

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 meta-paper describes the style to be used in articles and short papers for SBC conferences. For papers in English, you should add just an abstract while for the papers in Portuguese, we also ask for an abstract in Portuguese (“resumo”). In both cases, abstracts should not have more than 10 lines and must be in the first page of the paper.

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 1 x Angular 2

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. 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.

Angular 2 é uma estrutura mais eficiente que permite que os programadores se concentrem simplesmente na criação de classes de JavaScript. As visualizações e controladores são substituídos por componentes, que podem ser descritos como uma versão refinada de diretrizes. 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 diretrizes 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 . . 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, “Native JS” de injeção de dependência. Isso significa que os programadores que já são proficientes com o Angular terão um tempo mais fácil migrando para o Angular 2 do que outra biblioteca como Reagir ou estrutura como a Ember.

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-conclusão 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 da baunilha 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 (isso mesmo), é um “_superset_” do JavaScript, que, além de implementar as funcionalidades do ES6+, traz uma série de “poderes” no desenvolvimento. Uma das coisas que eu gosto bastante, é a capacidade de autocomplete nas IDEs (se você tiver uma que suporte, como o Sublime Text ou VSCode). Mas acredito que o mais interessante é a parte de organização do código. O TypeScript tem uma sintaxe muito mais clara e fácil de entender. Abaixo um mesmo código escrito em TypeScript e JavaScript:

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 ferramental, notadamente minificação e análise estática.

O Angular 2.x utiliza o sistema do módulo ES2015 e as modernas ferramentas de embalagem como o webpack ou o SystemJS. 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 eficientes que podem justificar melhor os projetos maiores.

3. Mobile

Angular 2 foi projetado para o celular desde o 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 e o hardware móvel foram considerados em Angular 2. Os computadores de mesa também verão melhorias dramáticas no desempenho e na capacidade

de resposta. O Angular 2, como Reagir e outras estruturas modernas, pode aproveitar os ganhos de desempenho ao renderizar HTML no servidor ou mesmo em um trabalhador da Web. Dependendo design do aplicativo / site esta Isomorphic renderização pode fazer a experiência do usuário se sentir ainda mais instantânea. 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 do dispositivo com Angular.

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.

Referências

- Boulic, R. and Renault, O. (1991). 3d hierarchies for animation. In Magnenat-Thalmann, N. and Thalmann, D., editors, *New Trends in Animation and Visualization*. John Wiley & Sons ltd.
- Knuth, D. E. (1984). *The T_EX Book*. Addison-Wesley, 15th edition.
- Smith, A. and Jones, B. (1999). On the complexity of computing. In Smith-Jones, A. B., editor, *Advances in Computer Science*, pages 555–566. Publishing Press.