

Vue3 集训营

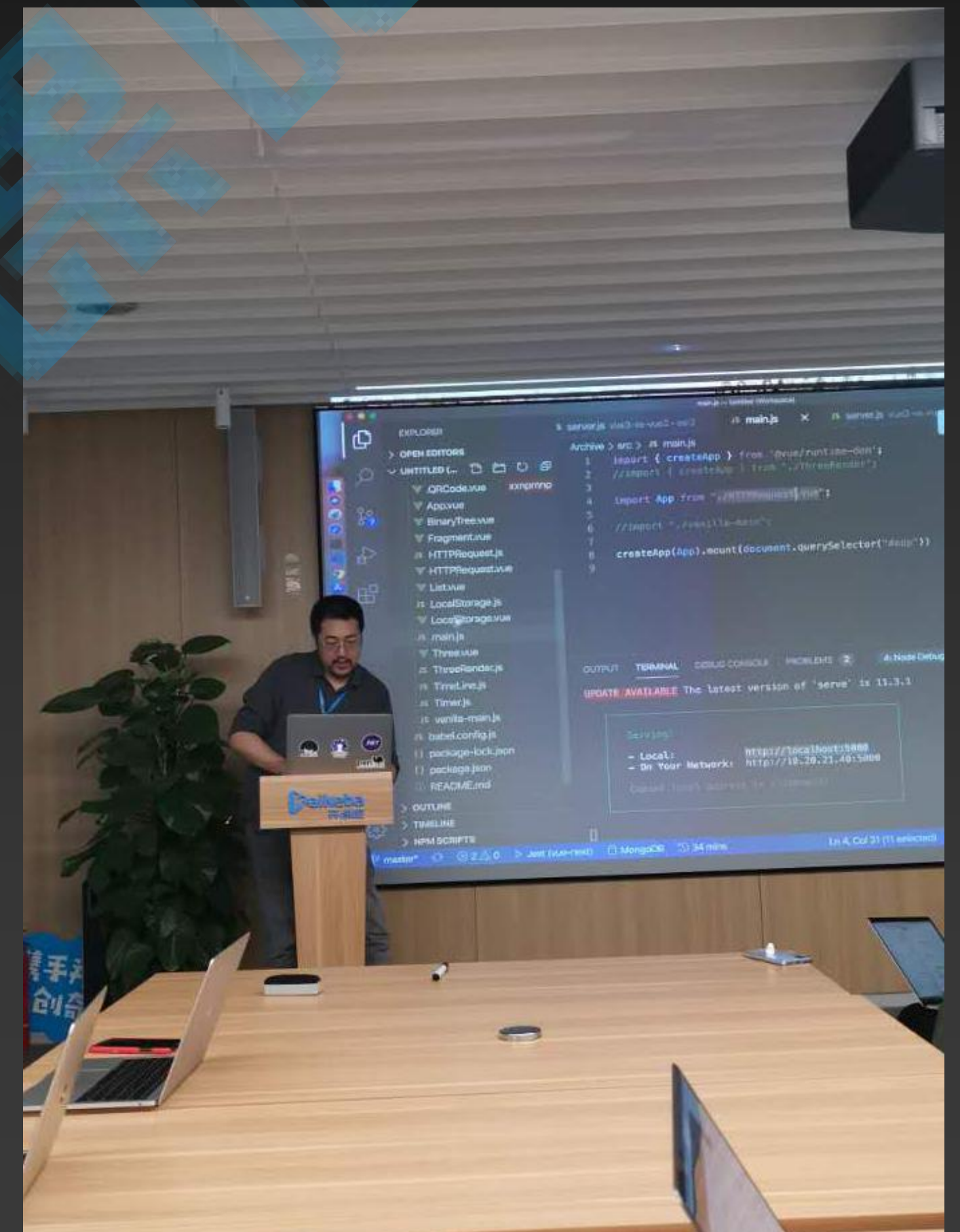
@崔效瑞

关于我

- 崔效瑞@键盘侠
- Github: <https://github.com/cuixiaorui>
- B站: <https://space.bilibili.com/175301983>
- 掘金: <https://juejin.im/user/58feaeec61ff4b00667515f7/posts>

关于集训营

- 玩中学 《飞机大战》
- 打造面试亮点杀器



今日目标

- 认识 *vue3*
- 构建《飞机大战》第一步
 - 使用 *Custom renderer api* 让视图渲染到 *canvas* 平台
- 分析 *vue3* 初始化流程

为什么要学习 vue3

- 投资知识
- 国内最火技术
- 招聘硬需求(亮点)
- 弯道超车最好的时机

Vue3 亮点

- Performance (比 vue2 runtime快了2倍)
- Tree shaking (按需编译代码)
- Ts support (更优秀的Ts支持)
- Composition API (组合 API)
- Custom Renderer API (自定义渲染器)
- 内置新特性组件

Highlights:

- Performance
- Tree-shaking support
- Composition API
- Fragment, Teleport, Suspense
- Better TypeScript support
- Custom Renderer API

性能

- 重写了虚拟 *dom* 的实现
- 编译模板的优化(运行时编译)
- *update* 性能提高
- *SSR* 速度提高

Performance

- Rewritten virtual dom implementation
- Compiler-informed fast paths
- More efficient component initialization
- 1.3~2x better update performance*
- 2~3x faster SSR*

Vue3 vs Vue2

	Chrome mount	Chrome update avg	Chrome update best	Chrome memory (mb)
Vue 2 template + with	93.46	23.17	16.44	11.9
Vue 3 template + with	77.43	9.18	7.69	4.5
Improvement over v2 (template + with)	20.70%	152.40%	113.78%	164.44%
Vue 2 render fn (manual h, no this access)	90.9	15.75	8.18	10.2
Vue 3 render fn (manual h, no this access)	76.46	7.56	5.33	7.4
Improvement over v2 (manual)	18.89%	108.33%	53.47%	37.84%
Vue 2 template no with	97.44	13.18	8.29	9.9
Vue 3 template no with	62.82	5.64	3.78	4.5
Improvement over v2 (template no with)	55.11%	133.69%	119.31%	120.00%
Vue 2 raw (no reactive state)	88.67	12.27	7.99	8.6
Vue 3 raw (no reactive state)	47.6	3.37	2.38	3.7
Improvement over v2 (raw)	86.28%	264.09%	235.71%	132.43%

Tree shaking

- 按需打包
- Vue2 纯 helloWorld: 31.94kb

Tree-shaking

- Most optional features (e.g. v-model, <transition>) are now tree-shakable
- Bare-bone HelloWorld size: **13.5kb**
 - 11.75kb with only Composition API support
- All runtime features included: **22.5kb**
 - More features but still lighter than Vue 2

TypeScript Support

- 自动的类型定义提示

```
    }); setup(props: {}, ctx:
    }, SetupContext<Record<string, any>>): void |
      RenderFunction
    setup(,){
  }
});
```

Better TypeScript Support

- Codebase written in TS w/ auto-generated type definitions
- API is the same in JS and TS
 - In fact, code will also be largely the same
- TSX support
- Class component is still supported ([vue-class-component@next](#) is currently in alpha)

Fragment

- 不再限于模板中的单个根节点

Fragments

- No longer limited to a single root node in templates
- Manual render functions can simply return Arrays
- “Just works”

▼ HelloWorld.vue vue3/src/components/HelloWorld.vue/ { } "HelloWorld.vue"/ script

You, a few seconds ago | 1 author (You)

```
1 <template>
2   <div>hello</div>
3   <div>world</div>
4 </template>
```

Composition API

- 灵活的逻辑组合与复用
- 响应式对象
 - *ref*
 - *reactive*
- 生命周期
 - *onMounted*
 - *onUnmounted*



Custom Renderer API

- 解决什么问题

Custom Renderer API

- NativeScript Vue integration underway by [@rigor789](#)
- Users already experimenting w/ WebGL custom renderer that can be used alongside a normal Vue application ([Vugel](#))

解决什么问题

```
App.vue demo-vue/src/App.vue/{} "App.vue"/script
1  <template>
2    <div id="app">
3      
4      <HelloWorld msg="Welcome to Your Vue.js App"/>
5    </div>
6  </template>
7
8  <script>
9    import HelloWorld from './components/HelloWorld.vue'
10
```



总结为什么学vue3

- 更快
- 更好

Custom Renderer API

- `createRenderer`
- 接口
 - `createElement`
 - `insert`
 - `patchProp`

```
import {
  createRenderer,
} from '@vue/runtime-core'

function ensureRenderer() {
  return renderer || (renderer = createRenderer(rendererOptions))
}
```


实现自定义的 *renderer*

- *canvas* 和 *vue3* 的碰撞
- *canvas* \rightarrow *pixi.js*
- 目标: *vue3* 结合 *pixi.js* 实现把图形绘制到 *canvas* 上

setup 环境

- 配置 *webpack*
- 配置 *scripts*
 - *build*
 - *serve*

实现 *renderer*

- `createRenderer({...nodeOps, patchProp})`
- `const nodeOps = {}`
- `function patchProp(){}`

构建程序入口

- `createApp()`

JS main.js demo-vue/src/main.js

You, a few seconds ago | 1 author (You)

```
1 | import { createApp } from "vue";  
2 | import App from "./App.vue";  
3 | createApp(App).mount("#app");  
4 |
```


构建 *canvas* 根容器

- 引入 *pixi.js*
- 初始化 *canvas* 容器

构建根组件

- `defineComposition()`
- `Renderer`
- `h`
- `vnode`

完善渲染接口

- *insert*
- *createElement*
- *setElementText*

完善 `patchProp`

- `patchProp(el, key, prevValue, nextValue)`

交心时间

Runtime-dom

- 读一读源码

custom Renderer 实现原理

- 深入 *runtime-core* 模块
- 分析 *insert* 和 *createElement* 的整个流程

总结

- 灵魂拷问 – 你今天学到了啥？
- 自定义渲染器
 - 是什么 (*what*)
 - 怎么用 (*how*)
 - 实现原理 (*why*)
- *Vue3* 运行时初始化流程分析

明天内容预告

- 飞机大战搞起来~

