

TSTS

TS

-
-
-

TS

-
-

-
-
- TS

,

TS

```
interface IResponseData<T>{  
  code: number;  
  message?: string;  
  data: T;  
}
```

data

```
interface User {  
  id: number;  
  name: string;  
  email: string;  
}  
  
// User  
const response: IResponseData<User> = {  
  code: 200,  
  message: "Success",  
  data: {  
    id: 1,  
    name: "xiaoming",  
    email: "xxx@qq.com"  
  }  
};
```

typeinterface

- 1.
- 2.
- 3.

JavaRust

@xxx

mobx@observableNestJS@Controller

```
// target
// propertyKey
// descriptor
function log(target: any, propertyKey: string, descriptor: PropertyDescriptor)
{
  const originalMethod = descriptor.value;

  descriptor.value = function (...args: any[]) {
    console.log(`  ${propertyKey}:    ${JSON.stringify(args)}`);
    const result = originalMethod.apply(this, args);
    console.log(`  ${propertyKey} :    ${JSON.stringify(result)}`);
    return result;
  };

  return descriptor;
}
```

```
class Test {
  @log
  add(a: number, b: number) {
    return a + b;
  }
}

const test = new Test();
console.log(test.add(2, 3)); // 2,35
```

TS

Vue3

TS

any

anyanyTS

any

any

TS

Partial

```
// User
interface User {
  name: string;
  age: number;
}

// PartialUserPartial
type PartialUser = Partial<User>; // { name?: string; age?: number; }
```

- **Partial<T>** T
- **Required<T>** T
- **Readonly<T>** T
- **Record<K, T>** K T
- **Pick<T, K>** T K
- **Omit<T, K>** T K
- **Exclude<T, U>** T U
- **Extract<T, U>** T U
- **NonNullable<T>** T null undefined
- **ReturnType<T>** T

TS

TSDefinitelyTyped

1. d.tsd.ts lodash.d.ts
2. declare module lodash

```
declare module 'lodash' {
  //
}
```

3. `interface typeclass`

4. `export utils`

```
declare module 'lodash' {  
  function utils(...args: any[]): any;  
  export { utils };  
}
```

5. `import`

```
import { utils } from 'lodash';
```

ReactTSVueTS

React

ReactTS

React

- `propsstate`
- `any`
-
- `Hooks`
-

Vue

- `Vue2 class-component` `vue-property-decorator TS`
- `Vue3TSvue-cliviteTS`