Sunday • 更新于 2022-10-19

上一节 11：基于编辑器... 13：基于编辑器... 下一节

12：基于编辑器的指令(v-xx)处理：JavaScript AST，transform 的转化逻辑

当 `vif` 模块构建完成之后，接下来我们就只需要在 `transform` 中针对 `IF` 使用 `vif` 模块进行转化即可

我们知道转化的主要方法为 `traverseNode` 函数，所以我们需要在该函数内增加如下代码：

代码块

```
export function traverseNode(node, context: TransformContext) {
    ...
    // 循环获取节点的 transform 方法，缓存到 exitFns 中
    for (let i = 0; i < nodeTransforms.length; i++) {
        const onExit = nodeTransforms[i](node, context)
        if (onExit) {
+           // 指令的 transforms 返回为 数组，所以需要解构
+           if (isArray(onExit)) {
+               exitFns.push(...onExit)
+           } else {
+               exitFns.push(onExit)
+           }
        }
+       // 因为触发了 replaceNode，可能会导致 context.currentNode 发生变化，所以需要在这里校
+       if (!context.currentNode) {
+           // 节点已删除
+           return
+       } else {
+           // 节点更换
+           node = context.currentNode
+       }
    }

    // 继续转化子节点
    switch (node.type) {
+       case NodeTypes.IF_BRANCH:
        case NodeTypes.ELEMENT:
        case NodeTypes.ROOT:
            traverseChildren(node, context)
            break
        // 处理插值表达式 {{}}
        case NodeTypes.INTERPOLATION:
            context.helper(TO_DISPLAY_STRING)
            break
+       // v-if 指令处理
+       case NodeTypes.IF:
+           for (let i = 0; i < node.branches.length; i++) {
+               traverseNode(node.branches[i], context)
+           }
+           break
    }

    ...
}
```

至此，我们在 `transform` 中就拥有了处理 `if` 的能力。

代码块

```
export function baseCompile(template: string, options = {}) {
    const ast = baseParse(template)

    transform(
        ast,
        extend(options, {
+           nodeTransforms: [transformElement, transformText, transformIf]
        })
    )
    console.log(JSON.stringify(ast))
    return generate(ast)
}
```

运行测试实例 `packages/vue/examples/compiler/compiler-directive.html`，打印出 `JavaScript AST`（**注意：** 因为 `Symbol` 不会在 `json` 字符串下打印，所以我们需要手动加上）：

代码块

```
{"type":0,"children":[{"type":1,"tag":"div","tagType":0,"props":[],"children":[{"type":2
```

直接把以上内容复制到 `vue3` 源码的 `generate` 方法调用处（替换 `ast`），页面可正常渲染。证明当前的 `JavaScript AST` 处理完成。

11：基于编辑器的指令(v-xx)处理：JavaScri... 上一节 下一节 13：基于编辑器的指令(v-xx)处理：生成 ren...

我要提出意见反馈

企业服务 网站地图 网站首页 关于我们 联系我们 讲师招募 帮助中心 意见反馈 代码托管

意见反馈 收藏教程 标记书签