



那些年我们一起踩过的坑

——WebIDE 前端札记

WebIDE 团队



WebIDE 前端历程

- ▶ v1版 14年 backbone + coffeescript + underscore + bower + npm + grunt (ace, sh.js)
- ▶ v2版 15年上 React + coffeescript + lodash + flux + bower + npm + webpack + gulp (ace, sh.js)
- ▶ code insight 项目 15年下 React + es6 + Redux + immutable.js + npm + webpack + gulp (codemirror)
- ▶ v3版 16年9月 React + es6 + lodash + Redux + yarn + webpack (codemirror, sh.js)



Coding WEBIDE V2 的架构

React + Flux



Coding WebIDE V3 的架构

- ▶ React + Redux
- ▶ React + MobX



状态管理解决什么问题

- ▶ 组件间通讯
- ▶ 它提供工具在某个地方保存状态、修改状态和更新状态。你可以从一个地方获得状态，一个地方修改它，一个地方得到它的更新。它遵循单一数据源的原则。这让我们更容易推断状态的值和状态的修改，因为它们与我们的组件是解耦的。
- ▶ 状态管理库一般都有附带的工具，例如在 React 中使用的有 React-Redux 和 MobX-React



Redux 与 MobX 比较

- ▶ 都是状态管理
- ▶ Redux 被 FP（函数式编程）原则所影响
- ▶ MobX 仍然是面向对象
- ▶ 单 store 和 多 stores



Redux

```
{  
  post: {  
    id: 'a',  
    authorId: 'b',  
    ...  
  },  
  author: {  
    id: 'b',  
    postIds: ['a', ...],  
    ...  
  }  
}
```



MobX

```
{  
  post: {  
    id: 'a',  
    ...  
    author: {  
      id: 'b',  
      ...  
    }  
  }  
}
```



编辑器

- ▶ ace
- ▶ codemirror



ace

- ▶ 实现了 diff view, merge view
- ▶ java 的代码提示



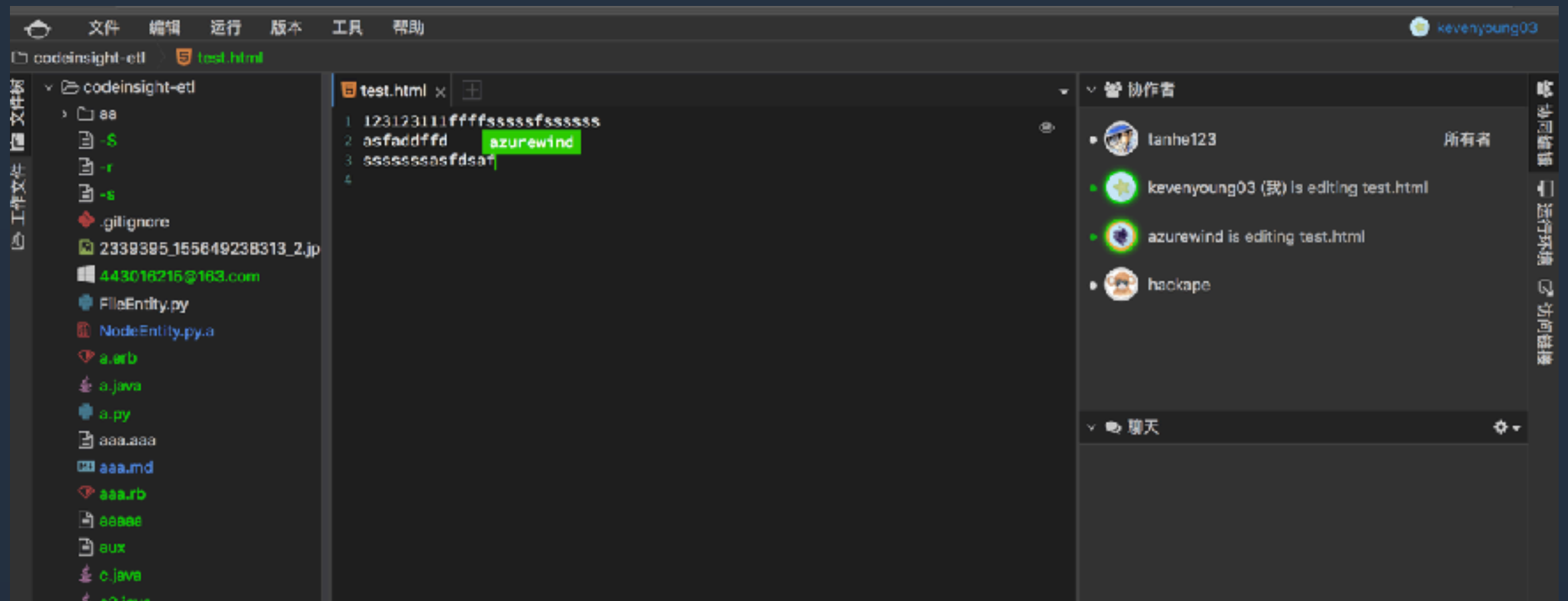
codemirror

- ▶ 自带 diff view, merge view
- ▶ java 的代码提示



协同编辑

- ▶ 基于 OT (Operational Transformation) 协议的协同编辑



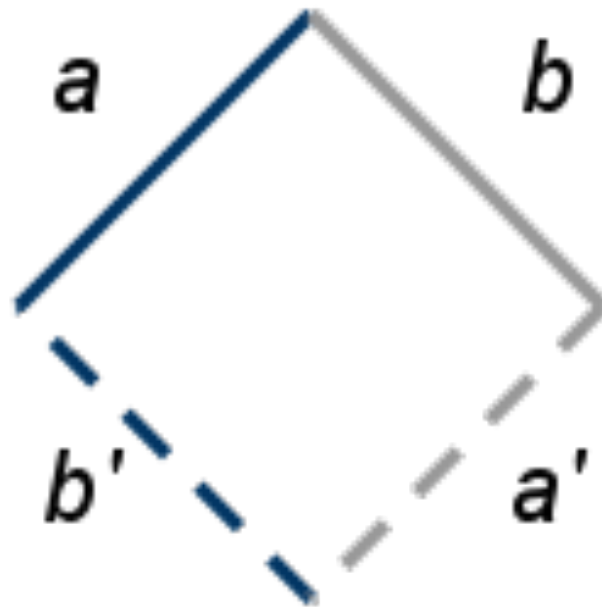
Operational Transformation

- ▶ `insertCharacters` — Inserts the specified string at the current index
- ▶ `deleteCharacters` — Deletes the specified string from the current index
- ▶ `openElement` — Creates a new XML open-tag at the current index
- ▶ `deleteOpenElement` — Deletes the specified XML open-tag from the current index
- ▶ `closeElement` — Closes the first currently-open tag at the current index
- ▶ `deleteCloseElement` — Deletes the XML close-tag at the current index
- ▶ `annotationBoundary` — Defines the changes to any annotations (starting or ending) at the current index
- ▶ `retain` — Advances the index a specified number of items



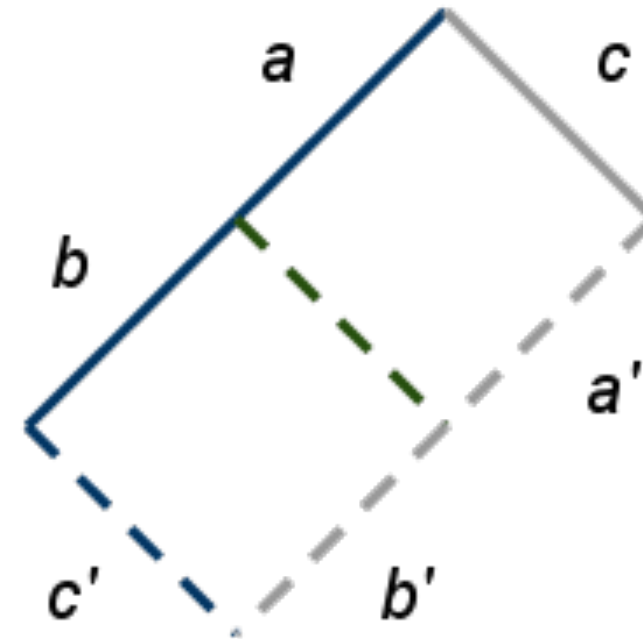
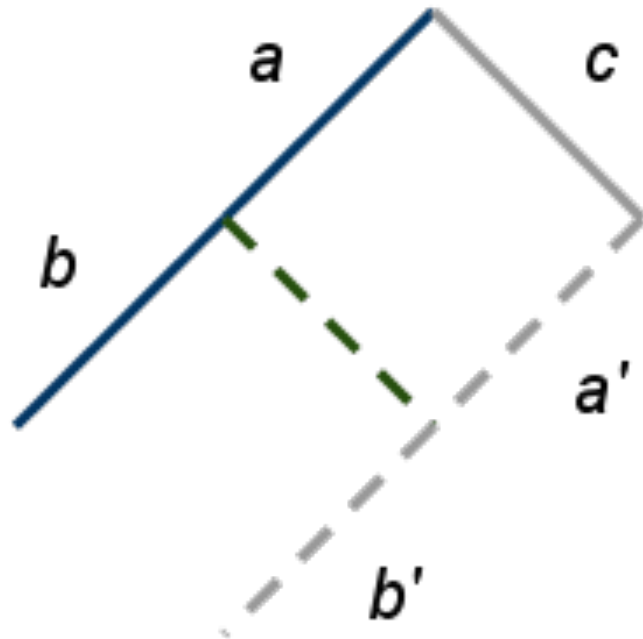
Operational Transformation

$\text{xform}(a, b) = (a', b')$, where $b' \circ a \equiv a' \circ b$



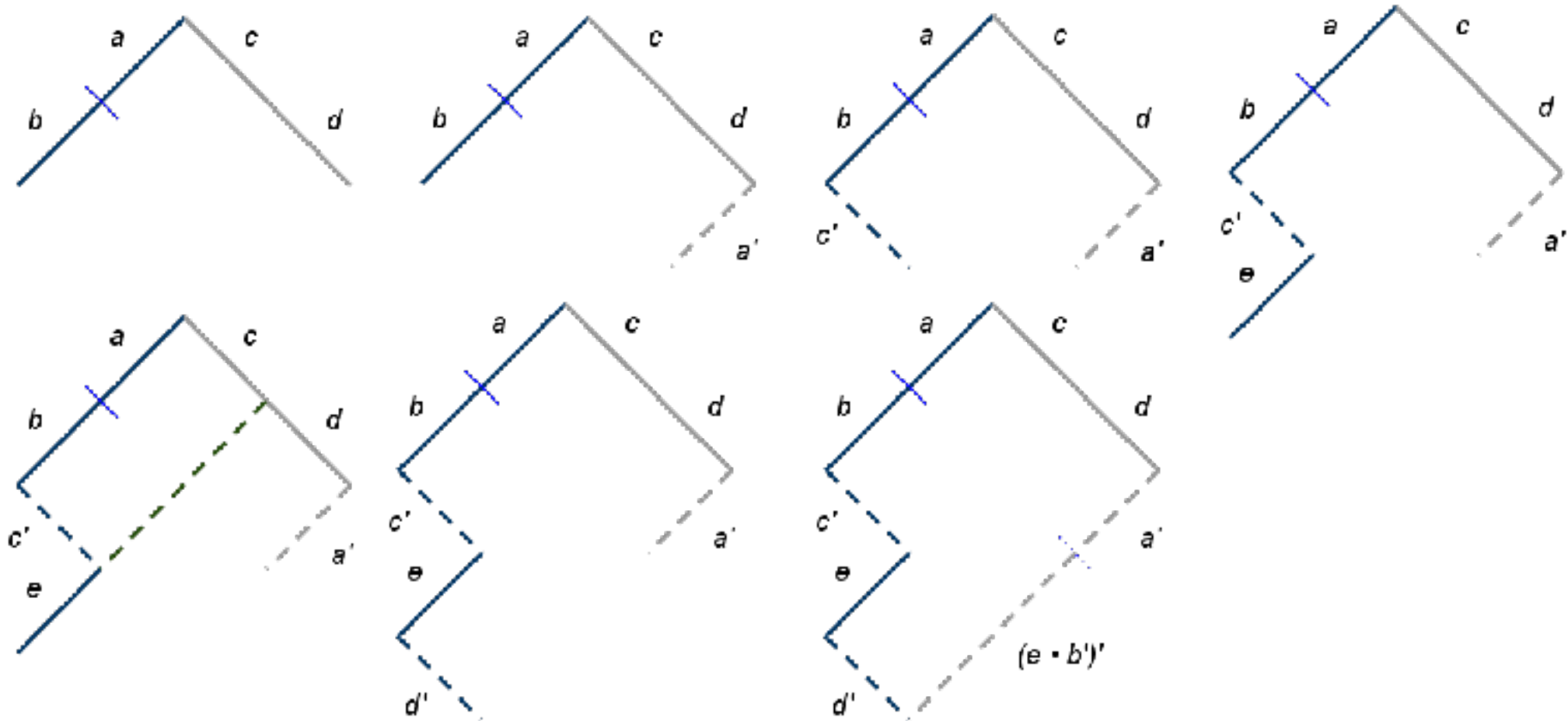
Operational Transformation

$\text{xform}(a, b) = (a', b')$, where $b' \circ a \equiv a' \circ b$



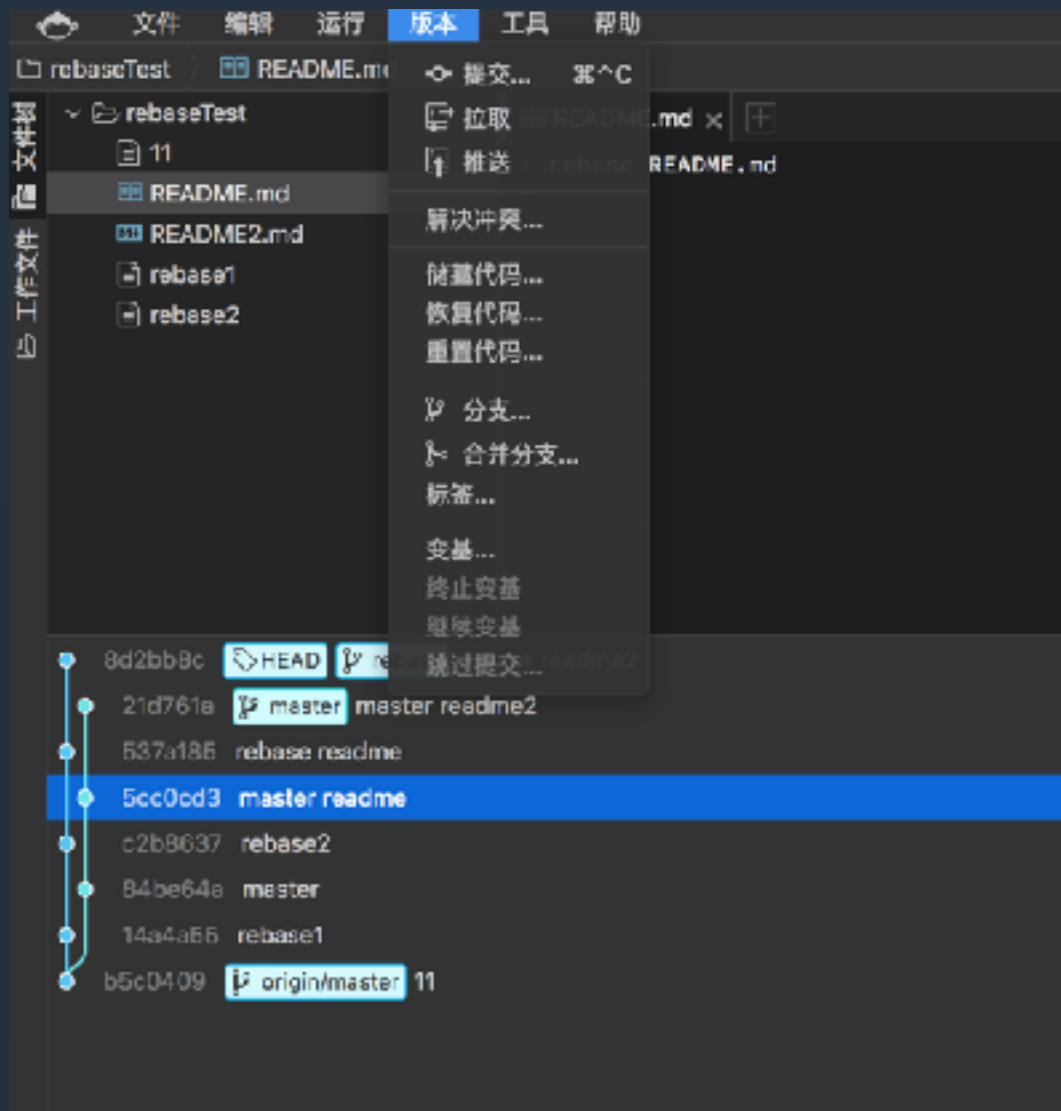
Operational Transformation

$\text{xform}(a, b) = (a', b')$, where $b' \circ a \equiv a' \circ b$



Git 相关功能

完整的 git ui 操作



Terminal

- ▶ sh.js
- ▶ xterm



sh.js

- ▶ 实现了宽字符显示支持
- ▶ 实现了中文输入法支持
- ▶ 用 dom 实现



xterm

- ▶ 自带宽字符显示支持
- ▶ 自带中文输入法支持
- ▶ 右键和剪贴板支持
- ▶ 以 canvas 进行绘制



DOM 的劣势

- ▶ DOM 很慢
- ▶ DOM 操作都是同步的
- ▶ 网页是单线程，并且没有硬件加速



canvas 的优势

- ▶ 降低操作延时
- ▶ 绕开 dom，不用等同步
- ▶ 有硬件加速
- ▶ flipboard 与 React-canvas



canvas 的劣势

- ▶ 只是位图，没有语义
- ▶ 实现 UI 要把所有页面元素都实现一遍（超链，组件，css效果等）
- ▶ 不能做 seo



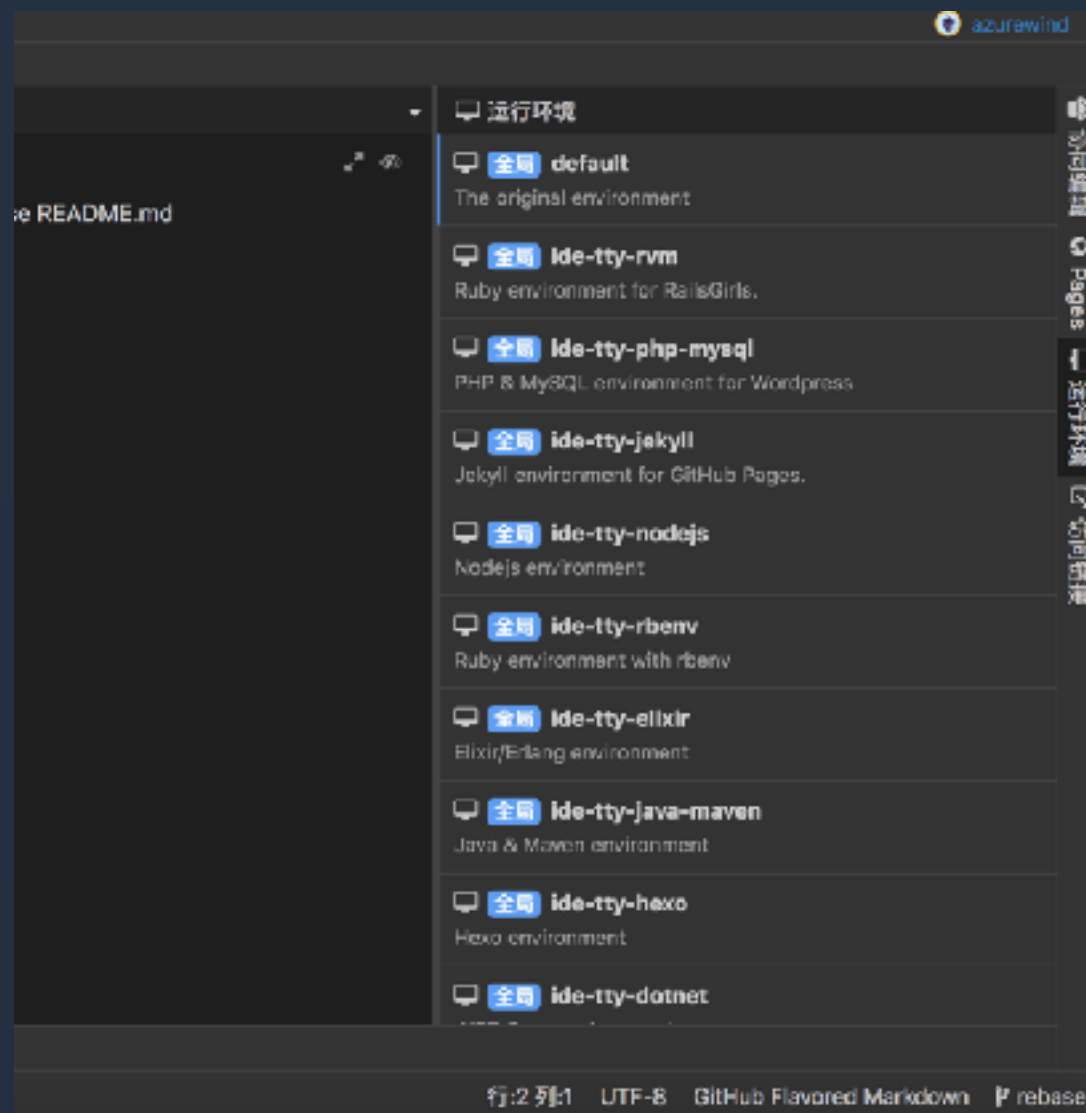
国际化

- ▶ [yahoo/React-intl](#)
- ▶ 自制国际化组件



插件体系

- ▶ v3 版我们实现了一套插件体系



Thanks!

