Sunday • 更新于 2022-10-19

上一节 04：源码阅读：... 06：框架实现：... 下一节

05：框架实现：构建 parse 方法，生成 context 实例

从这一小节开始，我们将实现 `vue-next-mini` 中的编辑器模块。首先我们第一步要做的就是生成 `AST` 对象。但是我们知道 `AST` 对象的生成颇为复杂，所以我们把整个过程分为成三步进行处理。

1. 构建 `parse` 方法，生成 `context` 实例
2. 构建 `parseChildren` ，处理所有子节点（**最复杂**）
    1. 构建有限自动状态机解析模板
    2. 扫描 token 生成 AST 结构
3. 生成 `AST` ，构建测试

那么本小节，我们就先处理第一步。

1. 创建 `packages/compiler-core/src/compile.ts` 模块，写入如下代码：

代码块

```
export function baseCompile(template: string, options) {
    return {}
}
```

2. 创建 `packages/compiler-dom/src/index.ts` 模块，导出 `compile` 方法：

代码块

```
import { baseCompile } from 'packages/compiler-core/src/compile'

export function compile(template: string, options) {
    return baseCompile(template, options)
}
```

3. 在 `packages/vue/src/index.ts` 中，导出 `compile` 方法：

代码块

```
export { compile } from '@vue/compiler-dom'
```

4. 创建 `packages/compiler-core/src/parse.ts` 模块下创建 `baseParse` 方法：

代码块

```
/**
 * 基础的 parse 方法，生成 AST
 * @param content tempalte 模板
 * @returns
 */
export function baseParse(content: string) {
    return {}
}
```

5. 在 `packages/compiler-core/src/compile.ts` 模块下的 `baseCompile` 中，使用 `baseParse` 方法：

代码块

```
import { baseParse } from './parse'

export function baseCompile(template: string, options) {
    const ast = baseParse(template)
    console.log(JSON.stringify(ast))

    return {}
}
```

那么至此，我们就成功的触发了 `baseParse` 。接下来我们去生成 `context` 上下文对象。

1. 在 `packages/compiler-core/src/parse.ts` 中创建 `createParserContext` 方法，用来生成上下文对象：

代码块

```
/**
 * 创建解析器上下文
 */
function createParserContext(content: string): ParserContext {
    // 合成 context 上下文对象
    return {
        source: content
    }
}
```

2. 创建 `ParserContext` 接口：

代码块

```
/**
 * 解析器上下文
 */
export interface ParserContext {
    // 模板数据源
    source: string
}
```

3. 在 `baseParse` 中触发该方法：

代码块

```
export function baseParse(content: string) {
    // 创建 parser 对象，未解析器的上下文对象
    const context = createParserContext(content)
  console.log(context)
    return {}
}
```

那么至此我们成功得到了 `context` 上下文对象。

我们可以创建测试实例 `packages/vue/examples/compiler/compiler-ast.html` ：

代码块

```
<script>
  const { compile } = Vue
  // 创建 template
  const template = `<div> hello world </div>`

  // 生成 render 函数
  const renderFn = compile(template)
</script>
```

可以成功打印 `context`

我要提出意见反馈