УДК 004.021

ПРОЕКТИРОВАНИЕ ГРАФИЧЕСКОГО ИНТЕРФЕЙСА ДЛЯ ВЕБ-ПРИЛОЖЕНИЯ, РАЗРАБОТАННОГО ДЛЯ ОТСЛЕЖИВАНИЯ СТАТУСА ВЫПОЛНЯЕМЫХ ИТЕРАЦИЙ И СЛЕЖЕНИЕ ВРЕМЕНИ ИХ ВЫПОЛНЕНИЯ

К.С. САМСОНОВ
(Представлено: ст. преп. Ю.Н. КРАВЧЕНКО)

Рассматривается построение графического интерфейса пользователя для веб-приложения.

Введение. Одним из самых известных языков программирования в разработке веб-приложений является JavaScript. Он далеко не прост в изучении, но предоставляет гарантию создания детально продуманного и многофункционального решения. Это язык объектно-ориентированный программирования, который дает возможность реализовать сложную механику веб-страниц. Пользователь может наблюдать разного рода 2D и 3D анимацию, прокрутку видео и обновление контента, одним словом – придает динамичность и не только. Все это реализуется с помощью JavaScript.

Главные плюсы JavaScript, за которые он и получил такую распространенность:

- Полная интеграция с HTML/CSS.
- По умолчанию поддерживается всеми распространенными браузерами и включен в них.
- Простая реализация простых задач.

Основной раздел. Angular 5 предоставляет такую функциональность, как двустороннее связывание, позволяющее динамически изменять данные в одном месте интерфейса при изменении данных модели в другом, шаблоны, маршрутизация. Одной из ключевых особенностей Angular является то, что он использует в качестве языка программирования JavaScript. Основная цель данного фреймворка предназначена для разработки одностраничных приложений. Его цель — расширение браузерных приложений на основе MVC-шаблона, а также упрощение тестирования и разработки.

Фреймворк работает с HTML, содержащим дополнительные пользовательские атрибуты, которые описываются директивами, и связывает ввод или вывод области страницы с моделью, представляющей собой обычные переменные JavaScript. Значения этих переменных задаются вручную или извлекаются из статических или динамических JSON-данных.

Веб-приложение состоит из множества файлов: текстового содержимого, кода, таблиц стилей, медиа-контента, и так далее. Когда вы создаете веб-сайт, вы должны собрать эти файлы в разумную структуру и убедиться, что они могут взаимодействовать друг с другом.

Язык гипертекстовой разметки (Hypertext Markup Language, HTML) - это код, который вы используете для структурирования веб-содержимого и придания ему смысла и цели. Каскадные таблицы стилей (Cascading Stylesheets, CSS) - это код, который вы используете для стилизации своего веб-сайта. avaScript – это язык программирования, который вы используете для добавления интерактивных функций для вашего веб-сайта, например, игр, событий, которые происходят при нажатии кнопок или ввода данных в формы, динамических эффектов стилизации, анимации и многого другого.

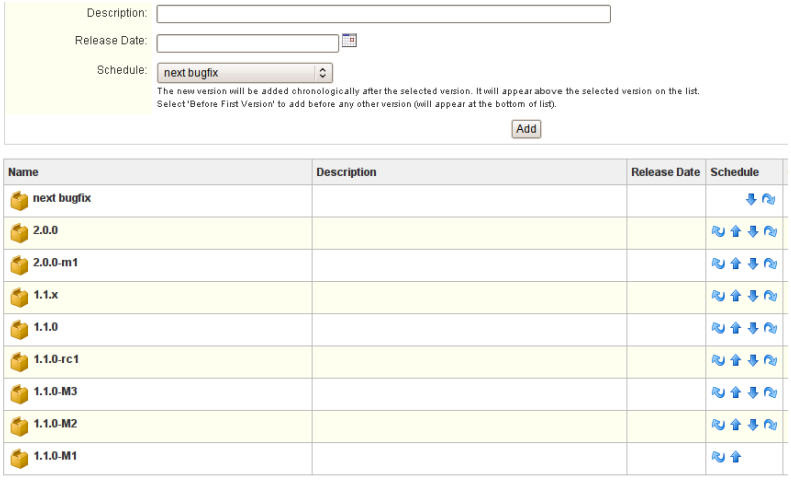

Рисунок. – Макет главной страницы приложения

Компоненты представляют основные строительные блоки приложения Angular 2. Каждое приложение Angular имеет как минимум один компонент. Поэтому создадим в папке src/app новый файл, который назовем app.component.ts и в котором определим следующий код компонента

Далее собственно идет функция-декоратор @Component, которая ассоциирует метаданные с классом компонента AppComponent. В этой функции, во-первых, определяется параметр selector или селектор css для HTML-элемента, который будет представлять компонент. Во-вторых, здесь определяется параметр template или шаблон, который указывает, как надо визуализировать компонент. В этом шаблоне задана двусторонняя привязка с помощью выражений [(ngModel)]="name" и {{name}} к некоторой модели name.

И в конце собственно экспортируется класс компонента AppComponent, в котором как раз определяется модель name - в данном случае это пустая строка.

Приложение Angular состоит из модулей. Модульная структура позволяет легко подгружать и задействовать только те модули, которые непосредственно необходимы. И каждое приложение имеет как минимум один корневой модуль. Поэтому создадим в папке src/аррновый файл, который назовем app.module.ts со следующим содержимым:

Этот модуль, который в данном случае называется AppModule, будет входной точкой в приложение.

С помощью директив import здесь импортируется ряд нужных нам модулей. Прежде всего, это модуль NgModule. Для работы с браузером также требуется модуль BrowserModule. Так как наш компонент использует элемент input или элемент формы, то также подключаем модуль FormsModule. И далее импортируется созданный ранее компонент.

Теперь нам надо указать Angular, как запускать наше приложение. Для этого создадим в папке src (на уровень выше, чем расположены файлы app.component.ts и app.module.ts) файл main.ts со следующим содержимым

Этот код инициализирует платформу, которая запускает приложение, и затем использует эту платформу для загрузки модуля AppModule.

Данный файл определяет полифилы - инструменты, которые необходимы для поддержки приложения на Angular старыми браузерами. В частности, первая большая группа полифилов предназвачена для работы в IE9-11. Однако если мы не планируем поддерживать этот браузер, то данную группу импорта пакетов можно не включать. Но в любом случае файл должен содержать импорт zone: import 'zone.js/dist/zone'.

Заключение. В данной статье рассмотрены основные элементы построения графического интерфейса для веб-приложения. Представлены примеры и результаты.