Введение

№ урока:	1	**Курс:**	**Angular 2 Essential**

Средства обучения:	Редактор кода Node.js и npm

Обзор, цель и назначение урока

Это урок-введение, в котором рассматриваются особенности фреймворка Angular 2 его основные отличия от предыдущих версий. В этом уроке вы настроите среду для разработки, изучите структуру и напишите первое Hello world приложение используя Angular 2. Также в уроке будет рассмотрено использование TypeScript в контексте Angular2.

Содержание урока

1. Почему Angular2
2. Разница между Angular 1 и Angular 2
3. Что такое TypeScript. Основы использования TypeScript
4. Настройка инструментов разработки
5. Структура проекта Angular2
6. Hello world приложение на Angular2
7. Обзор общей архитектуры приложений

Резюме

ОБЗОР АРХИТЕКТУРЫ

Angular 2 – современный Фреймворк, разработанный компанией Google для создания клиентских приложений с использованием HTML и JavaScript (иди другого языка, который компилируется в JavaScript, например, Dart или TypeScript).

Сам Фреймворк состоит из нескольких библиотек, некоторые их этих библиотек являются главными, некоторые опциональными и необязательными.

Главные строительные блоки Angular приложения:

- Модули (Modules)
- Компоненты (Components)
- Шаблоны (Templates)
- Метаданные (Metadata)
- Привязка данных (Data Binding)
- Директивы (Directives)
- Сервисы (Services)
- Внедрение зависимостей (Dependency Injection)

Модули

Angular приложения модульные и при создании приложения мы должны использовать ***Angular Modules*** или ***NgModules***.

Каждое Angular приложение имеет как минимум один модуль, который называется ***root module*** по соглашению такой модуль именуется AppModule

CyberBionic Systematics ®
19 Yevgena Sverstuka Str., 5 floor
Kyiv, Ukraine

t. +380 (44) 361-8937
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

Большинство приложений содержат дополнительные модули – ***feature modules***. Такие модули являются коллекцией связанного кода, относящегося к определенному бизнес процессу, домену приложения или просто набору функциональности.
Модуль - это класс с декоратором @NgModule (не зависимо от типа модуля root или feature)

Самые важные свойства декоратора @NgModule:

- **declarations** – компоненты, директивы и pipes, которые принадлежат модулю
- **exports** – компоненты, которые будут видимы для других модулей
- **imports** – модули, которые необходимы этому модулю для работы
- **providers** – сервисы, которые данный модуль добавляет в глобальную коллекцию сервисов
- **bootstrap** – главное представление приложения, также называется AppComponent, только root module должен устанавливать свойство bootstrap

Модули Angular и JavaScript модули

Angular модули это классы декорированные @NgModule. JS модули механизм для манипуляции JavaScript объектами, который ничего общего не имеет с модулями Angular. В JavaScript модулем является отдельный файл. Те элементы модуля, которые должны быть видимыми помечены ключевым словом export.

Библиотеки Angular

Angular представляет собой коллекцию JavaScript модулей или библиотечных модулей. Каждый библиотечный модуль начинается с префикса @angular. Эти модули устанавливаются с помощью npm и импортируются с помощью ключевого слова import
Например, импортирование Component из @angular/core

```
import { Component } from “@angular/core”;
```

Компоненты

Компонент контролирует определенный фрагмент интерфейса приложения.

Логика компонента находиться в определении класса, декорированного @Component. Пользовательских интерфейс взаимодействует с классом через открытые свойства и методы.

Angular создает, обновляет и уничтожает компоненты в то время как пользователь взаимодействует с приложением.

Шаблоны

Пользовательский интерфейс приложения представлен шаблонами, которые связаны со своими компонентами.

Шаблон является обычной HTML разметкой с дополнительным Angular синтаксисом. Шаблон содержит директивы, выражения привязок, привязки обработчиков событий и т.д. Также шаблон может содержать другие пользовательские компоненты.

Метаданные

Метаданные определяют, как Angular должен обрабатывать класс.

При использовании TypeScript метаданные прикрепляются к классу с помощью декораторов.

@Component декоратор используется для настройки компонентов. Ниже приведен список нескольких параметров декоратора:

CyberBionic Systematics ®
19 Yevgena Sverstuka Str., 5 floor
Kyiv, Ukraine
t. +380 (44) 361-8937
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

- moduleId – устанавливает источник для базового адреса для создания путей относительно модуля, например, для свойства templateUrl
- selector – CSS селектор, который определяет какой HTML элемент необходимо инициализировать как компонент.
- tempalteUrl – путь к HTML шаблону
- providers – массив провайдеров для внедрения зависимостей.

Привязка данных

Привязка данных – механизм, позволяющий определить связь между шаблоном (пользовательским интерфейсом) и компонентом (его свойствами и методами). Для привязки данных в шаблон нужно добавить специальную разметку.

Есть 4 формы привязки данных

Тип привязки	**Пример**
Интерполяция	{{ value }}
Привязка свойств	[property]=”value”
Привязка событий	(event)=”handler”
Двунаправленная привязка	[(ng-model)]=”property”

Директивы

Директива – механизм для контроля изменения DOM дерева в процессе визуализации шаблонов. Директива — это класс со специальными метаданными.

Компоненты похожи на директивы. Компонент — это директива к которой привязан шаблон. Из-за того, что компоненты занимают особую роль в Angular приложении их выделяют как отдельную сущность. Но на самом деле компонент является директивой с дополнительными свойствами, которые относятся к настройкам шаблона.

В Angular есть два вида директив – структурные (structural) и директивы-атрибуты (attribute)

Structural – директивы изменяют структуру страницы добавляя, удаляя или меняя DOM элементы.

Примеры структурных директив *ngFor, *ngIf

Attribute – изменяют отображение или поведение уже существующих элементов на странице.

Пример директивы атрибута – ngModel

Сервисы

Сервис – класс, предоставляющий функции или значения необходимые приложению. Angular не определяет специальных механизмов для создания и регистрации сервисов. Основные потребители сервисов – компоненты.

Примеры сервисов:
Ведение лога событий
Получение данных
Расчет значений
Конфигурация приложения

CyberBionic Systematics ®
19 Yevgena Sverstuka Str., 5 floor
Kyiv, Ukraine

t. +380 (44) 361-8937
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

Внедрение зависимостей (Dependency Injection)

Внедрение зависимостей – способ создания и предоставления экземпляра класса, который является зависимостью для другого класса. Например, компонент, который отображает на странице каталог товаров требует сервис для получения доступа к серверу. Для того, чтобы создать компонент, необходимо, предварительно создать экземпляр сервиса, чтобы передать его компоненту. Процесс создания экземпляра зависимости и установка ссылки на него называется внедрением зависимости.

Для того, чтобы определить зависимости Angular проверяет типы аргументов конструктора создаваемого компонента (или другого элемента). Перед тем как создавать компонент Angular взаимодействует с инжектором (injector). Инжектор сопровождает контейнер с экземплярами сервисов, которые ранее были созданы. Если запрашиваемый сервис еще не находиться в контейнере, инжектор создает его. Когда все сервисы для создания компонента доступны Angular вызывает конструктор, передавая в качестве параметров настроенные сервисы.

Для того чтобы Injector мог создавать сервисы, необходимо указать провайдеров этих сервисов. Для этого используется свойство providers в модуле или компоненте. Провайдером обычно выступает сам класс сервиса.

НАСТРОЙКА VISUAL STUDIO CODE

Для того чтобы спрятать файлы с расширением js и map.js, который создаются TypeScript компилятором, необходимо выполнить следующие действия.

1. Открыть настройки для пользователя или рабочей области File -> Preferences -> User Setting | Workspace Setting
2. В редакторе кода, в окне справа ввести следующий код:

```
"files.exclude": {
        "**/.git": true,
        "**/.DS_Store": true,
        "**/*.js.map": true,
        "**/*.js": {
            "when": "$(basename).ts"
        }
    }
```

3. Сохранить файл.

Для того, чтобы открыть консоль для текущего проекта, в Explorer окне, вызовите контекстное меню на директории или файле и выберите пункт “Open in Command Prompt”, либо, нажмите (по умолчанию) комбинацию клавиш **CTRL + `.** При этом командная строка будет запущена в самом редакторе кода, как одна из его панелей.

Закрепление материала

- Что такое TypeScript?
- Как выполнить компиляцию *.ts файла?
- В чем разница между JavaScript модулем и Angular модулем?
- Что делает компонент в Angular приложении?
- Что такое директива?
- Какие типы привязок данных существуют в Angular 2?
- Назовите типы директив в Angular 2 и опишите их особенности.

Самостоятельная деятельность учащегося

Задание 1
Создайте простое приложение используя Angular 2.

CyberBionic Systematics ®
19 Yevgena Sverstuka Str., 5 floor
Kyiv, Ukraine

t. +380 (44) 361-8937
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

Определите в AppComponent следующий массив.
const VALUES = [
"Hello World", "Привет Мир","Привіт Світ", "Hola Mundo", "Bonjour le monde", "Hallo Welt", "Ciao mondo", "Witaj świecie", "Hej världen", "Pozdravljen, svet", "Прывітанне Сусвет"];

Выведите массив в виде списка в шаблоне AppComponent.

Рекомендуемые ресурсы

Официальный сайт
https://angular.io/

Quick Start
https://angular.io/docs/ts/latest/quickstart.html

Особенности Angular 2
https://angular.io/features.html

IDES:
IntelliJ IDEA https://www.jetbrains.com/idea/
Visual Studio Code https://code.visualstudio.com/b?utm_expid=101350005-28.R1T8FshdTBWEfZjY0s7XKQ.1&utm_referrer=https%3A%2F%2Fangular.io%2F
Webstorm https://www.jetbrains.com/webstorm/

Базовые типы TypeScript
https://www.typescriptlang.org/docs/handbook/basic-types.html

Классы
https://www.typescriptlang.org/docs/handbook/classes.html

Интерфейсы
https://www.typescriptlang.org/docs/handbook/interfaces.html

Декораторы
https://www.typescriptlang.org/docs/handbook/decorators.html

Шпаргалка Angular 2
https://angular.io/docs/ts/latest/guide/cheatsheet.html

Рекомендуемое видео

Запись Вебинара ECMASCcritpt 6 новые возможности
https://www.youtube.com/watch?v=Wdg0p4QGV6s

Видео курс TypeScript Fundamentals
http://itvdn.com/ru/video/typescriptfundamentals

Как перестать бояться Angular 2 и начать использовать компоненты уже сегодня
https://www.youtube.com/watch?v=sSnPmLxW89s

CyberBionic systematics

CyberBionic Systematics ®
19 Yevgena Sverstuka Str., 5 floor
Kyiv, Ukraine

t. +380 (44) 361-8937
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com