04

BrowserSync e Gulp watcher, a união faz a força

Configurar o BrowserSync apenas indica qual pasta será servida através do navegador pelo pequeno servidor criado por esta ferramenta. Só essa configuração não realiza a mágica do livereloading. Precisamos chamar o método `reload()` do objeto BrowserSync sempre que quisermos que ele atualize a página no navegador. Para essa tarefa usamos `watchers` do Gulp que ao detectarem qualquer mudança nesse ou naquele arquivo, chamará o `reload()` do BrowserSync.

Qual das configurações abaixo configura corretamente um watcher para recarregar o BrowserSync na mudança de qualquer arquivo?

A

```
gulp.task('server', function() {
    browserSync.init({
        server: {
            baseDir: 'src'
        }
    });

    gulp.watch('src/**/*').on('change');
});
```

✓

```
gulp.task('server', function() {
    browserSync.init({
        server: {
            baseDir: 'src'
        }
    });

    gulp.watch('src/**/*').on('change', browserSync.reload);
});
```

C

```
gulp.task('server', function() {
    browserSync.init({
        server: {
            baseDir: 'src'
        }
    });

    gulp.watch('src/**/*').on(browserSync.reload);
});
```

D

```
gulp.task('server', function() {
    browserSync.init({
        server: {
            baseDir: 'src'
        }
    });
```

```
            gulp.watch('src/**/*').on('change', browserSync.reload());
        });
```

A resposta correta é:

```
gulp.task('server', function() {
    browserSync.init({
        server: {
            baseDir: 'src'
        }
    });

    gulp.watch('src/**/*').on('change', browserSync.reload);
});
```

A função `gulp.watch` recebe como primeiro parâmetro o evento que desejamos observar, em nosso caso, estamos interessados apenas nos arquivos que mudaram, por isso passamos `change`. Por fim, o útlimo parâmetro é a função que desejamos executar, nesse caso, passamos diretamente BrowserSync.reload. Como passamos a função e não a invocamos, será o watcher que a chamará toda vez que um arquivo for alterado.

É por isso que não podemos fazer:

```
gulp.watch('src/**/*').on('change', browserSync.reload());
```

Neste exemplo, já estamos pedindo para o BrowserSync executar seu `reload`. Como esta função não retorna nada, nosso `watcher` não terá função alguma para executar. É por isso que devemos passar a função como `browserSync.reload`. Em JavaScript funções podem ser passadas como parâmetro. Uma alternativa mais verbosa é a seguinte:

```
gulp.watch('src/**/*').on('change', function() {
     browserSync.reload();
    });
```

Veja que não passamos mais diretamente a função `browserSync.reload` como parâmetro, em seu lugar, passamos uma função anônima e quando ela for executada pelo nosso `watcher` do Gulp ela chamará `browserSync.reload()`, recarregando assim nossa página. A primeira é menos verbosa que a segunda, mas ambas são válidas.

PRÓXIMA ATIVIDADE