

TypeScript 2.9



What's New?

Generics in TSX

Before

```
interface Props<T> { data: T };  
class Component<T> extends React.Component<Props<T>> {}  
  
const ComponentOfNumber = Component as new (props: Props<number>) => Component<number>;  
export const someElement = <ComponentOfNumber data={5} />
```

After

```
interface Props<T> { data: T };  
class Component<T> extends React.Component<Props<T>> {}  
  
export const someElement = <Component<number> data={5} />
```



Generics in TSX

--resolveJsonModule

config.json

```
{ "host": "mysite.com", "port": 5023 }
```

someTs.ts

```
import config from "./config.json";  
config.host;  
config.protocol;
```



--resolveJsonModule

--declarationMap

--declarationMap

- Generates `.d.ts.map` files along side of regular `.d.ts` files

this means:

- Hitting go-to-definition on a declaration from a `.d.ts` file generated with `--declarationMap` will take you to the source file (`.ts`) location where that declaration was defined, and not to the `.d.ts`.



--declarationMap

**Support number and symbol
named properties with keyof and
mapped types**

Numbers in keyof

```
interface StringAndNum {  
  ... "someString": object,  
  ... 5: object,  
}  
  
type K = keyof StringAndNum;  
// in 2.8: "someString" | "5";  
// in 2.9: "someString" | 5;
```



**Support number and symbol
named properties with keyof and
mapped types**

Thanks!

Gauge image credit to
<http://www.optixl.com/gauge/index.php>

Learn more at
<https://github.com/Microsoft/TypeScript/wiki/What's-new-in-TypeScript#typescrip-29>

Created by
Jason Killian
jasonkillian.com

