

AngularJS Introduction

AngularJS는?

AngularJS는 동적 Web 어플리케이션을 위한 프레임 워크입니다. 템플릿 언어로 HTML을 사용하여 응용 프로그램의 구성 요소를 알기 쉽게 간결하게 하기 위해 HTML 문법을 확장합니다. AngularJS를 사용하여 데이터 바인딩 의존 주입을 통해 대부분의 불필요한 코드를 제거하고 브라우저의 JavaScript와 서버 기술이 이상적인 파트너가 될 수 있도록 도움을 줍니다.

AngularJS는 HTML이 가진 특징을 바탕으로, 그것을 응용 프로그램 용으로 사용하기 위한 설계가 되어 있습니다. HTML은 정적 문서를 위한 뛰어난 언어이지만, 응용 프로그램을 구축하기 위한 방법은 포함되어 있지 않습니다. 이러한 동적 응용 프로그램 및 정적 문서 사이에 존재하는 불일치를 해소하기 위해 다음과 같은 기능을 가지고 있습니다.

라이브러리

Web 응용 프로그램을 작성할 때 매우 유용한 기능을 모아 놓은 컬렉션. 코드에서 개발자가 필요에 따라 라이브러리의 적합한 함수를 호출합니다.

(예: jQuery)

프레임 워크

개발자의 코드의 세부 사항을 포함하는 Web 어플리케이션 특유의 구현. 프레임 워크는 애플리케이션 자체 용도에 맞게 개발자의 코드를 호출합니다. (예: knockout, ember, 기타)

이와는 달리 AngularJS는 다른 접근 방식을 취합니다. AngularJS는 HTML 중심의 문서와 응용 프로그램이 필요에 따라 생성하는 새로운 HTML 사이의 불일치를 최대한 줄이기 위해 노력하고 있습니다. AngularJS는 우리 (AngularJS 개발팀)이 "지시문 (Directive)"라고 부르는 방법을 통해 브라우저에 새로운 문법을 알려줍니다. 이러한 새로운 문법에는 다음과 같은 것이 있습니다.

- {{ }}에 의한 데이터 바인딩.
- DOM의 부분 반복 / 숨기기 위한, DOM 제어 구조
- 양식 및 양식 입력 값의 검증 지원
- DOM 요소에 대한 코드 바인딩 할당
- 재사용 가능한 구성 요소에 대한 HTML 그룹화

클라이언트 측 해결책

AngularJS는 Web 어플리케이션의 클라이언트 사이드를 구축하는 전체 퍼즐의 한 조각이 아닙니다. 명확하게 정의된 구조에 Ajax 제휴 코드를 작성하고 그 코드와 DOM을 처리합니다. 따라서 AngularJS는 CRUD(Create, Read, Update, Delete)의 기본 데이터 처리 기능) 애플리케이션을 구축

방법에 대해 집착적인 경향이 있습니다. 하지만 그 집착적인 부분을 확인하려고 하는 것이 그것을 자유롭게 변경할 수 있는 첫 걸음이라 할 수 있습니다. AngularJS을 시작함에 있어 다음 사항을 염두에 두어야 합니다.

- 개발자는 데이터 바인딩 (템플릿) 지시어 폼 검증, 라우팅, 딥 링크, 재사용 가능한 구성 요소 종속성 주입, 구조 속에서 CRUD 애플리케이션을 구축해야 합니다.
- 테스트 기능으로, 단위 테스트, end-to-end 테스트 모의 테스트 장치가 준비되어 있습니다.
- Application 을 만들기 시작하는데 있어 출발점으로 디렉터리 레이아웃과 테스트 스크립트를 제공합니다.

AngularJS의 사용 장소

AngularJS 높은 계층의 추상화를 개발자에게 제공함으로써 애플리케이션의 개발을 단순화합니다. 그러나 추상화는 유연성을 희생하는 일면도 있기 때문에 모든 응용 프로그램이 AngularJS에 적합하지는 않습니다. AngularJS는 CRUD 애플리케이션 구축을 염두에 두고 만들어진 것으로, 다행히 CRUD 애플리케이션이 Web 애플리케이션의 대부분을 차지하고 있습니다. AngularJS이 어떤 응용 프로그램에서 사용하는 것이 적절한 지, 또한 부적절한 방법을 이해하는 것이 중요합니다.

JavaScript 및 HTML 의한 게임이나 GUI 편집기는 강렬하면서도 복잡한 DOM 조작이 필요한 응용 프로그램의 예입니다. 이러한 종류의 응용 프로그램은 CRUD 애플리케이션과는 크게 달리 AngularJS에는 적합 하지 않습니다. 이러한 경우에는 jQuery와 같은 낮은 계층의 추상화를 할 라이브러리를 사용하는 것이 좋습니다.

AngularJS의 사상

Angular는 UI 구축 및 소프트웨어 구성 요소를 동시에 구성 할 경우, 명령형 코드가 비즈니스 로직을 표현하는 데 탁월하지만, 명령형 코드보다 선언 코드가 더 우수하다는 신념 하에 구축되어 있습니다.

- 이것은 응용 프로그램 논리에서 DOM 조작을 분리 한 아주 좋은 아이디어입니다. 또한 코드의 테스트에서도 장점이 많습니다.
- 응용 프로그램 테스트 응용 프로그램을 작성할 만큼 중요하다고 생각하는 것은 정말 정말 중요한 것입니다. 테스트 용이성은 코드의 구축 방법에 따라 많은 영향을 받습니다.
- 이러한 응용 프로그램의 클라이언트 측과 서버 측을 분리하는 좋은 생각입니다. 이렇게 하면 개발 작업을 병행하여 진행하는 것을 허용하고 쌍방의 재사용을 가능하게 합니다.
- 만약 프레임 워크의 응용 프로그램 구축 가이드에 따라 개발자가 UI 디자인, 비즈니스 로직 테스트 It is very helpful indeed if the framework guides developers through the entire journey of building an app : from designing the UI, through writing the business logic, to testing. It is always good to make common tasks trivial and difficult tasks possible.

Angular는 다음과 같은 것에서 개발자에게 제공합니다

콜백 등록

콜백 등록은 코드를 어지럽히고 코드의 움직임을 확인하는 것을 어렵게 합니다. 이러한 콜백 등의 일반적인 정형 코드를 삭제하는 것은 좋은 방법으로, Angular는 하지 않으면 안 되는 많은 JavaScript 코딩을 대폭 줄이고 애플리케이션이 무엇을 하고 있는지를 확인하기 쉽게 해 줍니다.

프로그램에 기초한 HTML의 DOM 조작

HTML의 DOM 조작은 Ajax 애플리케이션의 핵심이지만, 개발자에게 짐이며 오류의 원인이 되기 쉬운 것입니다. Angular는 어떻게 UI가 애플리케이션의 상태에 따라 변화 하는지를 선언으로 지정함으로써 낮은 계층의 DOM 조작 작업을 개발자에게 열어줍니다. Angular에서 응용 프로그램을 쓰면 대부분의 경우 DOM 조작을 프로그램으로 작성하지 않고 모든 작업을 구현하는 것이 가능합니다.

UI에서 데이터를 수집

CRUD 작업은 Ajax 애플리케이션의 대부분을 차지하고 있습니다. 서버로부터 받은 데이터의 집합을 HTML 양식의 내부 개체에 옮겨 사용자에게 양식의 편집을 허용하고 폼 값을 확인하고, 유효성 검사 오류를 표시하며, 내부 모델에 값을 반환하고 서버에 그것을 반환 등 흐름 등의 많은 작업을 정형화된 코드로 만들어 버립니다. Angular는 정형화된 코드를 많이 제거하여, 모든 것을 상세하게 구현하는 것보다 응용 프로그램의 전체 흐름을 그대로 코드에 반영합니다.

개발을 시작하기 위하여 대량의 초기화 코드를 작성

일반적으로 "Hello Word"를 표시하는 Ajax 응용 프로그램을 만드는 것도, 토대가 되는 코드를 많이 작성해야 합니다. Angular는 [Guice](#) 와 같은 의존 주입 (DI) 형식의 자동 주입을 사용하여 응용 프로그램을 쉽게 만들고 시작하는 것을 가능하게 해줍니다. 이를 통해 빠르게 기능 개발을 시작할 수 있게 더욱 자동화 된 테스트에서 초기화 프로세스 제어를 완벽하게 얻을 수 있습니다.