Как написать поддержку фреймворка?

Старовойт Андрей
WSD Екатеринбург

Обо мне

- Работаю в команде WebStorm ~ 6 лет
- Занимаюсь поддержкой TypeScript / JavaScript / JSX, …

@JetBrains. All rights reserved

Фреймворк

@JetBrains. All rights reserved

Фреймворк?

Встраивается в существующую экосистему (Web), представляет новый синтаксис с новыми возможностями*

* С точки зрения разработчика IDE

@JetBrains. All rights reserved

Фреймворк

Angular, React, Vue, Svelte, Riot, …

~~Bootstrap, Express, Lodash, Moment~~

@JetBrains. All rights reserved

Поддержка фреймворка

см. Поддержка языка

@JetBrains. All rights reserved

Синтаксис

@JetBrains. All rights reserved

```
import React from 'react';
import logo from './logo.svg';
import './App.css';

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <img src={logo} className="App-logo" alt="logo" />
        <p>
          Edit <code>src/App.js</code> and save to reload.
        </p>
        <a
          className="App-link"
          href="https://reactjs.org"
          target="_blank"
          rel="noopener noreferrer"
        >
```

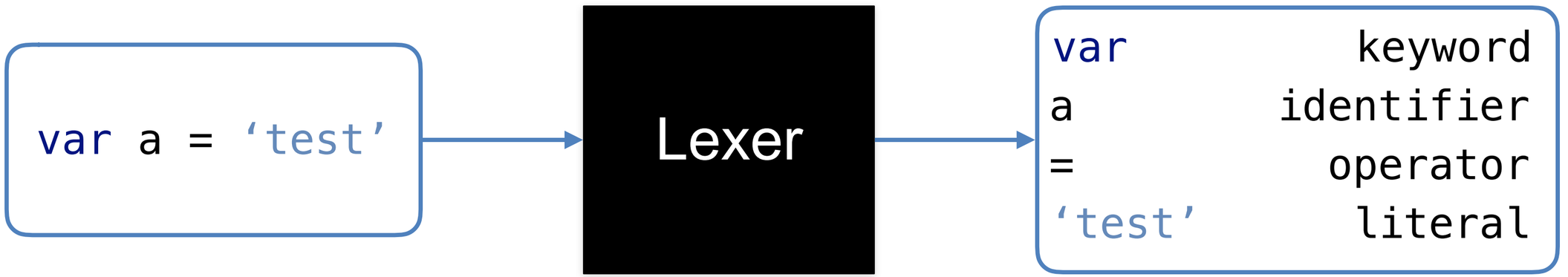
Синтаксическая подсветка

≈ понимание синтаксиса

```
function helloFirst() {
    return "hello " + 1;
}

helloFirst();
```

@JetBrains. All rights reserved

Лексер



@JetBrains. All rights reserved

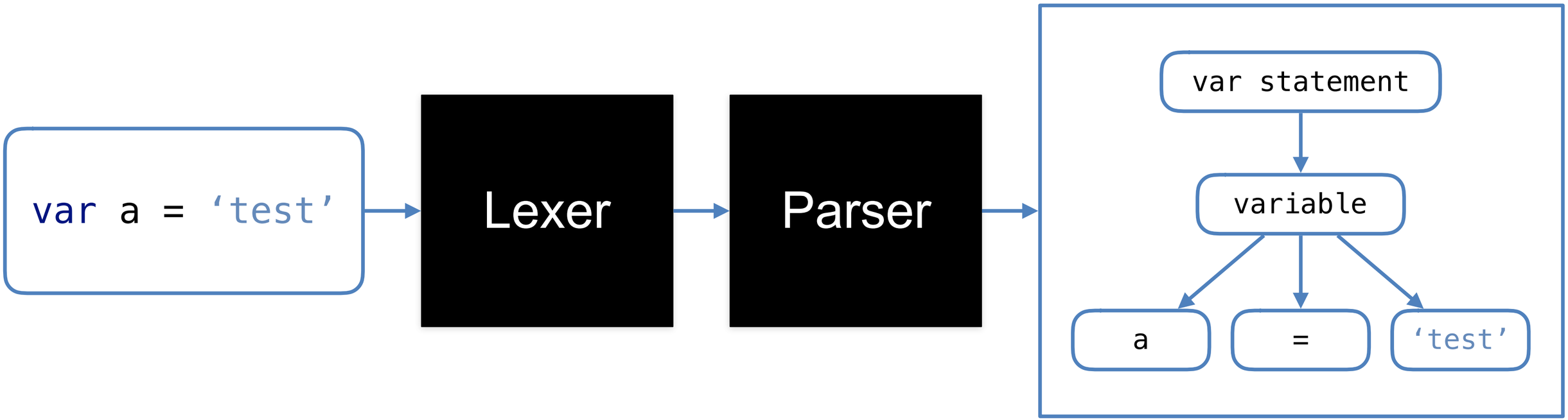
Лексер

Flex
Набор регулярных выражений

```
{WHITE_SPACE_CHAR}+ { return JavaTokenType.WHITE_SPACE; }

{C_STYLE_COMMENT} { return JavaTokenType.C_STYLE_COMMENT; }
{END_OF_LINE_COMMENT} { return JavaTokenType.END_OF_LINE_COMM
{DOC_COMMENT} { return JavaDocElementType.DOC_COMMENT; }

{LONG_LITERAL} { return JavaTokenType.LONG_LITERAL; }
{INTEGER_LITERAL} { return JavaTokenType.INTEGER_LITERAL; }
{FLOAT_LITERAL} { return JavaTokenType.FLOAT_LITERAL; }
{DOUBLE_LITERAL} { return JavaTokenType.DOUBLE_LITERAL; }
{CHARACTER_LITERAL} { return JavaTokenType.CHARACTER_LITERAL;
{STRING_LITERAL} { return JavaTokenType.STRING_LITERAL; }

"true" { return JavaTokenType.TRUE_KEYWORD; }
"false" { return JavaTokenType.FALSE_KEYWORD; }
"null" { return JavaTokenType.NULL_KEYWORD; }

"abstract" { return JavaTokenType.ABSTRACT_KEYWORD; }
"assert" { return myAssertKeyword ? JavaTokenType.ASSERT_KEYW
"boolean" { return JavaTokenType.BOOLEAN_KEYWORD; }
"break" { return JavaTokenType.BREAK_KEYWORD; }
"byte" { return JavaTokenType.BYTE_KEYWORD; }
"case" { return JavaTokenType.CASE_KEYWORD; }
"catch" { return JavaTokenType.CATCH_KEYWORD; }
"char" { return JavaTokenType.CHAR_KEYWORD; }
"class" { return JavaTokenType.CLASS_KEYWORD; }
```

@JetBrains. All rights reserved

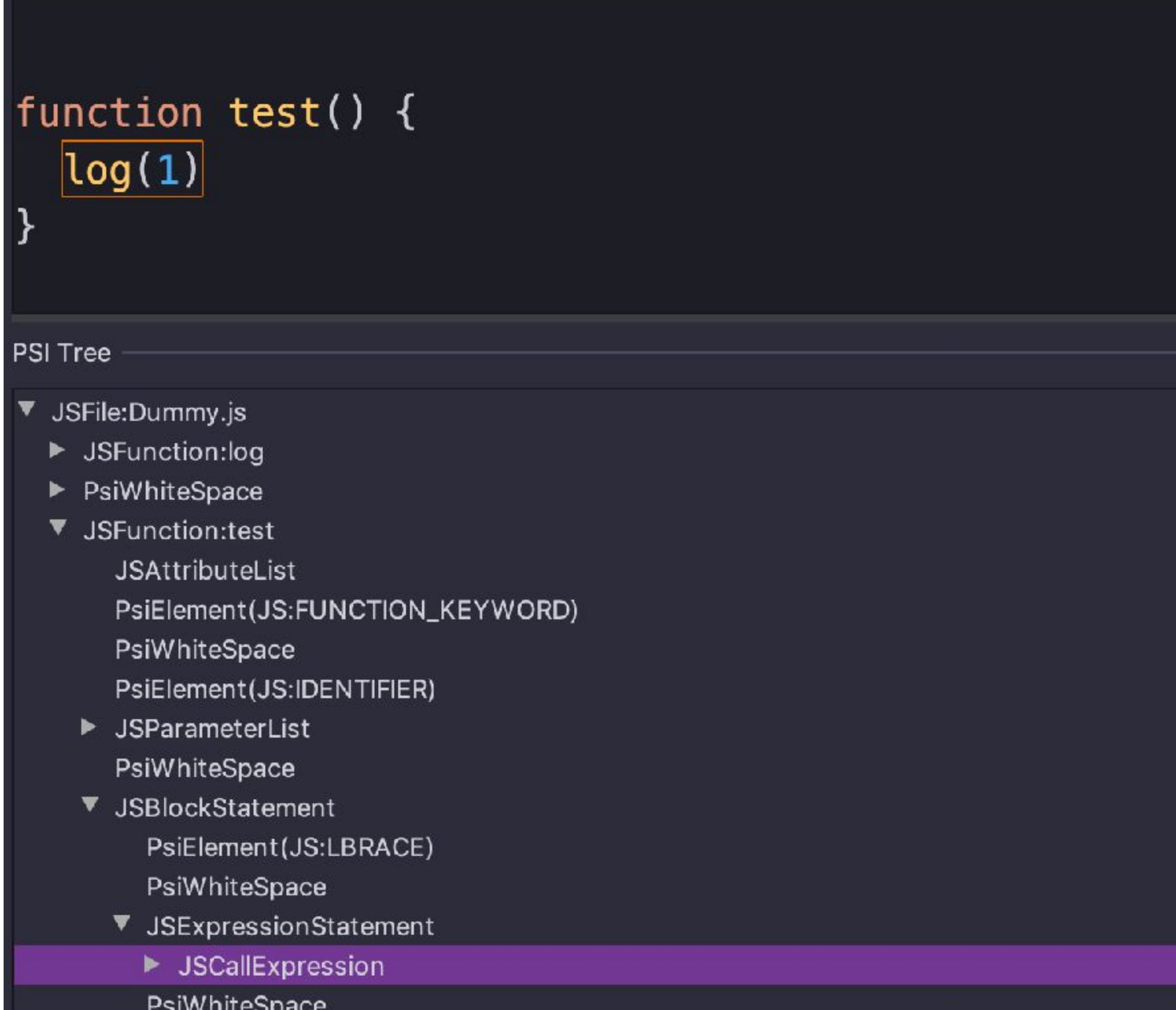
Синтаксическое дерево



@JetBrains. All rights reserved

Парсер

JavaScript function



@JetBrains. All rights reserved

Синтаксическое дерево

- Подсветка синтаксических элементов
- Подсветка синтаксических ошибок

@JetBrains. All rights reserved

Синтаксическое дерево

—

JavaScript, TypeScript, HTML, CSS, JSON,

…

@JetBrains. All rights reserved

Синтаксическое дерево

Фреймворки?

@JetBrains. All rights reserved

Надстройка над языком

HTML в Angular template

```
import {Component} from '@angular/core';

@Component({
  selector: 'app-root',
  template: `
  <h1>This template brought to you by:
    <span>ES6!</span>
  </h1>
`,
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'untitled24';
}
```

@JetBrains. All rights reserved

Надстройка над языком

—

HTML в Angular template

```
import {Component} from '@angular/core';

@Component({
  selector: 'app-root',
  template: `
  <h1>This template brought to you by:
    <span>ES6!</span>
  </h1>
`,
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'untitled24';
}
```

@JetBrains. All rights reserved

Расширение синтаксиса

Интерполяция в Angular template

```
@Component({
  selector: 'app-root',
  template: `
      <h1>
          Welcome to {{ title }}!
      </h1>`,
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'untitled24';
}
```

@JetBrains. All rights reserved

Расширение синтаксиса

Интерполяция в Angular template

```
<div style="...">
  <h1>
    Welcome to {{ title }}!
  </h1>
  <img width="300" alt="Angular Logo" src="data:">
</div>
<h2>Here are some links to help you start: </h2>
```

PSI Tree

```
▼ HtmlTag:h1
    XmlToken:XML_START_TAG_START
    XmlToken:XML_NAME
    XmlToken:XML_TAG_END
  ▶ XmlText
    PsiElement(NG:INTERPOLATION_START)
  ▼ ASTWrapperPsiElement(NG:INTERPOLATION_EXPR)
      PsiWhiteSpace
    ▶ Angular2Interpolation
      PsiWhiteSpace
    PsiElement(NG:INTERPOLATION_END)
  ▶ XmlText
    XmlToken:XML_END_TAG_START
```

@JetBrains. All rights reserved

Расширение синтаксиса

Интерполяция в Angular template

```
@Component({
  selector: 'app-root',
  template: `
      <h1>
          Welcome to {{ title }}!
      </h1>`,
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'untitled24';
}
```

@JetBrains. All rights reserved

Расширение синтаксиса

JSX теги

```
function App() {
  return (
    <div className="App">
      <header className="App-header">
        <img src={logo} className="App-logo" alt="logo"
        <p>
          Edit <code>src/App.js</code> and save to reloa
        </p>
        <a
          className="App-link"
          href="https://reactjs.org"
          target="_blank"
          rel="noopener noreferrer"
        >
          Learn React
        </a>
      </header>
    </div>
  );
}
```

@JetBrains. All rights reserved

Расширение синтаксиса

JSX теги

```
function App() {
  return (
    <div className="App">
      <header className="App-header">
        <img src={logo} className="App-logo" alt="logo" />
        <p>
          Edit <code>src/App.js</code> and save to reload.
        </p>
```

PSI Tree

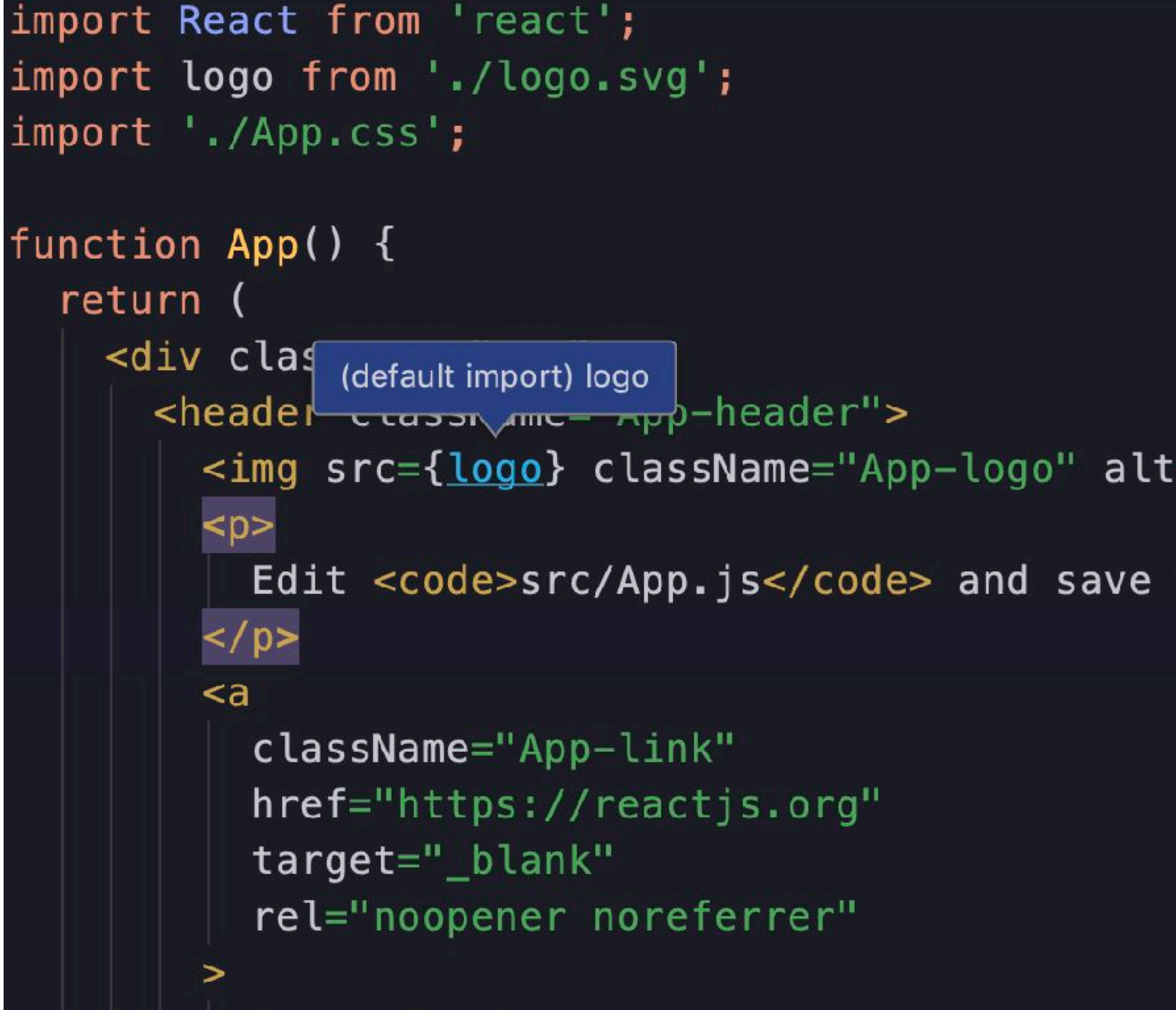
```
PsiWhiteSpace
▼ JSBlockStatement
    PsiElement(JS:LBRACE)
    PsiWhiteSpace
  ▼ JSReturnStatement
      PsiElement(JS:RETURN_KEYWORD)
      PsiWhiteSpace
    ▼ JSParenthesizedExpression
        PsiElement(JS:LPAR)
        PsiWhiteSpace
      ▼ XmlTag:div
          XmlToken:XML_START_TAG_START
          XmlToken:XML_TAG_NAME
          PsiWhiteSpace
```

@JetBrains. All rights reserved

Семантика

@JetBrains. All rights reserved

References



@JetBrains. All rights reserved

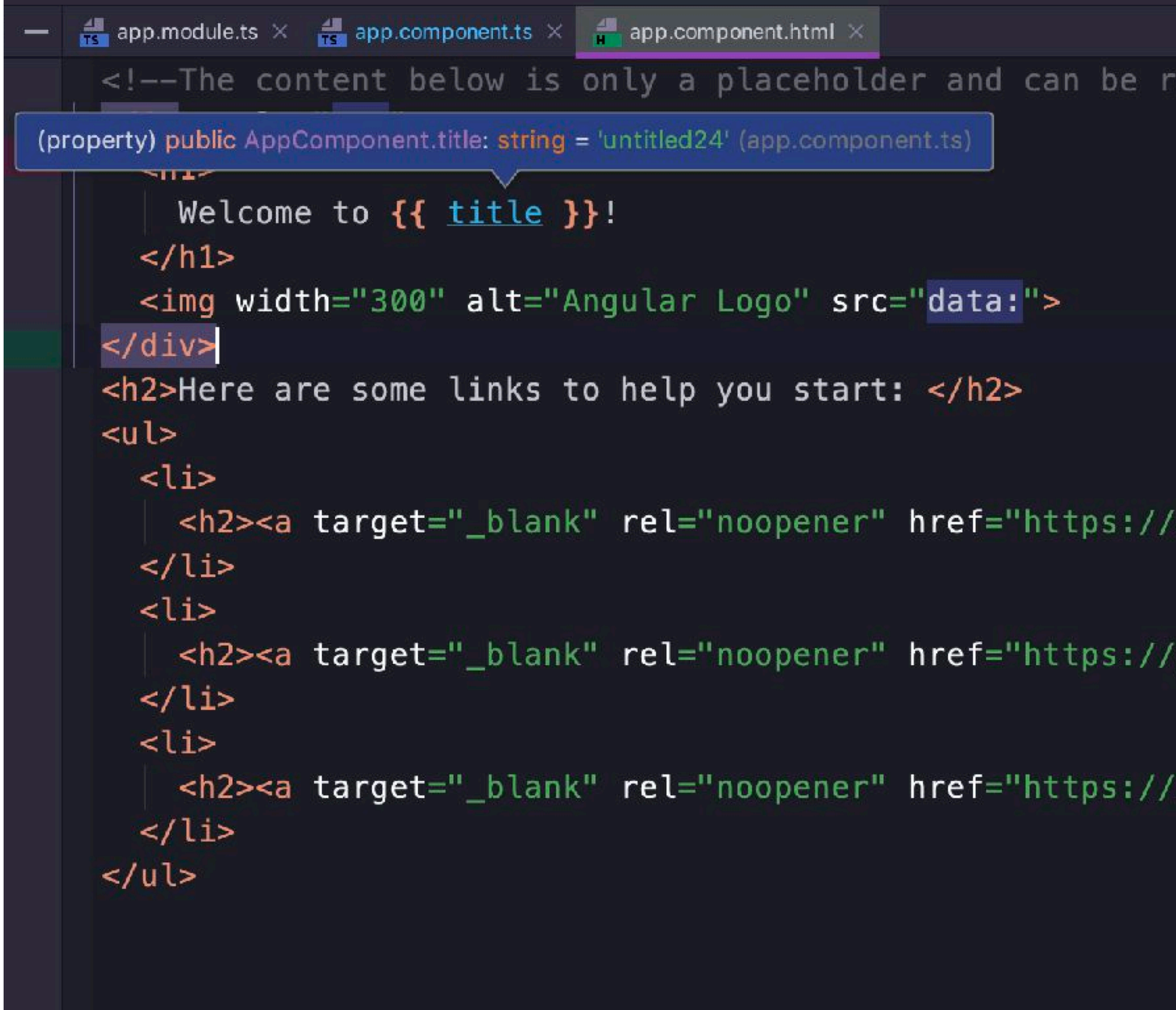
References

Механизм перехода в место декларации?

@JetBrains. All rights reserved

References

- Взять имя файла
- Вычислить ts файл
- Найти класс
- Найти свойство с нужным именем



@JetBrains. All rights reserved

Индексация

@JetBrains. All rights reserved

Индексация

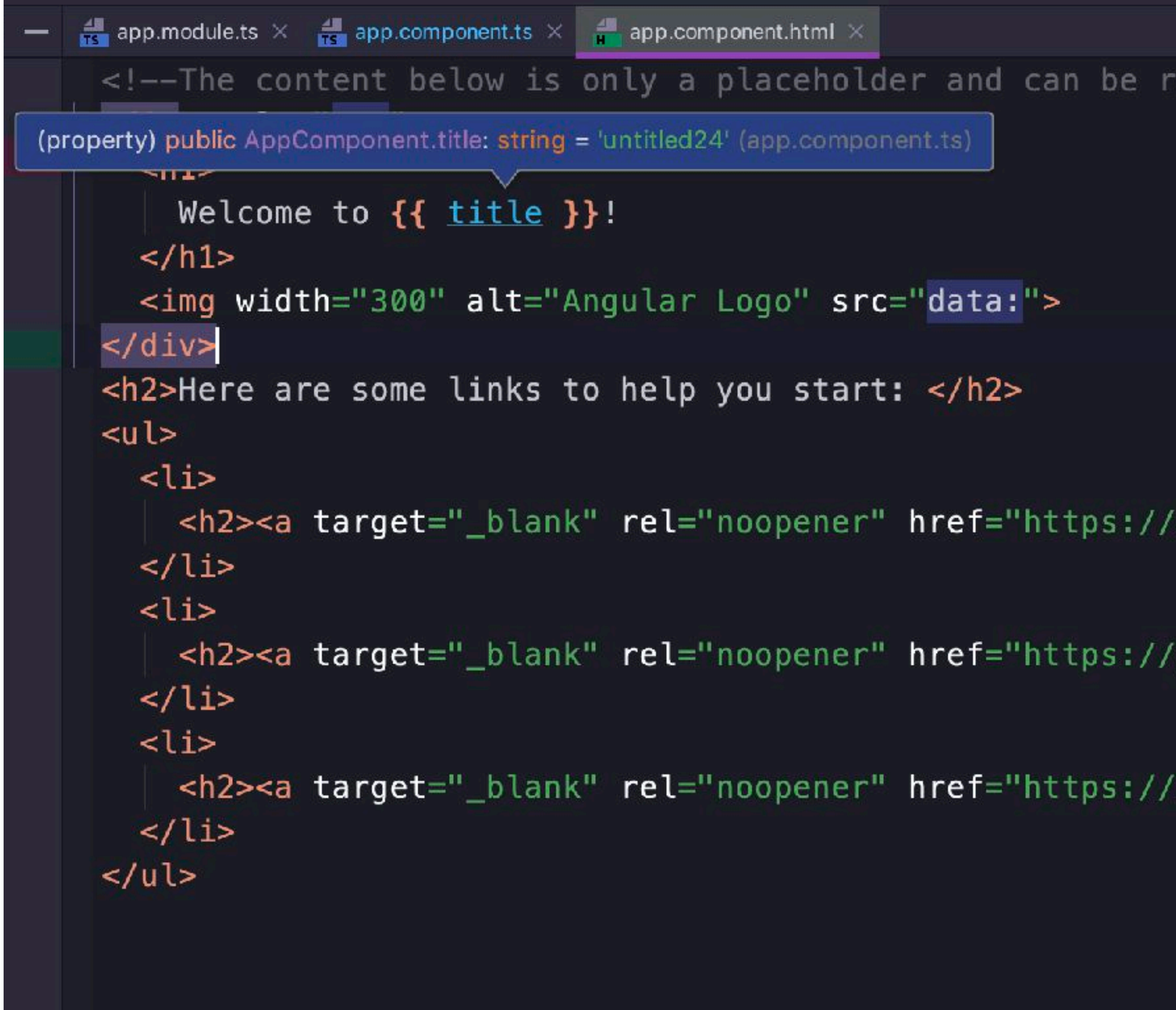
- Один раз при первом открытии проекта проходим по всем файлам
- Строим синтаксическое дерево
- Сохраняем необходимую информацию в индекс

Используем индекс для навигации

@JetBrains. All rights reserved

References

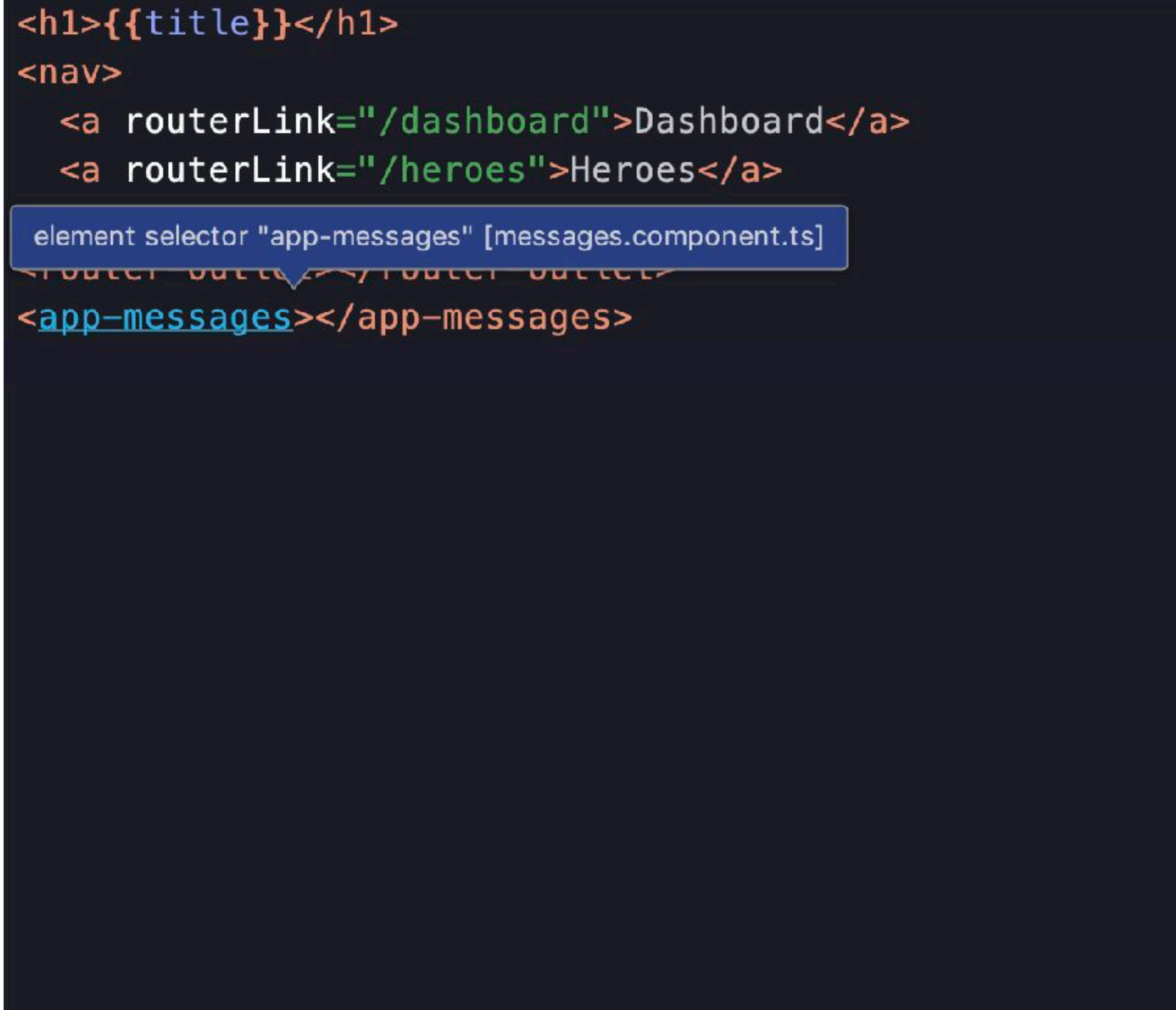
- Взять имя файла
- Найти класс
- Найти свойство с нужным именем



@JetBrains. All rights reserved

References

Навигация в “ts” файл



@JetBrains. All rights reserved

References

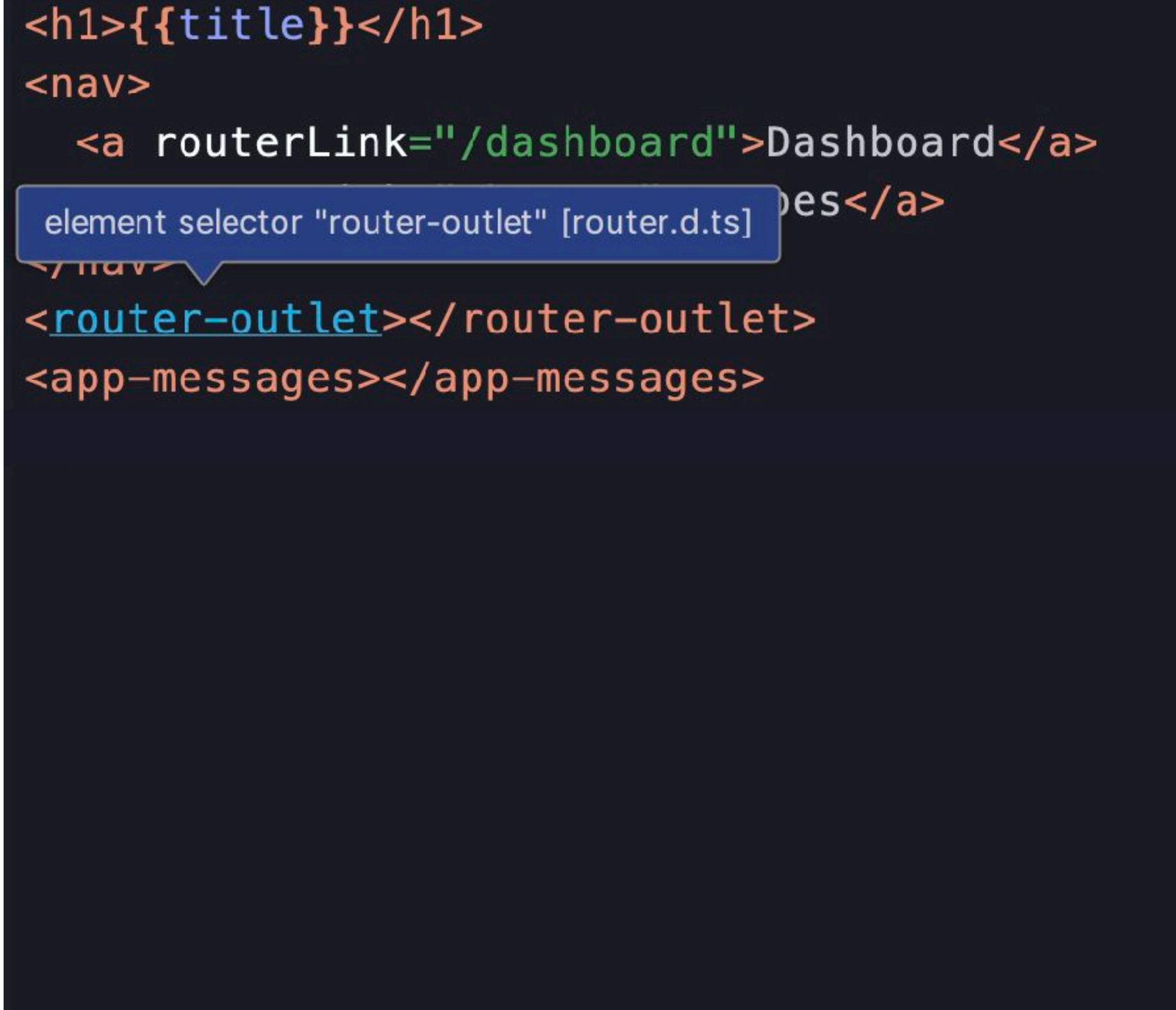
Навигация в “ts” файл

```
@Component({
  selector: 'app-messages',
  templateUrl: './messages.component.html',
  styleUrls: ['./messages.component.css']
})
export class MessagesComponent implements OnInit {

  constructor(public messageService: MessageServic

  ngOnInit() {
  }

}
```

@JetBrains. All rights reserved

References

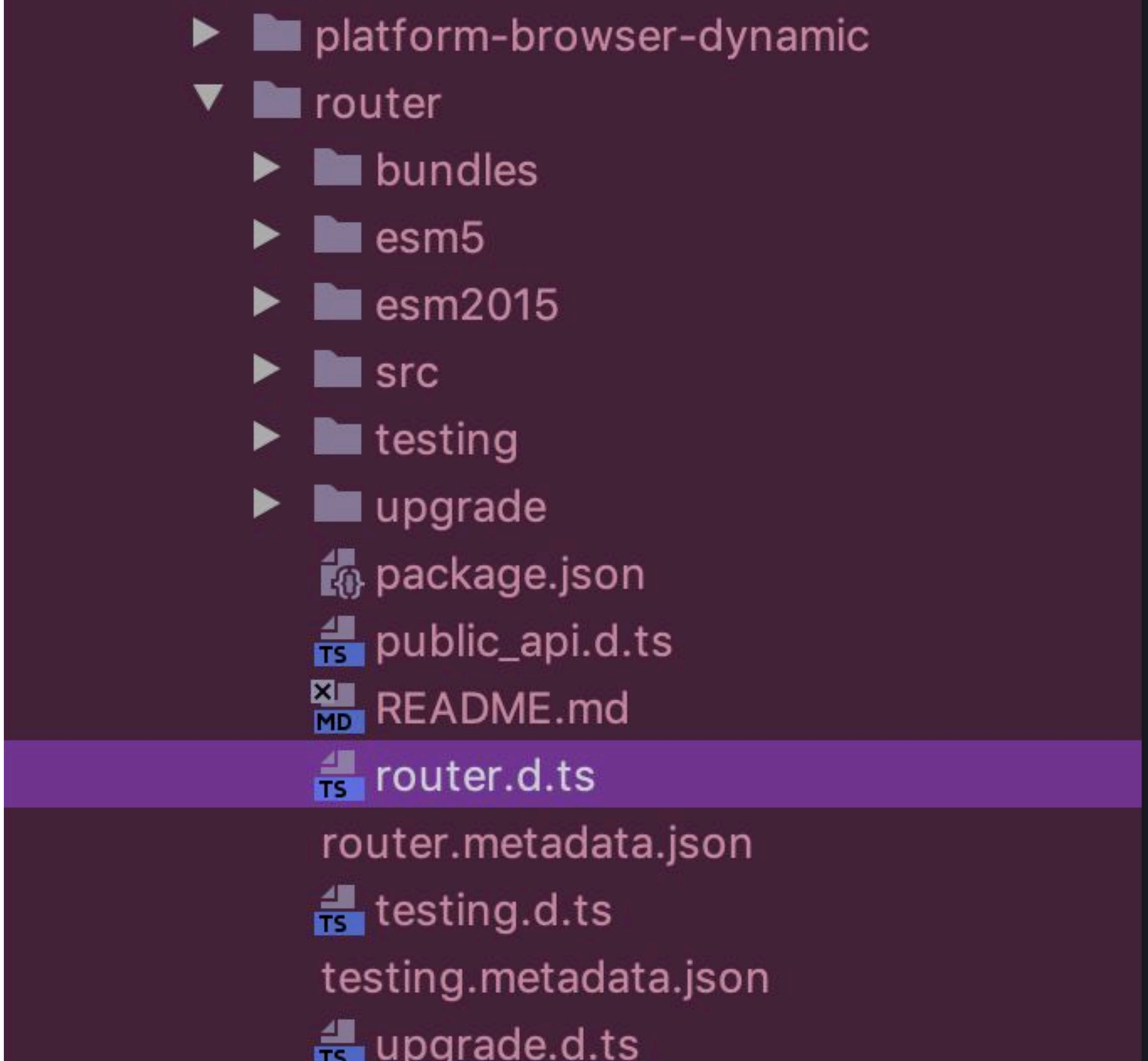
Навигация в “.d.ts” файл



@JetBrains. All rights reserved

References

Навигация в “.d.ts” файл



@JetBrains. All rights reserved

References

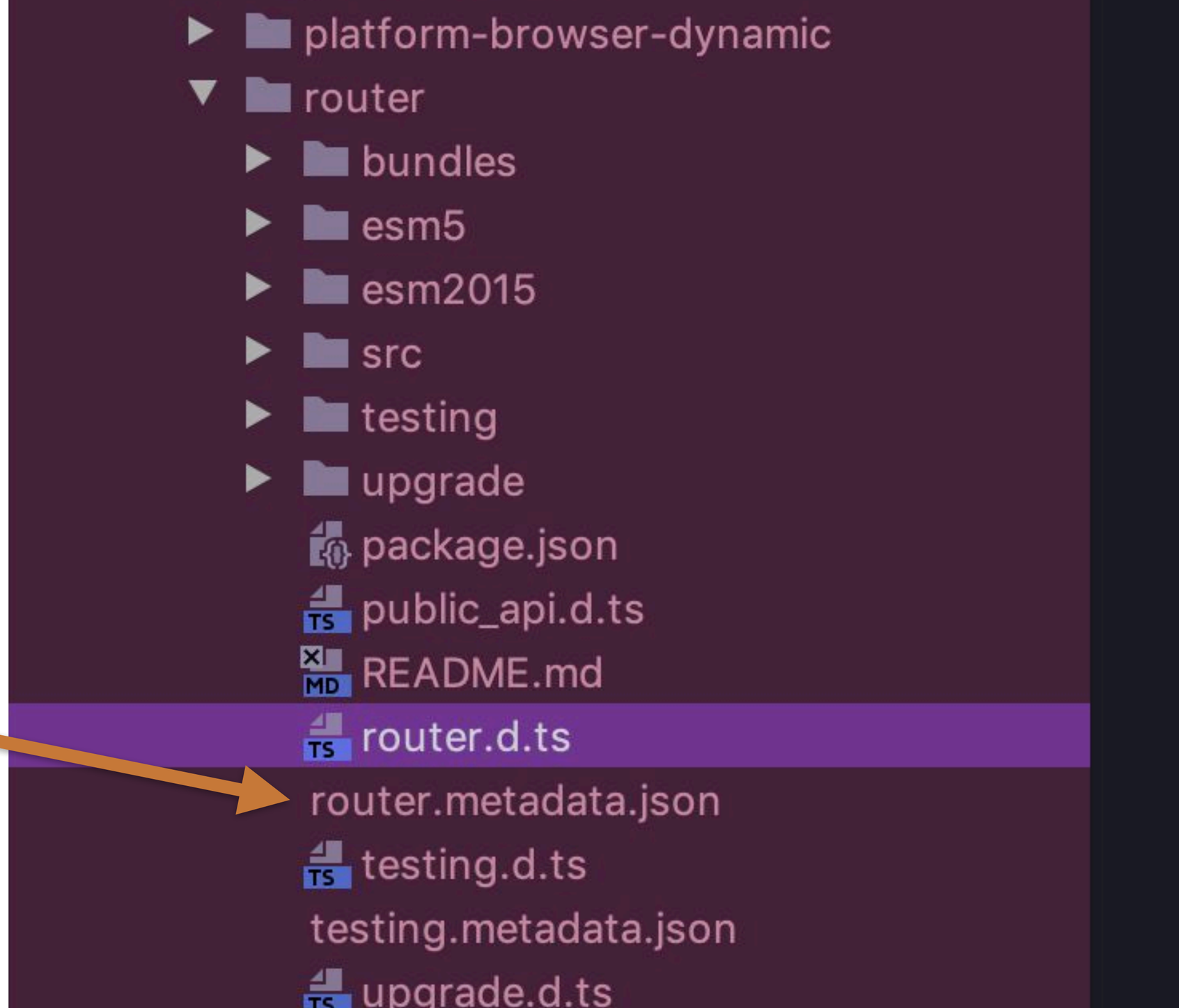
Навигация в “.d.ts” файл

```
 * @publicApi
 */
export declare class RouterOutlet implements OnDestroy, OnInit {
    private parentContexts;
    private location;
    private resolver;
    private changeDetector;
    private activated;
    private _activatedRoute;
    private name;
    activateEvents: EventEmitter<any>;
    deactivateEvents: EventEmitter<any>;
    constructor(parentContexts: ChildrenOutletContexts, location:
    ngOnDestroy(): void;
    ngOnInit(): void;
    readonly isActivated: boolean;
    readonly component: Object;
    readonly activatedRoute: ActivatedRoute;
    readonly activatedRouteData: Data;
    /**
     * Called when the `RouteReuseStrategy` instructs to detach th
     */
    detach(): ComponentRef<any>;
    /**
     * Called when the `RouteReuseStrategy` instructs to re-attach
```

@JetBrains. All rights reserved

References

Навигация в “.d.ts” файл



@JetBrains. All rights reserved

References

@JetBrains. All rights reserved

```
"RouterOutlet": {
  "__symbolic": "class",
  "decorators": [
    {
      "__symbolic": "call",
      "expression": {
        "__symbolic": "reference",
        "module": "@angular/core",
        "name": "Directive",
        "line": 38,
        "character": 1
      },
      "arguments": [
        {
          "selector": "router-outlet",
          "exportAs": "outlet"
        }
      ]
    }
```

Стратегии

- Индексируем сами файлы
- Индексируем связанные файлы (.metadata.json, .js)
- Пре- генерируем информацию (web-types)

@JetBrains. All rights reserved

Что осталось за кадром?

—

- Рефакторинги, Intention’ы, Quick fix’ы
- Data Flow анализ
- Инспекции

@JetBrains. All rights reserved

Ссылки

- Angular / Vue плагины для WS: https://github.com/JetBrains/intellij-plugins
- Доклад про поддержку языков в IDE: https://www.youtube.com/watch?v=aGmygYnwc2s

@JetBrains. All rights reserved

Спасибо за внимание

@anstarovoyt

andrey.starovoyt@jetbrains.com

jetbrains.com

@JetBrains. All rights reserved