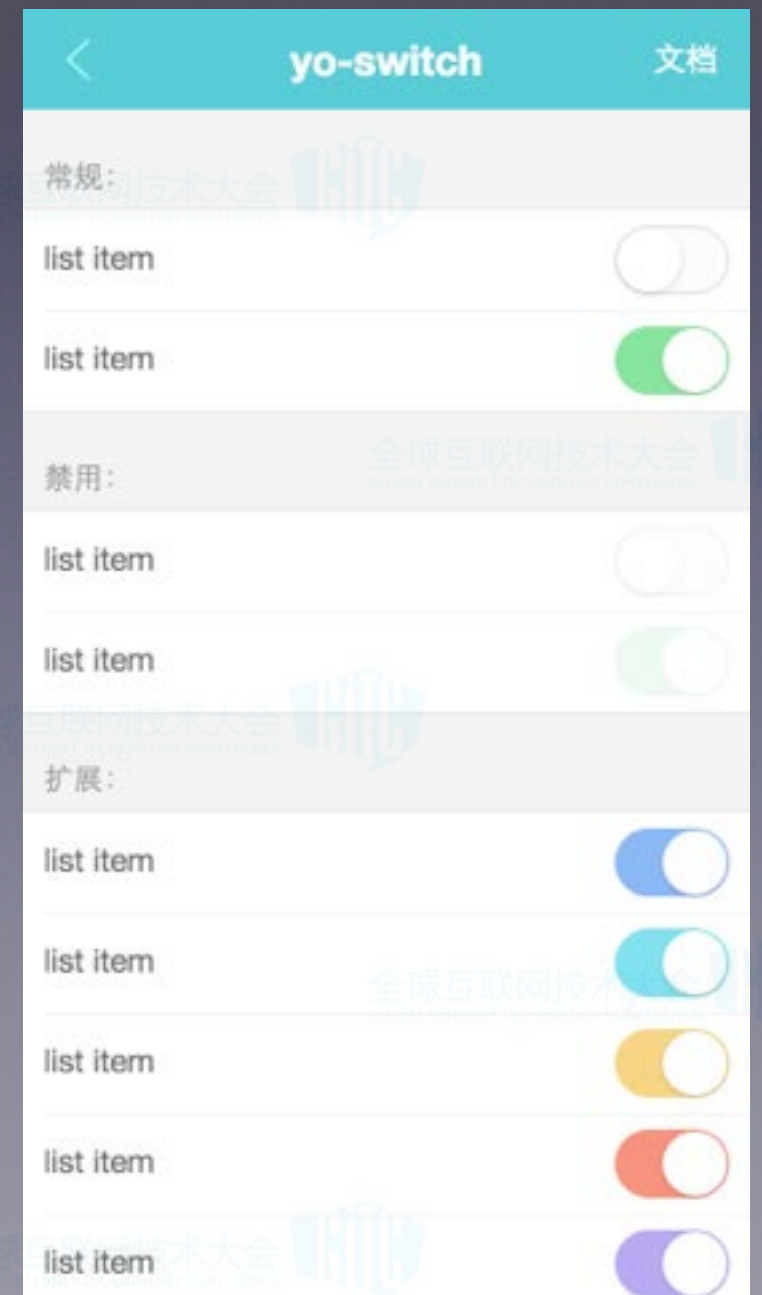


分层的意义

- 高度解耦和复用
- 避免混乱的依赖关系
- 代码结构和逻辑更加清晰

举个栗子

- Switch组件
- CSS 扩展：不同的颜色
- JS 扩展：拼成列表



目录结构

▼ style

▶ ani

▼ component

actionsheet.scss

alert.scss

calendar.scss

carousel.scss

confirm.scss

datetimepicker.scss

▶ core

▶ element

▶ fragment

▶ layout

▼ component

▶ actionsheet

▶ alert

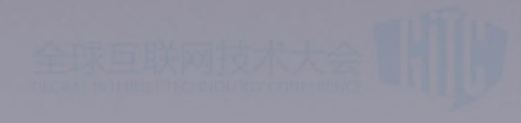
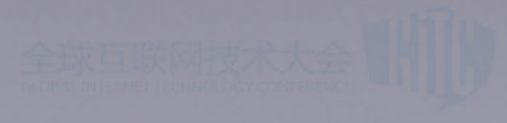
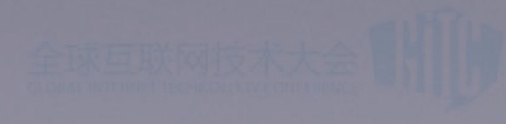
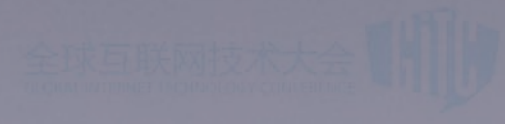
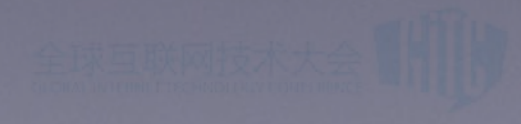
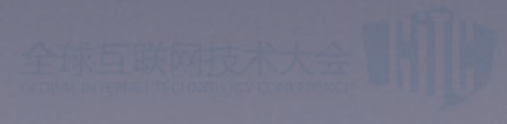
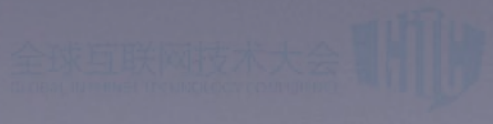
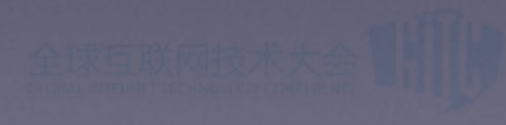
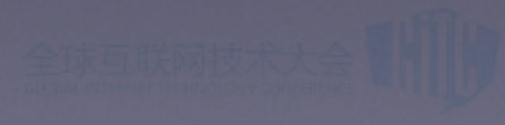
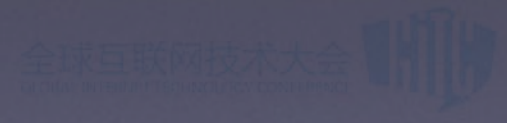
▶ calendar

▶ carousel

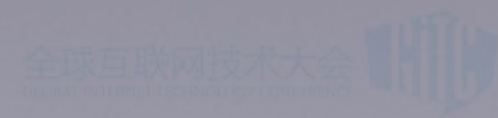
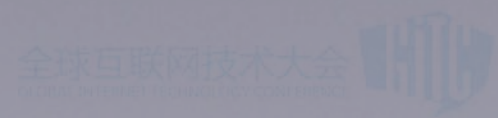
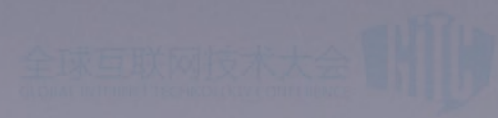
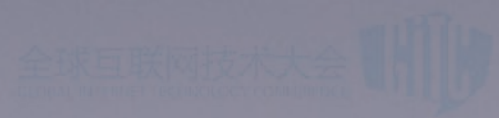
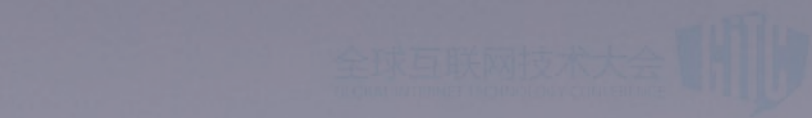
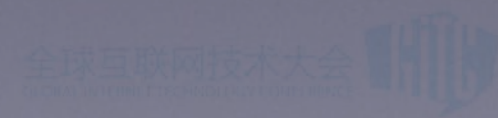
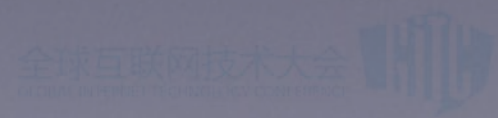
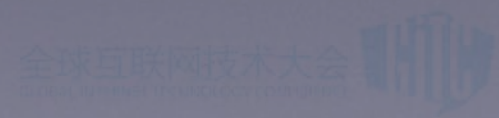
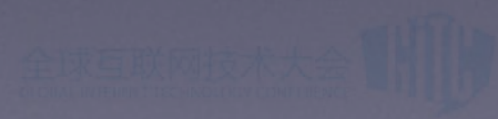
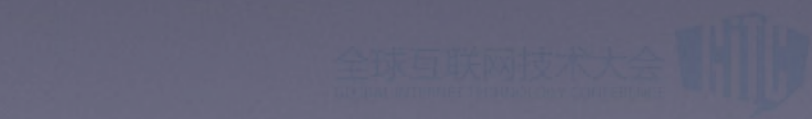
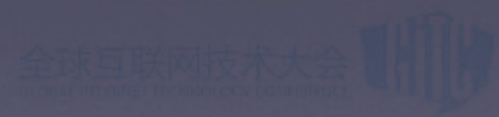
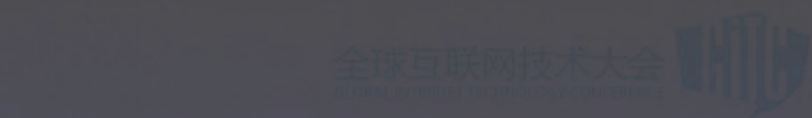
▶ common

▶ confirm

▶ datetimepicker



Yo 的特性



易用性

易用性

- JS 中直接引用 CSS，依赖清晰

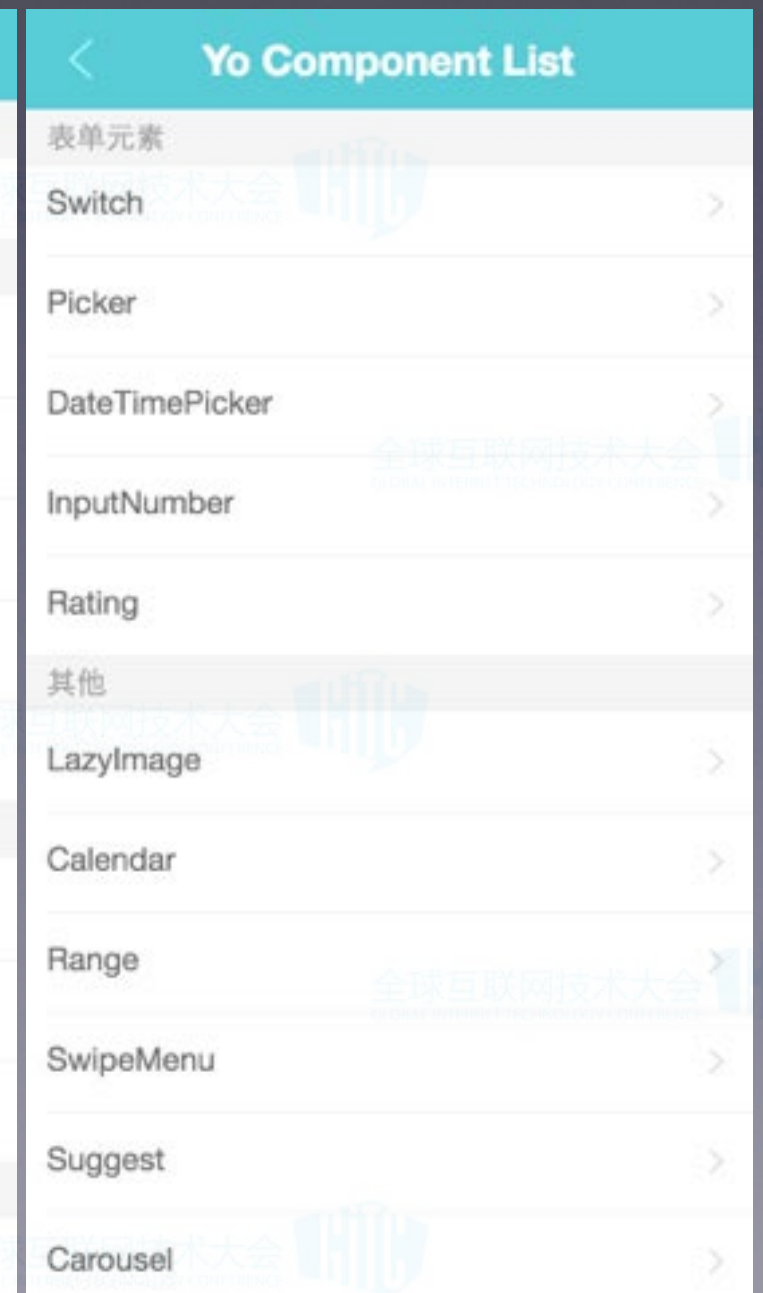
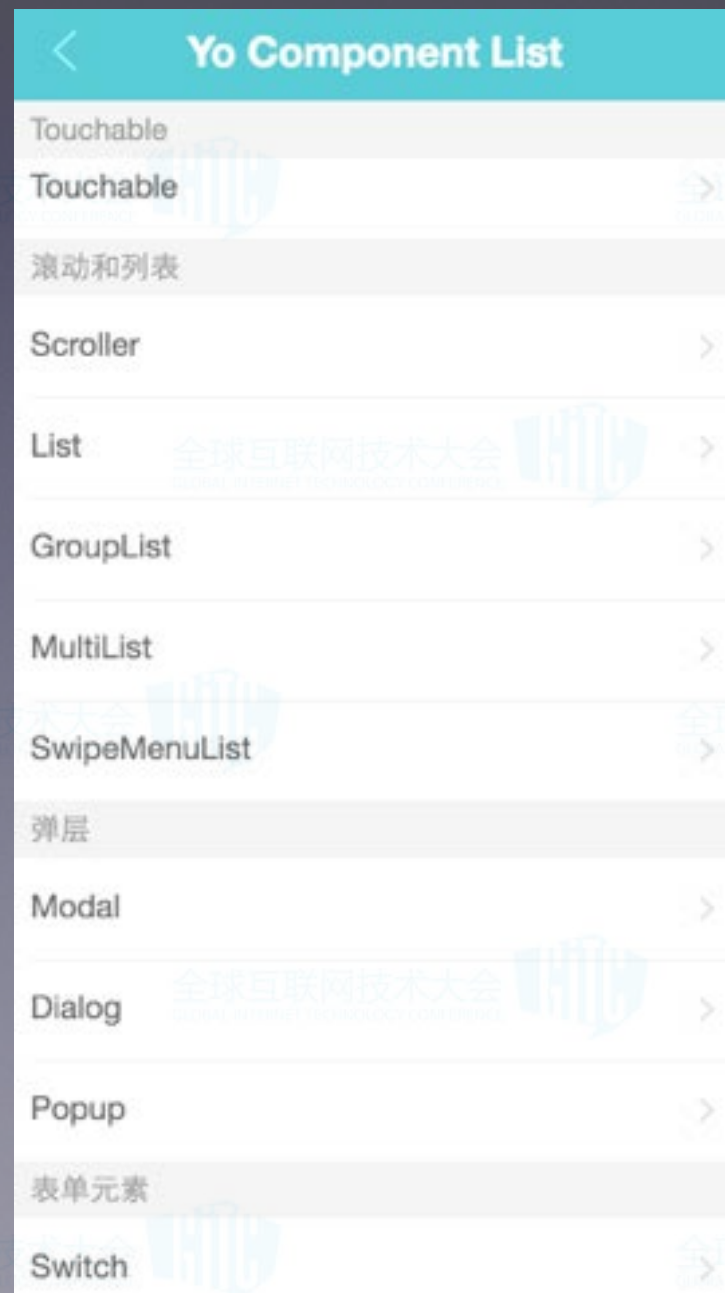
```
// ----- index.js -----  
import List from '$component/list';  
import './index.scss';
```

```
// ----- list.js -----  
import './list.scss';
```

```
// ----- list.scss -----  
@charset "utf-8";  
@import "../scroller";  
@import "../../fragment/yo-list";
```

易用性

- 组件丰富



易用性

- 丰富的 CSS 常用解决方案：
 - 形状方法、文本方法等
 - 1px 边框
 - iconfont

易用性

- 文档详尽
- 说明
- Demo
- API

Switch 3.0.0

开关组件，在checkbox基础上封装，具有打开关闭以及过程动画。

- 支持设置禁用
- 可配合样式扩展自定义样式，对颜色的修改请传入activeColor&defaultColor

使用说明

基本用法

展示一个给用户评分的组件，`switch1` 属性为当前的评分。由于这是一个受控的组件，所以需要通过 `onChange` 回调来设置当前值。

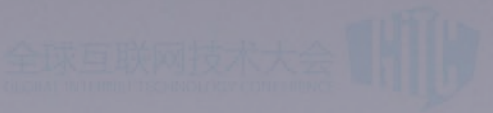
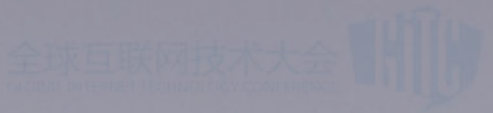
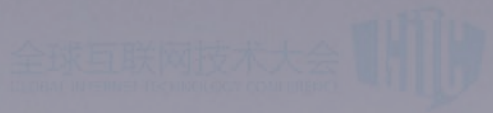
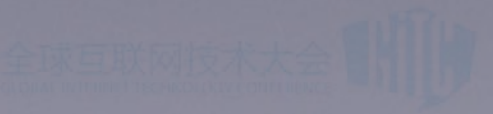
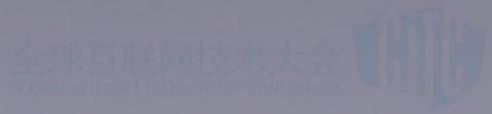
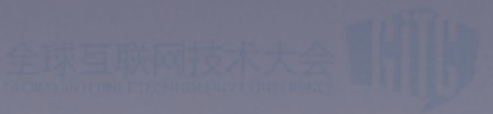
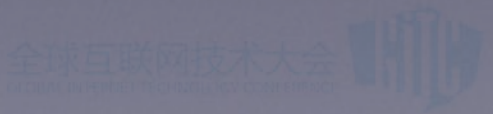
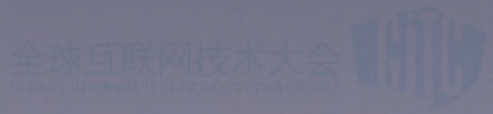
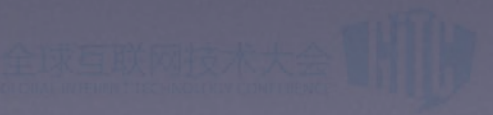
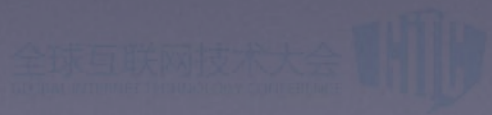
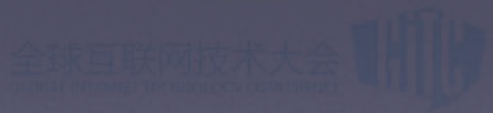
```
<Switch
  checked={this.state.switch1}
  onChange={(value) => { this.getValue('switch1', value); }}
/>
```

禁用组件

展示一个处于不可操作状态的 `switch` 组件 `checked` 表示当前的组件的状态。

```
<Switch disabled checked={false} />
<Switch disabled />
```





可扩展性

可扩展性

- CSS 可扩展性
- 全局配置 `yo-switch`
- 特定扩展 `yo-switch-light`
- 超扩展

yo-switch

描述: 构造yo-switch的自定义使用方法

版本: 3.0.0

源码位置: [./style/lib/element/yo-switch.scss:65](#)

Demo地址: [点击访问](#)

参数名	类型	描述	支持版本
<code>\$name</code>	String	定义扩展名称	3.0.0
<code>\$on-bgcolor</code>	Color	定义switch激活背景色	3.0.0
<code>\$bgcolor</code>	Color	定义switch背景色	3.0.0
<code>\$follow-bgcolor</code>	Color	定义switch跟随背景色	3.0.0
<code>\$thumb-color</code>	Color	定义switch滑块色	3.0.0

可扩展性

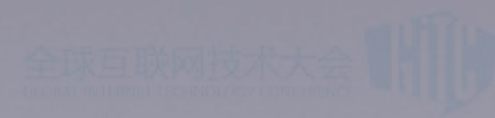
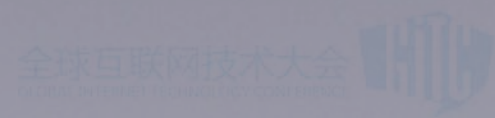
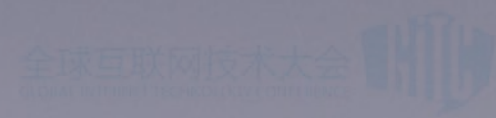
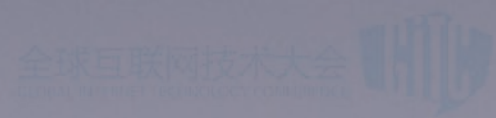
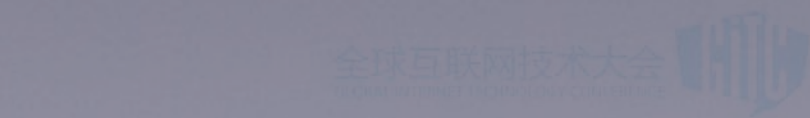
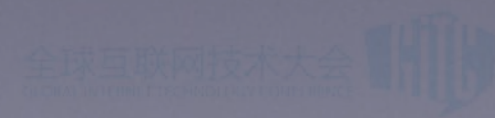
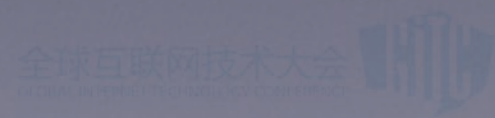
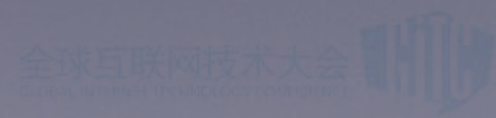
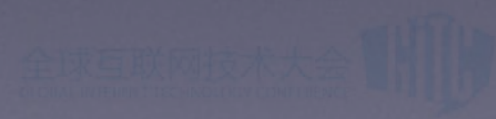
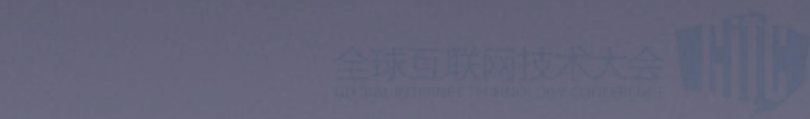
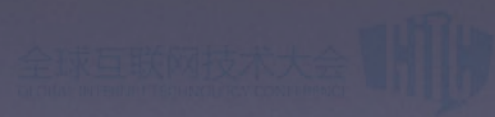
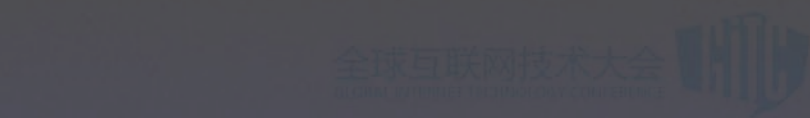
- 换肤

- 核心层提供了统一配置的能力
- 通过变量的管理来实现不同的配置

可扩展性

- JS 可扩展性

```
<Popup extraClass="yo-popup-operator">  
  <Switch extraClass="yo-switch-light"/>  
  <Switch extraClass="yo-switch-dark"/>  
</Popup>
```

用户体验

用户体验

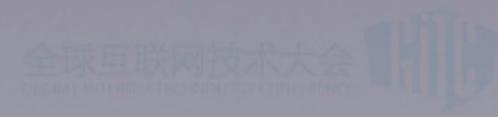
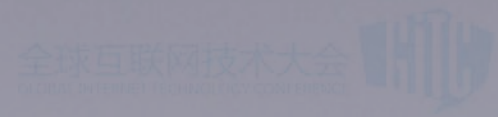
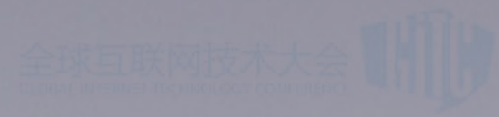
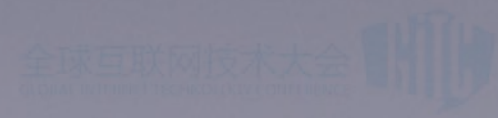
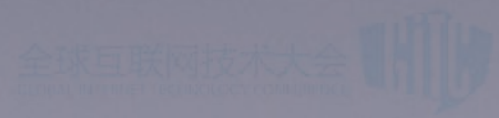
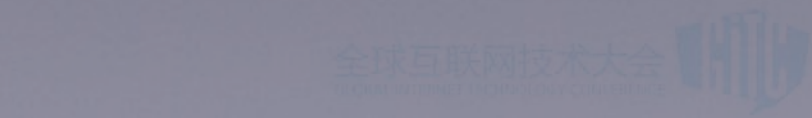
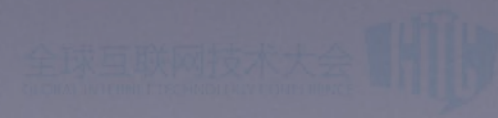
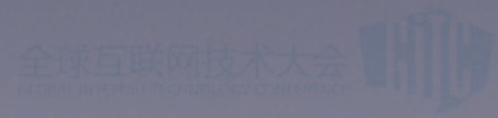
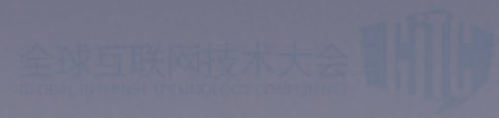
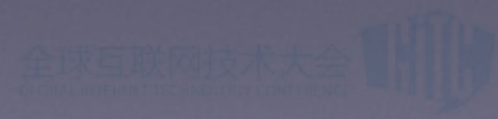
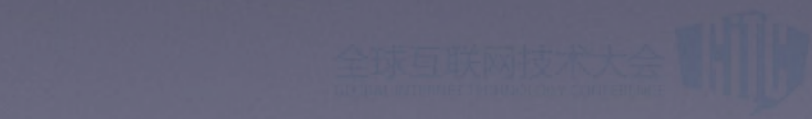
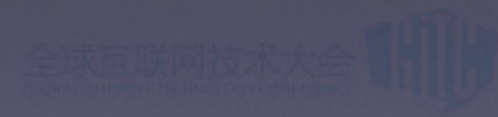
- Touchable
- 绑定点击反馈
- 适配滚动容器



用户体验

- LazyImage
- 延迟加载图片
- 适配滚动容器



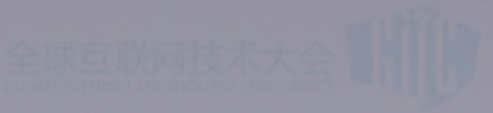
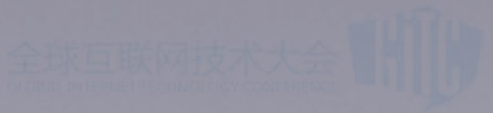
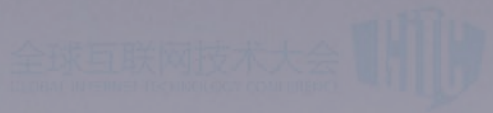
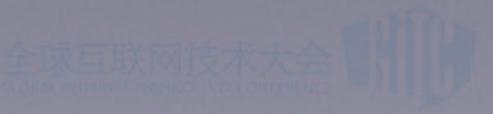
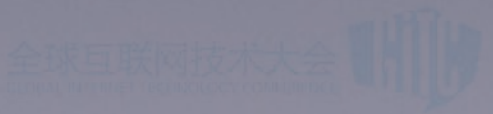
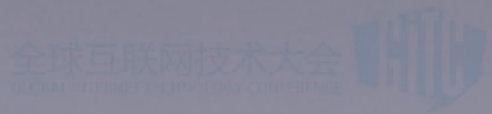
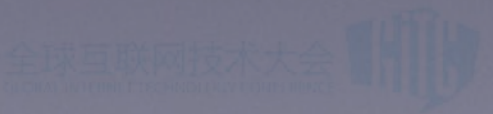
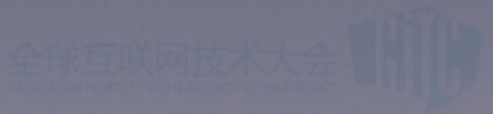
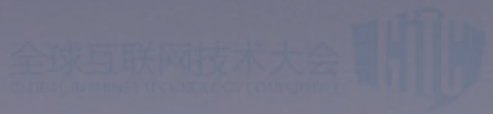
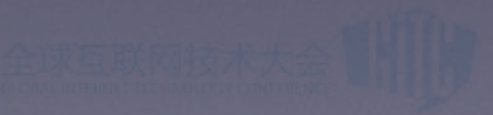
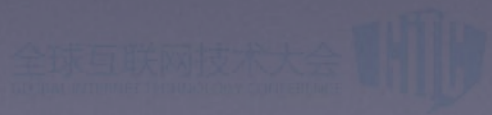
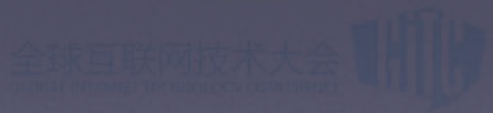


性能

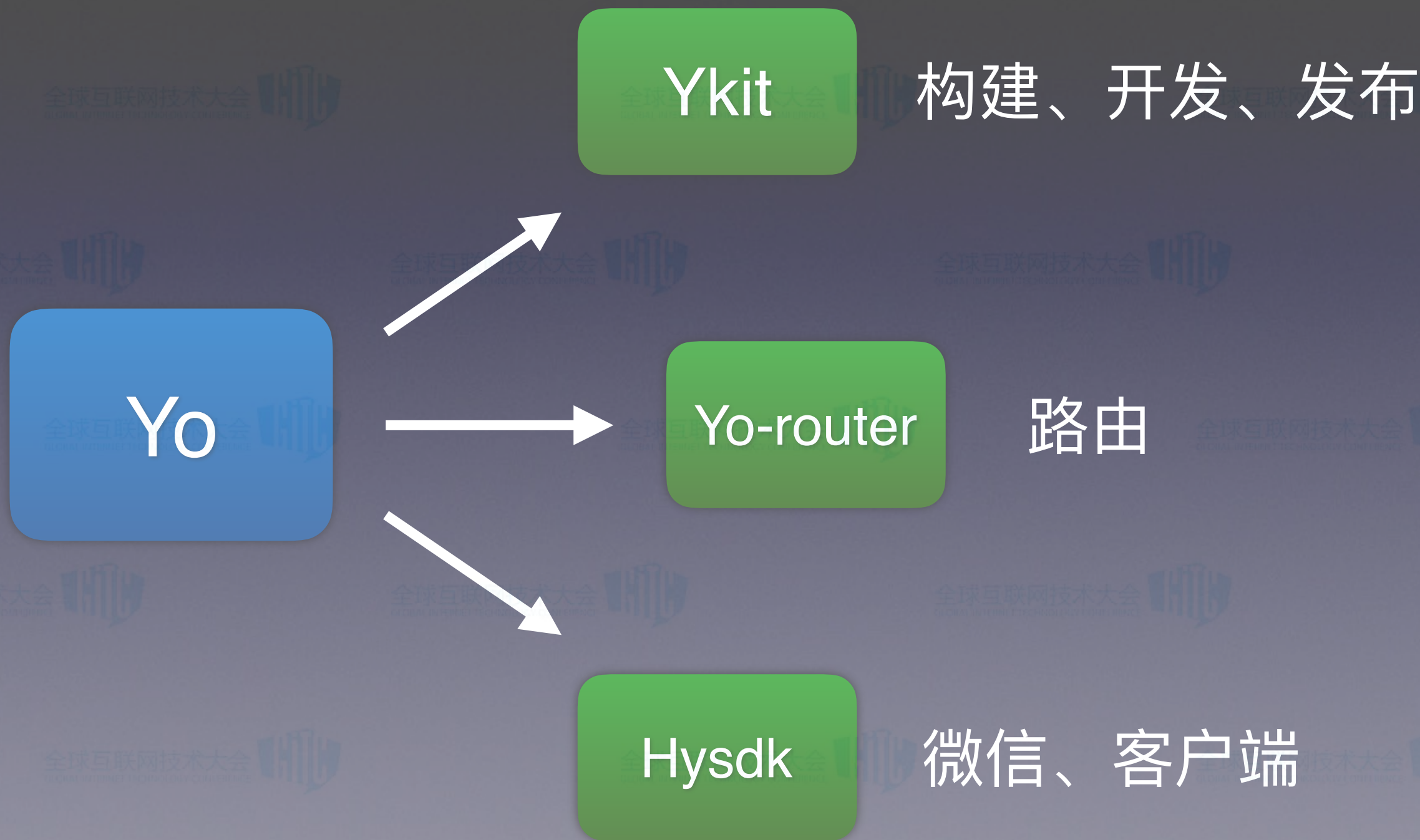
列表性能

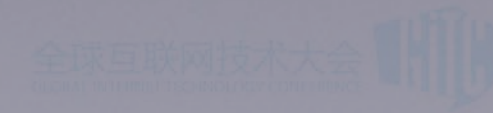
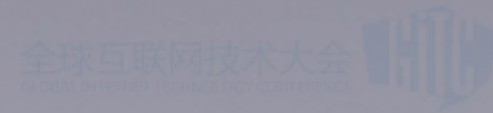
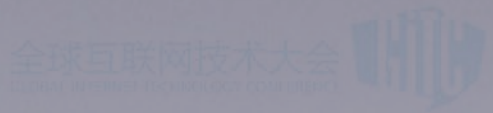
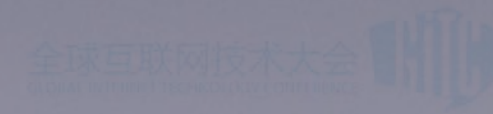
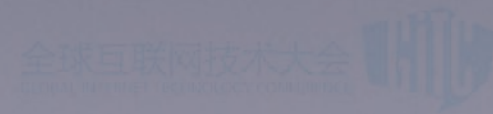
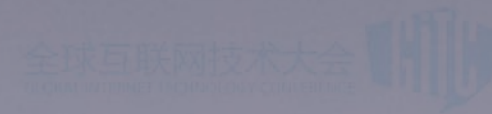
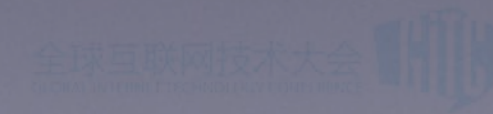
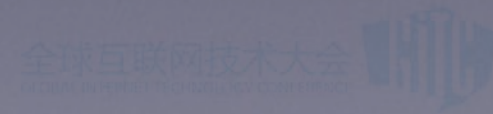
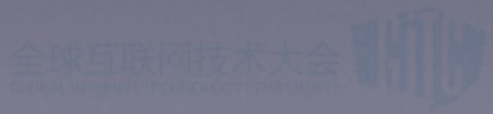
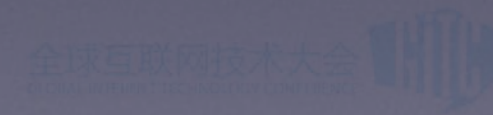
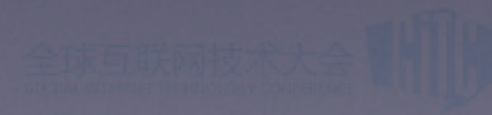
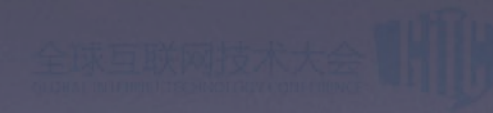
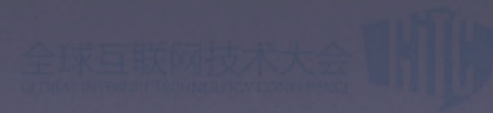
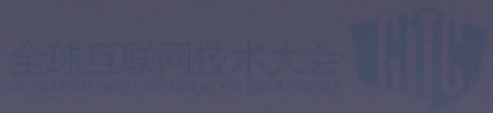
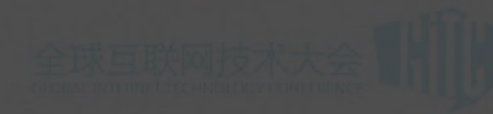
- 无限滚动列表
- 定高 & 不定高
- 扩展到其他组件





生态链





Yo 的展望

Yo 带来了什么?

- 高效、统一的开发方式
- 流畅的用户体验

Yo 的未来

- 逐步替代公司内部现有框架
- 开源并持续维护和改进
- 尝试引入一些更新更有挑战的东西

Thank You!