

Angular 2 Conceitos Básicos

Rafael Gonçalves de Oliveira Viana¹

¹Sistemas de Informação – Universidade Federal do Mato Grosso do Sul (UFMS)
Caixa Postal 79400-000 – Coxim – MS – Brazil

rafael.viana@aluno.ufms.br

Abstract. This article describes how to build an Angular 2 application, use Angular-CLI, which allows administration of application components, offering better performance and speed in the production of Front End APIs.

Resumo. Este artigo descreve como construir uma aplicação Angular 2, utilizando o Angular-CLI, que permite administrar os componentes da aplicação praticamente, tendo um melhor desempenho e velocidade na produção de APIs Front-End.

1. Angular 2

Angular 2 é uma estrutura mais eficiente que permite que os programadores se concentrem apenas na criação de classes de JavaScript. As visualizações e controladores do Angular 1.x são substituídos por componentes, que podem ser descritos como uma versão refinada de diretivas.

Até mesmo os programadores angulares experientes nem sempre estão conscientes de todas as capacidades das diretivas Angular 1.x. Os componentes Angular 2 são consideravelmente mais fáceis de ler, e sua API possui menos jargão do que as diretivas Angular 1.x. Além disso, para ajudar a facilitar a transição para o Angular 2, a equipe Angular adicionou um `.component` método ao Angular 1.5, que foi apoiado pelo membro da comunidade Todd Motto para a v1.3. A nova base de código angular é mais moderna, mais capaz e mais fácil para os novos programadores aprender do que Angular 1.x, ao mesmo tempo em que é mais fácil para os veteranos de projetos trabalharem. Com o Angular 1, os programadores tiveram que entender as diferenças entre controladores, serviços, fábricas, fornecedores e outros conceitos que poderiam ser confusos, especialmente para novos programadores. Apesar de ser uma reescrita completa, o Angular 2 manteve muitos dos seus principais conceitos e convenções com Angular 1.x, por exemplo, uma implementação simplificada, de injeção de dependência. [?]

2. TypeScript

Angular 2 foi escrito em TypeScript, um superconjunto de JavaScript que implementa muitos novos recursos ES2016 +. Ao se concentrar em tornar a estrutura mais fácil para os computadores processarem, Angular 2 permite um ecossistema de desenvolvimento muito mais rico. Os programadores que usam editores de texto sofisticados (ou IDEs) notarão melhorias dramáticas com auto-complete e sugestões de tipo. Essas melhorias ajudam a reduzir o ônus cognitivo da aprendizagem Angular 2. Felizmente para os programadores JavaScript tradicionais do ES5 isso não significa

que o desenvolvimento deve ser feito em TypeScript ou ES2015: os programadores ainda podem escrever o JavaScript vanilla que funciona sem transpilação.

Uma vez que a maioria dos navegadores não estão habilitados para rodar ES6 e ES7, surgiram alguns pré-compiladores, que geram todo o código para o JavaScript “entendível” pelo navegador. Mas o Typescript vai um pouco mais longe.

O TypeScript, criado pela Microsoft, é um “_superset_” do JavaScript, que, além de implementar as funcionalidades do ES6+, traz uma série de “poderes” no desenvolvimento. A capacidade de autocomplete nas IDEs é impressionante. Mas o mais interessante é a parte de organização do código. O TypeScript tem uma sintaxe muito mais clara e fácil de entender.

A primeira iteração da Angular forneceu programadores web com uma estrutura altamente flexível para o desenvolvimento de aplicativos. Esta foi uma mudança dramática para muitos programadores da web, e enquanto essa estrutura era útil, tornou-se evidente que muitas vezes era muito flexível. Ao longo do tempo, as melhores práticas evoluíram, e uma estrutura orientada pela comunidade foi endossada. Angular 1.x tentou trabalhar em torno de várias limitações do navegador relacionadas ao JavaScript. Isto foi feito através da introdução de um sistema de módulos que utilizou a injeção de dependência. Este sistema era novo, mas infelizmente tinha problemas com as ferramentas, como na minificação e análise estática.

O Angular 2.x utiliza o sistema do módulo ES2015 e as modernas ferramentas de embalagem como o webpack, SystemJS e atualmente o Angular-CLI. Os módulos são muito menos acoplados ao “caminho angular”, e é mais fácil escrever JavaScript mais genérico e conectá-lo ao Angular. A remoção de soluções alternativas de minificação e a adição de prescrições rígidas tornam a manutenção das aplicações existentes mais simples. O novo sistema de módulos também facilita o desenvolvimento de ferramentas eficazes que podem justificar melhor os projetos maiores.[?]

3. Mobile

Angular 2 foi projetado para o celular desde do seu início. Além do poder de processamento limitado, os dispositivos móveis possuem outros recursos e limitações que os separam dos computadores tradicionais. As interfaces de toque, as propriedades de tela limitada por esse motivo o hardware móvel foram considerados no Angular 2. Os computadores de mesa também tiveram melhorias dramáticas no desempenho e na capacidade de resposta. O Angular 2, como React e outras estruturas modernas, pode aproveitar os ganhos de desempenho ao renderizar HTML no servidor ou mesmo em um trabalhador da Web. A busca pelo desempenho não termina com pré-renderização. O Angular 2 torna-se portátil para o celular nativo, integrando-se ao NativeScript, uma biblioteca de código aberto que engata o JavaScript e o celular. Além disso, a equipe Ionic está trabalhando em uma versão Angular 2 de seu produto, fornecendo outra maneira de alavancar recursos nativos dos dispositivos com Angular 2.

Se você pretende desenvolver aplicativos híbridos, cá está mais um excelente motivo para usar Angular 2, a equipe do Ionic está finalizando o desenvolvimento da sua segunda versão, que é totalmente escrita em Angular 2.

4. Node JS

Node.js é uma plataforma para desenvolvimento de aplicações server-side baseada no V8, o interpretador de JavaScript open source implementado pelo Google em C++ e utilizado pelo Chrome. O que sem dúvidas gera uma grande expectativa em relação ao desempenho do Node.js. Sendo assim utilizando JavaScript e o interpretador V8 JavaScript podemos criar uma variedade de aplicações Web utilizando apenas código em JavaScript.

5. Angular Cli

Iniciar e gerenciar um projeto em Angular 2 pode não ser uma tarefa tão simples. A existência de dezenas de bibliotecas, frameworks e ferramentas, que muitas vezes tentam resolver o mesmo problema, pode ser uma barreira aos primeiros contatos com a tecnologia.

Ao observar esse problema, a equipe do Angular criou uma ferramenta de linha de comando chamada Angular CLI (Command Line Interface - Interface de Linha de Comando) cujo objetivo principal é facilitar o gerenciamento de projetos escritos nesse framework.

Com o Node.JS instalado, temos à disposição o comando npm, necessário para instalar o Angular CLI. Neste ponto, será preciso abrir o terminal do seu sistema operacional e digitar a instrução:

```
npm install -g @angular/cli
```

Após a instalação podemos ver se está tudo correto executando o comando:

```
ng help
```

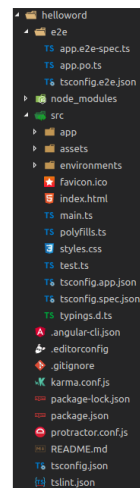
Agora para podemos começar a trabalhar com o Angular 2, basta criar um projeto com o comando de inicialização do Angular-CLI:

```
ng new nomedoprojeto
```

Após a criação do projeto com o comando de inicialização do Angular-CLI demonstrado na Figura 5, você terá um projeto, "Hello World" como na Figura 5, pronto para ser executado:

```
[rafael@ ~]$ ng new helloworld
Installing ng
create .editorconfig
create README.md
create src/app/app.component.css
create src/app/app.component.html
create src/app/app.component.spec.ts
create src/app/app.component.ts
create src/app/app.module.ts
create src/assets/.gitkeep
create src/environments/environment.prod.ts
create src/environments/environment.ts
create src/favicon.ico
create src/index.html
create src/main.ts
create src/polyfills.ts
create src/styles.css
create src/test.ts
create src/tsconfig.app.json
create src/tsconfig.spec.json
create src/typings.d.ts
create .angular-cli.json
create e2e/app.e2e-spec.ts
create e2e/app.po.ts
create e2e/tsconfig.e2e.json
create .gitignore
create karma.conf.js
create package.json
create protractor.conf.js
create tsconfig.json
create tslint.json
Successfully initialized git.
Installing packages for tooling via npm.
Installed packages for tooling via npm.
Project 'helloworld' successfully created.
[rafael@ ~]$
```

(a) Comando ng new



(b) Projeto

Figura 1. Criação do projeto, Angular 2 com Angular-CLI