AngularJS — JavaScript-framework с открытым исходным кодом. Предназначен для разработки одностраничных приложений. Его цель — расширение браузерных приложений на основе MVC-шаблона, а также упрощение тестирования и разработки.

Это не совсем так, AngularJS это совокупность ПО и среды в которой выполняется разработка.

AngularJS - это интерфейсная часть стека MEAN, состоящая из базы данных MongoDB, серверного фреймворка Express.js, самого Angular.js и среды выполнения сервера Node.js.

**MVC.**

Angular не является классическим шаблоном MVC, он больше похож на MVVM (Model View ViewModel). Одно из главных различий между MVC и MVVM заключается в том, что в MVVM логика представления (UI) вытаскивается из контроллера в так называемом ViewModel, тогда как логика данных перемещается в часть модели.

**История разработки.**

Angular - разработан в 2009 году Мишко Хевери и Адамом Абронсом в Brat Tech LLC как программное обеспечение позади сервиса хранения JSON-данных, измеряющихся мегабайтами, для облегчения разработки корпоративных приложений. Сервис располагался на домене «GetAngular.com» и имел нескольких зарегистрированных пользователей, прежде чем они решили отказаться от идеи бизнеса и выпустить Angular как библиотеку с открытым исходным кодом.

Абронс покинул проект, но Хевери, работающий в Google, продолжает развивать и поддерживать библиотеку с другими сотрудниками Google Игорем Минаром и Войта Джином.

AngularJS, на данный момент в основном поддерживаемая Google и сообществом частных лиц и корпораций для решения многих проблем.

**Философия Angular**

AngularJS спроектирован с убеждением, что декларативное программирование лучше всего подходит для построения пользовательских интерфейсов и описания программных компонентов, в то время как императивное программирование отлично подходит для описания бизнес-логики. Фреймворк адаптирует и расширяет традиционный HTML, чтобы обеспечить двустороннюю привязку данных для динамического контента, что позволяет автоматически синхронизировать модель и представление. В результате AngularJS уменьшает роль DOM-манипуляций и улучшает тестируемость.

**Цели разработки**

- Отделение DOM-манипуляции от логики приложения, что улучшает тестируемость кода.

- Отношение к тестированию как к важной части разработки. Сложность тестирования напрямую зависит от структурированности кода.

- Разделение клиентской и серверной стороны, что позволяет вести разработку параллельно.

- Проведение разработчика через весь путь создания приложения: от проектирования пользовательского интерфейса, через написание бизнес-логики, к тестированию.

- Angular придерживается MVC-шаблона проектирования и поощряет слабую связь между представлением, данными и логикой компонентов. Используя внедрение зависимости, Angular переносит на клиентскую сторону такие классические серверные службы, как видозависимые контроллеры. Следовательно, уменьшается нагрузка на сервер и веб-приложение становится легче.

**Angular 2, Angular 4, что присходит?**

В октябре 2014 года на конференции ng-Europe разработчиками веб-платформы Google AngularJS в качестве языка, на котором будет построен предстоящий Angular 2.0 был представлен AtScript.

Первоначально AtScript предназначался для работы поверх MachScript, включая некоторые функции от Dart. В октябре 2014 года Google объявила, что Angular 2.0 будет написана на AtScript. В марте 2015 года Microsoft объявила, что многие функции AtScript будут реализованы в Releasecript 1.5, а Angular 2.0 - на чистом Typescript тексте.

Название «AtScript» происходит от символа @ «at», используемого для аннотаций на многих языках. AtScript - это скриптовый язык на основе JavaScript, расширяющий «TypeScript от Microsoft» и транскодирование в JavaScript.

Разработка Angular 2 началась в начале 2015 года.

В начале 2016-го фреймворк пошел в бету, а в мае 2016 пришло радостное известие: вышел Angular 2 Release Candidate 1.

**Angular 4 а где Angular 3?**

Резкий скачек версий был из-за крупных изминений в фреймворке и в устройсте взаимодействия между програмным обеспечением стека MEAN.

Разработчики следовали спецификации семантического версионирования SEMVER.

Angular 4 Март 2017

Angular 5 Сентябрь 2017

Angular 6 Март 2018

Angular 7 Сентябрь 2018

Термин «Angular 2» будет устаревшим с выходом 4-ой или 5-ой версии. Тем не менее, теперь следует начать называть его просто «Angular» без версии.

Кроме того, вы должны начать избегать библиотек с GitHub/NPM с префиксом **ng2-** или **angular2**.

**Статистика (на сентябрь 2016 года)**

Согласно аналитическому сервису JavaScript Libscore, AngularJS используется на сайтах Wolfram Alpha, NBC, Walgreens, Intel, Sprint, ABC News и приблизительно 12 000 других сайтов из 1 миллиона, протестированных в октябре 2016 года. (1.5%)

AngularJS - шестой по счету самый популярный проект всех времен на GitHub.

1.2 миллиона разработчиков использующих Angular 1

623 тысячи уже используют Angular 2

Если Angular 1 был монолитным проектом, то теперь Angular 2 следует рассматривать как платформу.

**Angular 2 быстрее**

Изначально Angular 1 весил 56K. При этом ядро самого Angular 2 было — 170K. Angular Team оптимизировали тонны кода и теперь ядро Angular 2 весит 45K.

**Отложенная загрузка**

Основная задача Angular Team была в том, чтобы добиться автоматической отложенной загрузки. Это гарантирует нашим приложениям только то, что необходимо для текущей страницы. Ленивая загрузка также была включена при маршрутизации в приложениях.

**Быстрый рендеринг**

По сравнению с первой версией приложения на Angular 2 работают в 2,5 раза быстрее, а перерисовка в 4,2 раза быстрее. Команда Angular считает, что она может сделать рендеринг еще более быстрым.

**Топ категорий сайтов на Angular**

**Топ местоположений бизнеса использующего Angular**

**Топ стран использующих Angular**

AngularJS разработчики используют фреймворк не только для создания веб-сайтов социального направления и аналогов Google Docs. Сфера применения AngularJS гораздо шире, о чем свидетельствуют примеры ниже. Разработка веб-приложений с использованием AngularJS становится все более и более популярной, о чем свидетельствует стремительный рост количества сайтов, использующих этот фреймворк, зафиксированный компанией BuiltWith:

**AngularJS версии 1 и Angular 2** - это два совершенно разных продукта.

(популярное сравнение Java и JavaScript).

Большой проблемой приложений на Angular 1 было то, что они собирались только в браузере. Это означает, что ваши приложения были совершенно SEO-неориентированные.

Команда Angular вынесли основной компонент сборки, и теперь ваши приложения можно собирать где угодно, в том числе и на сервере.

Достаточно легко это сделать при помощи NodeJS для сборки шаблонов на стороне сервера.

**Angular, а не React. O React.**

- Нет сущности для событий. Т.е. компоненты могут общаться только с помощью коллбеков, которые вверх всплывают.

- Нет Dependency Injection (DI) из коробки. Из-за этого приходится немного костылей вставлять в виде High Order Component (A higher-order component)

- Нет типизации на уровне языка для описания интерфейсов компонентов (только встроенная через propTypes и возможность привернуть Flow).

- JSX — на тему которого есть много мнений за и против, т.е. смешивать ли вёрстку и код или нет.

То есть, в целом, React умеет все, у него много точек расширения, отданных на откуп внешним разработчикам, но при этом не так много ресурсов «из коробки».

**Разработка и ресурсы Angular 2**

**DEVELOPMENT**

**IDES (Среды разработки)**

**Angular IDE** by Webclipse

**IntelliJ IDEA**

**Visual Studio Code**

**WebStorm** - Легкий, но мощный IDE, отлично оснащенный для комплексной разработки на стороне клиента и разработки на стороне сервера с помощью Node.js

**TOOLING (инструменты)**

**Angular CLI** - Официальный Angular CLI позволяет легко создавать и разрабатывать приложения от первоначальной фиксации до развертывания производства. Это уже следует нашим передовым практикам прямо из коробки!

**Angular Universal** - Server-side Rendering for Angular 2 apps.

**Augury** - A Google Chrome Dev Tools extension for debugging Angular 2 applications.

**Celerio Angular Quickstart** - Generate an Angular 2 CRUD application from an existing database schema

**Codelyzer** - A set of tslint rules for static code analysis of Angular 2 TypeScript projects.

**Lite-server** - Lightweight development only node server

**DATA LIBRARIES - Сторонние библиотеки.**

**Angular Fire** - Официальная библиотека Firebase и Angular 2

**Apollo** - это стек данных для современных приложений, созданных с помощью GraphQL. GraphQL - среда выполнения и язык запросов данных с открытым кодом. (используется Facebook для работы с данными пользователей)

**Meteor** - Используйте Angular 2 и Meteor для создания приложений JavaScript с полным стеком для мобильных и настольных компьютеров.

**ngrx** - Реактивные расширения для angular2.

RxJS powered state management for Angular applications, inspired by Redux

@ngrx/store is a controlled state container designed to help write performant, consistent applications on top of Angular. Core tenets:

- State is a single immutable data structure

- Actions describe state changes

- Pure functions called reducers take the previous state and the next action to compute the new state

- State accessed with the Store, an observable of state and an observer of actions

**UI COMPONENTS**

**ag-Grid**

**Angular Material 2** - Material Design components for Angular 2

**Clarity Design System** - UX guidelines, HTML/CSS framework, and Angular components

**Kendo UI** - One of the first major UI frameworks to support Angular 2

**ng-bootstrap** - The Angular 2 version of the Angular UI Bootstrap library. Used Bootstrap 4 CSS framework.

**Prime Faces** - PrimeNG is a collection of rich UI components for Angular 2

**Onsen UI** - UI components for hybrid mobile apps with bindings for both Angular 1 & 2. (интерфейс для гибридных мобильных )

**Semantic UI** - UI components for Angular using Semantic UI

**CROSS-PLATFORM DEVELOPMENT (Ресурсы для кросс-платформанной разработки)**

**Electron** - Electron Platform for Angular 2.

**Ionic** - Ionic offers a library of mobile-optimized HTML, CSS and JS components and tools for building highly interactive native and progressive web apps.

**NativeScript** - NativeScript is how you build cross-platform, native iOS and Android apps with Angular and TypeScript. Get 100% access to native APIs via JavaScript and reuse of packages from NPM, CocoaPods and Gradle. Open source and backed by Telerik.​​​

**ReactNative** - Angular 2 and React Native to build applications for Android and iOS

**Windows (UWP)** - An Universal Windows App (uwp) powered by Angular 2

**Google Chart Tools Directive For AngularJS**

**Angular 2 плюсы**

**Support for web components.**

Поддержка веб-компонентов. Вместо того, чтобы иметь закрытую систему для модуляции приложений AngularJS, основы Angular 2 лежат на новом стандарте веб-компонентов. Это означает, что AngularJS сможет напрямую, без каких-либо оберточных кодировок, использовать любой компонент, написанный как веб-компонент.

**Usage of Typescript.**

Использование Typescript. Самой большой точкой продажи TypeScript является инструментарий. Он предоставляет расширенные функции автозаполнения, навигации и рефакторинга. Наличие таких инструментов является почти обязательным требованием для крупных проектов.

**Great performance.**

Отличная производительность. Angular2 - не делает сравнения глубоких объектов. Если элементы добавлены / удалены в / из массива, обнаружение изменений не обнаружит его. То же самое для свойств объекта, если они не связаны непосредственно с представлением.

**Angular2 CLI.**

Angular2 CLI. Утилита Angular2 CLI позволяет легко создать приложение, которое уже работает, прямо из коробки. Это уже следует нашим рекомендациям!

Большой плюс — его нативная поддержка Dart. Dart — язык программирования, созданный Google. Dart позиционируется в качестве замены/альтернативы JavaScript.

Крупнейшая рекламная площадка в мире **Google AdWords** использует технологии (Angular 2 + Dart).

**Angular 2 минусы**

**Difficulty in learning.**

Трудность в обучении. Если вы раньше не пользовались машинописными текстами, вам нужно научиться этому и потратить время на это.

**Regular DOM**

Обычный DOM. Падает производительность если рендеринг шаблонов происходит на стороне клиента. Angular манипулирует фактическим DOM напрямую, что делает его более медленным и неэффективным по сравнению с React.

**Difficulties in the server templating**

Трудности с серверными шаблонами Угловой 2 по-прежнему сохраняет физическое разделение между JavaScript, управляющим вашим приложением, и отображаемым HTML.

**Основные конкуренты Angular 2**

**React**. Тут сталкиваются не только экосистемы, но и религии в каком-то смысле (JSX).

**Aurelia**. Это тоже достаточно молодой фреймворк, написан выходцем из Google, который разочаровался в Angular 2 и решил написать свою библиотеку «с преферансом и куртизантками». Выглядит достаточно перспективно, но порой создаётся впечатление, что он писался, исходя из принципа «назло маме отморожу уши».

**Polymer**. Тут Google конкурирует сам с собой. Впрочем, не совсем корректно их сравнивать, все же Polymer — это скорее огромная библиотека компонентов, а не цельный фреймворк.