

AWAITED

Вспомогательный тип awaited был введен в язык последним и нужен для формирования типа, который возвращает указанный промис:

```
1 type FromPromise = Awaited<Promise<number>>; // number, а не Promise<number>
```

Этот тип так же позволяет **рекурсивно** раскрывать промисы:

```
1 type FromPromise = Awaited<Promise<Promise<number>>>; // number
```

Пример использования для получения типа:

```
1 interface User {
2   name: string;
3 }
4
5 async function fetchUsers(): Promise<User[]> {
6   const users: User[] = [
7     {
8       name: "Alex",
9     },
10  ];
11
12   return users;
13 }
14
15 const users = fetchUsers(); // users type Promise<User[]>
16
17 type FetchUsersReturnType = Awaited<ReturnType<typeof fetchUsers>>; // User[]
```

До версии 4.5 в коде TS вы можете встретить **ручное** создание этого типа:

```
1 type UnwrappedPromise<T> = T extends Promise<infer Return> ? Return : T;
2 type FetchDataReturnType = UnwrappedPromise<ReturnType<typeof fetchUsers>>;
```