

## Evitando que nossa tarefa pare

Temos o seguinte fragmento de um `gulpfile.js` :

```
gulp.watch('src/less/**/*.less').on('change', function(event) {  
  gulp.src(event.path)  
    .pipe(less())  
    .pipe(gulp.dest('src/css'));  
});
```

Este código faz com que arquivos LESS alterados sejam compilados. Excelente, mas sabemos que se houver um erro de compilação o stream de `less()` lançará um erro fazendo com que nosso serviço `server` termine. Sendo assim, qual das opções abaixo trata corretamente o erro permitindo que nossa tarefa `server` continue rodando?

Qual das opções abaixo

A

```
gulp.watch('src/less/**/*.less').on('change', function(event) {  
  var stream = gulp.src(event.path)  
    .pipe(less().on('error' {  
      console.log('LESS, erro compilação: ' + erro.filename);  
      console.log(erro.message);  
    })))  
    .pipe(gulp.dest('src/css'));  
});
```



```
gulp.watch('src/less/**/*.less').on('change', function(event) {  
  var stream = gulp.src(event.path)  
    .pipe(less().on('error', function(erro) {  
      console.log('LESS, erro compilação: ' + erro.filename);  
      console.log(erro.message);  
    })))  
    .pipe(gulp.dest('src/css'));  
});
```

C

```
gulp.watch('src/less/**/*.less').on('change', function(event) {  
  var stream = gulp.src(event.path)  
    .pipe(less().onError(function(erro) {  
      console.log('LESS, erro compilação: ' + erro.filename);  
      console.log(erro.message);  
    })))  
    .pipe(gulp.dest('src/css'));  
});
```

D

```
gulp.watch('src/less/**/*.less').on('change', function(event) {  
  var stream = gulp.src(event.path)  
    .pipe(less().on(function(erro) {
```

```
    console.log('LESS, erro compilação: ' + erro.filename);  
    console.log(erro.message);  
  }  
})  
  .pipe(gulp.dest('src/css'));  
});
```

A resposta correta é:

```
gulp.watch('src/less/**/*.less').on('change', function(event) {  
  var stream = gulp.src(event.path)  
    .pipe(less().on('error', function(erro) {  
      console.log('LESS, erro compilação: ' + erro.filename);  
      console.log(erro.message);  
    }  
  )  
    .pipe(gulp.dest('src/css'));  
});
```

Podemos adicionar o evento `error` em qualquer Stream através da função `on`. Quando um erro for disparado, no lugar dele terminar nossa tarefa, o código do callback passado para a função `on` será executado. Não fazemos nada demais na função, apenas imprimimos o nome do arquivo e a mensagem de erro para sabermos onde ferimos a sintaxe less.

PRÓXIMA ATIVIDADE