

# Node e TypeScript

O TypeScript é uma linguagem tipada que estende a sintaxe do JavaScript, trazendo uma série de benefícios. Como no final ela é transpilada para JavaScript, seus casos de uso são muitos, desde projetos front-end (inclusive sendo hoje a escolha padrão no framework Angular) quanto back-end (Node.js). Nesse apêndice será explorado como iniciar um projeto Express usando TypeScript ao invés de JavaScript.

O primeiro passo é criar um novo projeto usando `npm`:

```
npm init
```

Instale agora as seguintes dependências:

```
npm i --save-dev typescript nodemon ts-node
npm i express @types/express
```

TypeScript é o suporte para a linguagem em si (juntamente com o compilador `tsc`), `nodemon` será usado para reinicializações automáticas durante o desenvolvimento, e `ts-node` é um utilitário que permite executar o seguinte comando: `ts-node <script.ts>` e cuida internamente das chamadas necessárias ao compilador do TypeScript para que isso funcione da maneira esperada. Express será usado para exemplificar o resultado com uma API simples. Por fim, `@types/**` são pacotes que documentam a tipagem das APIs de bibliotecas JavaScript de modo a melhorar a experiência de uso delas com TypeScript.

Configure agora o TypeScript, com o seguinte comando:

```
tsc --init
```

Isso irá gerar um arquivo chamado `tsconfig.json`. Para esse apêndice nada precisa ser alterado, e está fora do escopo a análise de cada opção (no entanto é muito útil para entender até onde se propõe chegar a linguagem).

Com isso já é possível criar o arquivo de entrada da aplicação, `index.ts` (note a extensão):

```
import express from 'express';

import { extrairNome } from './lib';

const app = express();

app.get('/', (req, res) => {
  res.send(`Olá, ${extrairNome(req) || 'anônimo'}!`);
});

app.listen(3000);
```

E o arquivo `lib.ts` (note o uso de tipagem do TypeScript):

```
import { Request } from "express";

export function extrairNome(req: Request): string {
```

```
    return req.query.nome;
}
```

Execute dessa maneira:

```
nodemon -e ts ./node_modules/.bin/ts-node index.ts
```

Para facilitar, adicione esse comando como um script no `package.json` :

```
{
  // ...
  "scripts": {
    "dev": "nodemon -e ts ./node_modules/.bin/ts-node index.ts"
  },
  // ...
}
```

Isso permite que ele seja executado da seguinte forma:

```
npm run dev
```

Por fim, como gerar uma build para ser usada em produção? Basta usar o compilador `tsc` diretamente. Mas antes, indique no arquivo `tsconfig.json` um diretório específico para a geração do código JavaScript, caso contrário ele ficará junto com o código-fonte, o que não é desejado:

```
{
  // ...
  "outDir": "./dist",
  // ...
}
```

E agora execute o comando abaixo:

```
./node_modules/.bin/tsc
```

Para executar usando essa build basta usar o seguinte comando:

```
node dist/index.js
```

Note que o `ts-node` não faz mais parte do processo.