

TypeScript の@デコレータを触 ってみた

tomoyahiroe

自己紹介

Name : Tomoya Hiroe

Keyword : bass, soft-tennis, Lumos

Language : JavaScript, TypeScirpt

@decorator とはなんぞや

by tomoyahiroe

nestJS のルーティング処理



```
1 @Controller('tasks')
2 export class TasksController {
3   constructor(private tasksService: TasksService) {}
```

nestjs のエンティティファイル

```
● ● ●  
1  @Entity()  
2  export class Task {  
3      @PrimaryGeneratedColumn('uuid')  
4      id: string;  
5  
6      @Column()  
7      title: string;  
8  
9      @Column()  
10     description: string;  
11  
12     @Column()  
13     status: TaskStatus;  
14 }
```

どうやら、
experimental support らしい？

デコレータ作ってみたい

by tomoyahiroe

ググってみると...

色々あるらしい

- クラスデコレータ
- メソッドデコレータ
- アクセサデコレータ
- プロパティデコレータ
- パラメータデコレータ

クラスデコレータ触ってみた

by tomoyahiroe

デコレータファクトリがこちら



```
1  function classDecorator<T extends { new (...args: any[]): {} }>(
2    constructor: T
3  ) {
4    return class extends constructor {
5      newProperty = "new property";
6      hello = "override";
7    };
8  }
```

デコレートするクラスがこちら



```
1  @classDecorator
2  class Greeter {
3    greeting: string;
4    constructor(message: string) { this.greeting = message; }
5    greet() { return this.greeting; }
6  }
```

以下のコードで出力すると



```
1 const morning = new Greeter("hello");
2 console.log(morning.newProperty);
```

デコレータファクトリで定義した プロパティが認識されてない



```
1  Property 'newProperty' does not exist on type 'Greeter'.
```

そこで、とりま@ts-ignore



```
1 const morning = new Greeter("hello");
2 //@ts-ignore
3 console.log(morning.newProperty);
```

出力結果



```
1 tomoya decorator $ts-node class_decorator_sample.ts
2 new property
```

Lumos Lightning Talks !!!

感想

by tomoyahiroe

- まだ自分で使うことはない
- 初めてリリースされてない言語の機能を触って面白かった
- LT いいわ。好きだわ。みんなの発表聞きたいわ

以上！

参考文献

- [TypeScript の Decorator について – 公式ドキュメント日本語訳](#)
- [TypeScript によるデコレータの基礎と実践](#)
- [もう怖くない TypeScript のデコレータ機能](#)
- [TypeScript をブラウザで実行できる Playground](#)
- [TypeScript の公式ドキュメント](#)
- [Property does not exist のエラー](#)
- [今こそ伝えたい@ts-ignore の魅力](#)