

○ 数据驱动前端开发

APICloud 前端技术负责人

申成波



AI时代的移动技术革新

Era of AI: Innovations in Mobile Technologies



+1

-1

{ "count" : 0 }





什么是数据？

科学实验、检验、统计等所获得的和用于科学研究、技术设计、查证、决策等的数值



数据的来源

· API接口调用

· Websocket

· Memory

· Push推送

· Stream数据流



前端的数据

```
{
  "id": "123",
  "author": {
    "id": "1",
    "name": "Paul"
  },
  "title": "My awesome blog post",
  "comments": [
    {
      "id": "324",
      "commenter": {
        "id": "2",
        "name": "Nicole"
      }
    }
  ]
}
```



后端的数据 (normalizr)

```
{
  "articles": {
    "123": {
      id: "123",
      author: "1",
      title: "My awesome blog post",
      comments: [ "324" ]
    }
  },
  "users": {
    "1": { "id": "1", "name": "Paul" },
    "2": { "id": "2", "name": "Nicole" }
  },
  "comments": {
    "324": { id: "324", "commenter": "2" }
  }
}
```



前端工程化

模块化

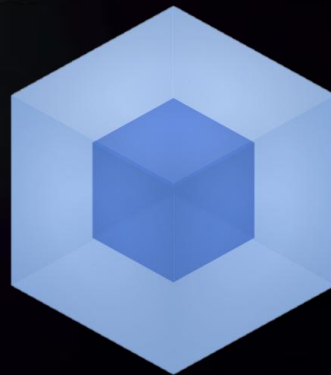
规范化

组件化

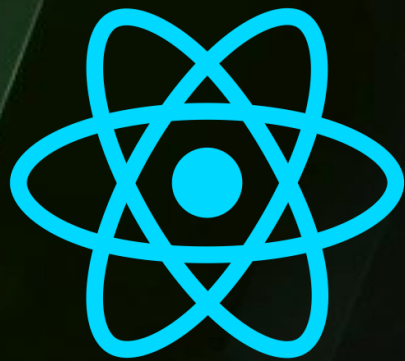
自动化



自动化



组件化



主流框架对比

名称	特点
React.js	虚拟DOM、单向数据流
Angular.js	强类型、双向绑定、流式数据处理
Vue.js	包容性

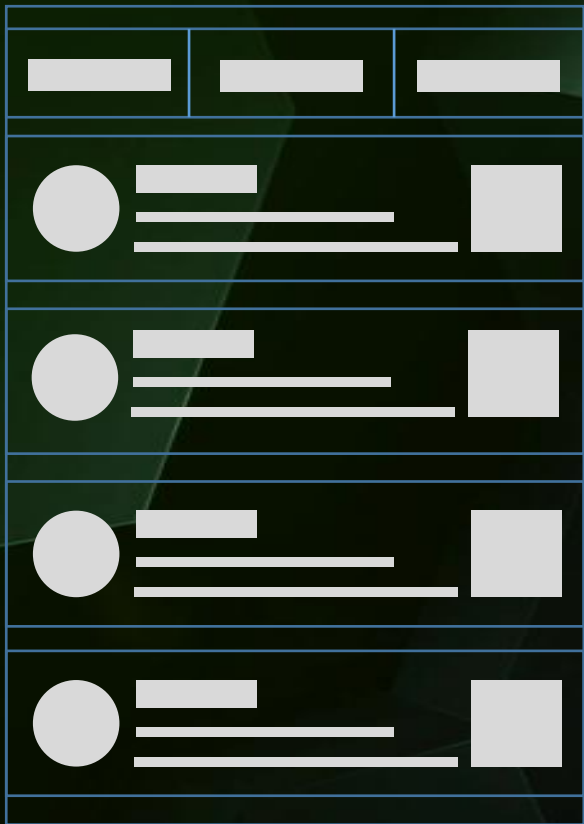


多样性



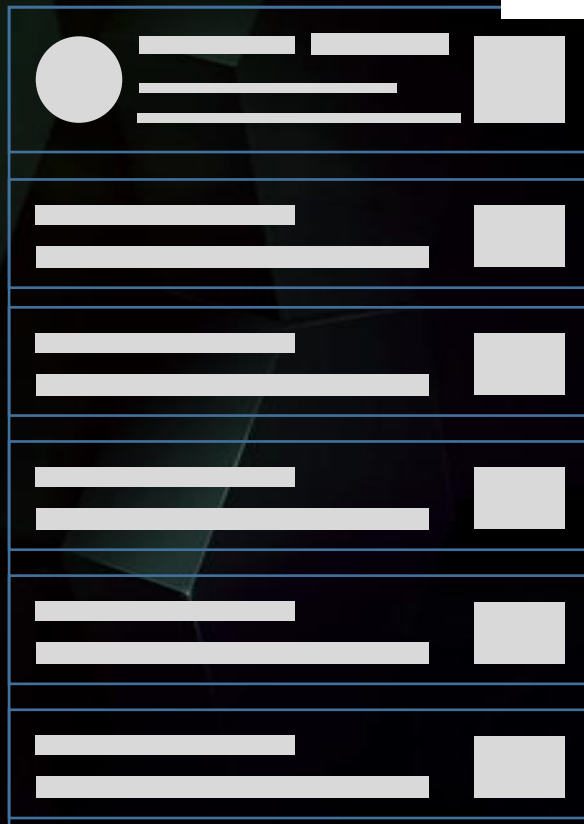
热闹编程





查询条件

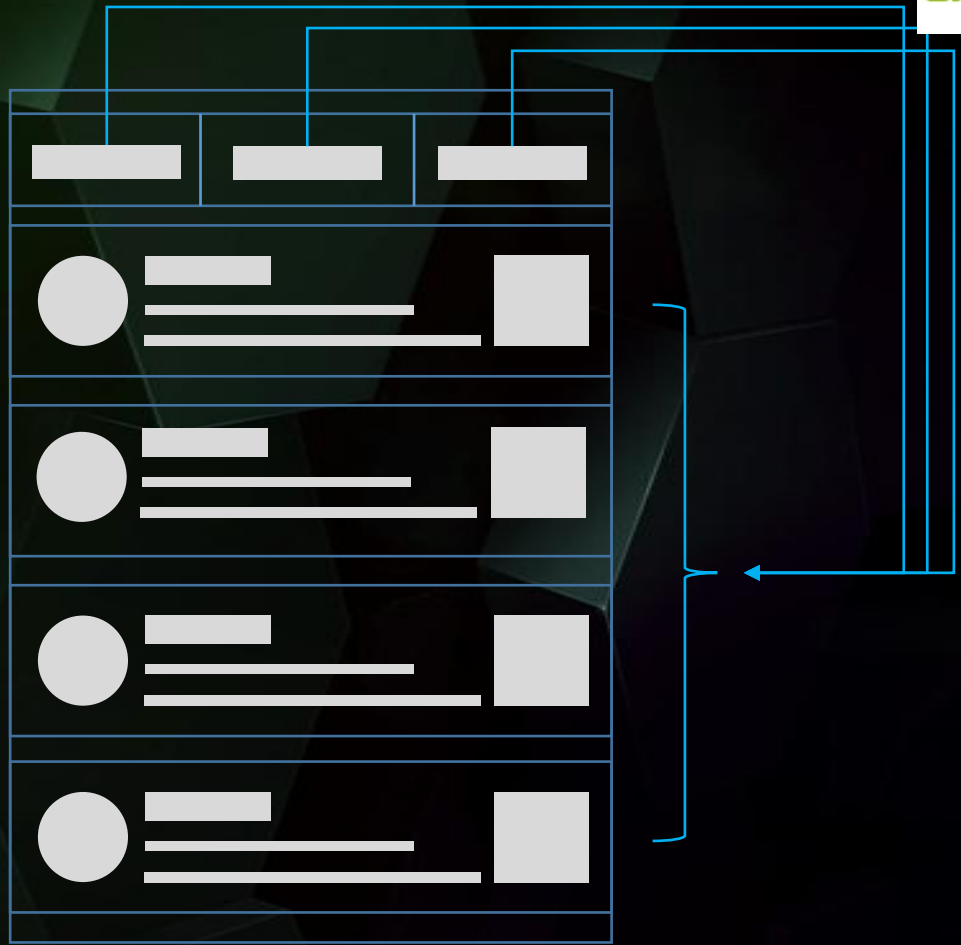
列表项



基本信息

打分项

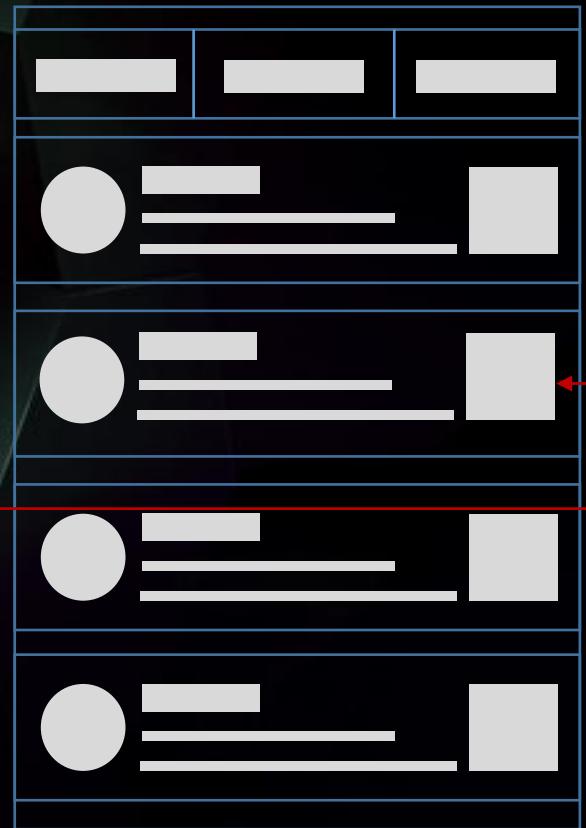
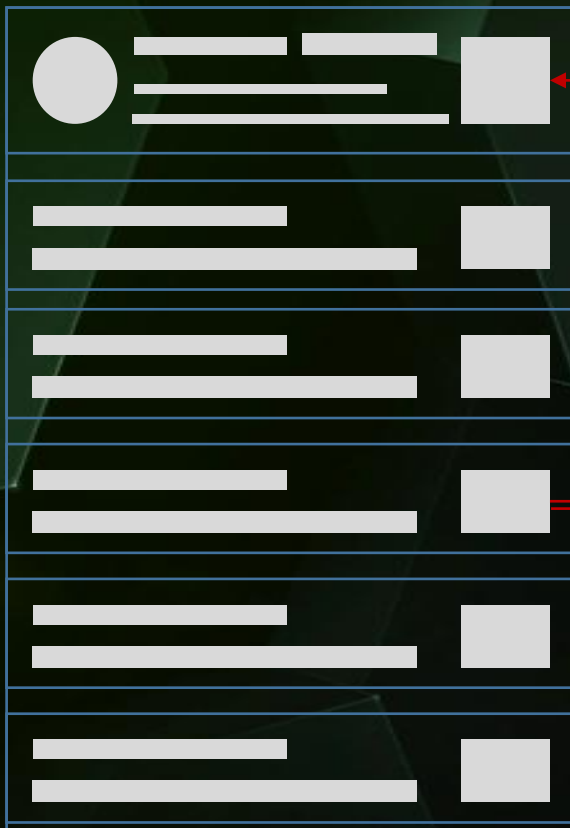


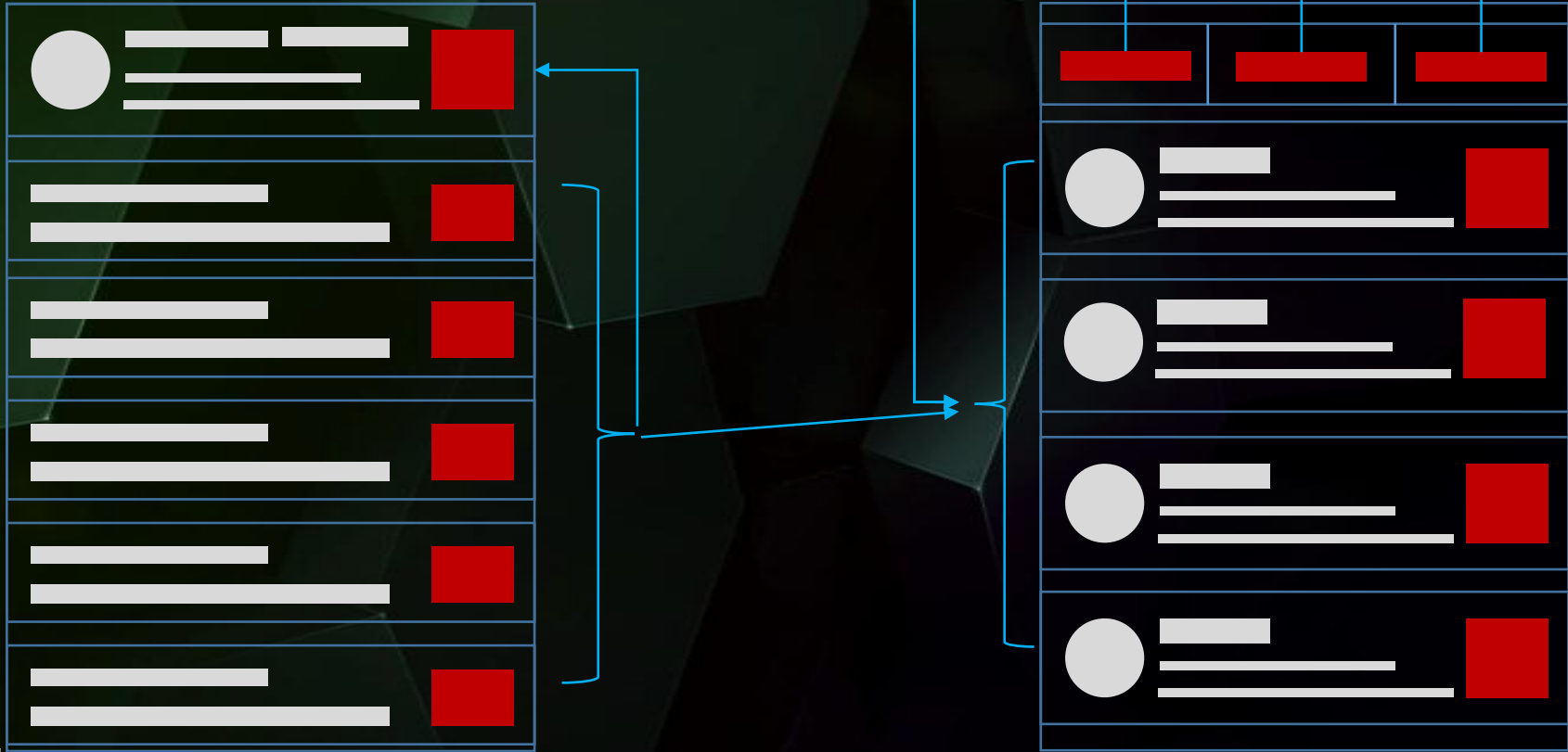


○	—	—	■
—	—	—	■
—	—	—	■
—	—	—	■
—	—	—	■
—	—	—	■
—	—	—	■

全部打分后汇总结果

当前打分结果






	A	B	C	D	E
1	用户ID	头像	部门	职位	得分
2	1	avatar.jpg	前端开发部	总监	90.85
3	2	avatar.jpg	前端开发部	Leader	95.12
4	3	avatar.jpg	前端开发部	开发	85.01
5	4	avatar.jpg	前端开发部	开发	59.45
6	5	avatar.jpg	测试部	Leader	89.56
7	6	avatar.jpg	测试部	测试	40.11
8	7	avatar.jpg	测试部	测试	45.12



```
"avatar": "avatar.jpg",
"name": "张三",
"department": "前端开发部",
"position": "开发",
"rules": [
  {
    "id": 1,
    "title": "这是第一个考评项",
    "description": "这个考评只要看被测人员的基本能力",
    "score": null
  },
  {
    "id": 2,
    "title": "这是第二个考评项",
    "description": "这个考评只要看被测人员的基本能力",
    "score": 5
  },
  {
    "id": 3,
    "title": "这是第三个考评项",
    "description": "这个考评只要看被测人员的基本能力",
    "score": null
  },
  {
    "id": 4,
    "title": "这是第四个考评项",
    "description": "这个考评只要看被测人员的基本能力",
    "score": null
  }
]
```

	<input type="text"/> <input type="text"/> <input type="text"/>	<input type="checkbox"/>
<input type="text"/> <input type="text"/>	<input type="checkbox"/>	
<input type="text"/> <input type="text"/>	<input type="checkbox"/>	
<input type="text"/> <input type="text"/>	<input type="checkbox"/>	
<input type="text"/> <input type="text"/>	<input type="checkbox"/>	
<input type="text"/> <input type="text"/>	<input type="checkbox"/>	



分布式系统的CAP定理

一致性
C

可用性
A

分区容错性
P



II 函数式编程

- 函数无副作用

- 数据不可变

- Immutable.js

如果我们不把“函数式编程”当做一个语言特性，只是当做一种“风格”，那么JavaScript就非常适合使用这种风格



数据



组件化



数据引力



谢谢观看

THANKS

