



Politechnika Krakowska im. Tadeusza
Kościuszki
Wydział Fizyki, Matematyki
i Informatyki



Technologie Aplikacji Internetowych

Projekt: *Anatomia aplikacji AngularJS. Wywołanie AngularJS, komunikacja z serwerem, szablony i dołączanie danych, formatowanie danych za pomocą filtrów, dyrektywy do zmiany elementów drzewa DOM*

Data oddania:

26-05-2018

Kierunek: *Informatyka Stosowana*
Rok akademicki: 2017/2018

Wykonał:

Przemysław Kleszcz
Nr albumu: 124624

Spis treści

1. Wprowadzenie	3
2. Cel projektu.....	3
3. Środowisko implementacyjne	3
4. Wykorzystane technologie	4
5. Wprowadzenie do AngularJS.....	5
6. Filary aplikacji AngularJS.....	6
7. Projekt aplikacji	7
8. Implementacja aplikacji	9
9. Podsumowanie	11
10. Bibliografia	11

1. Wprowadzenie

W zakresie projektu zaliczeniowego zrealizowano sprawozdanie na temat technologii AngularJS. AngularJS jest otwartym frameworkiem opartym o język Javascript. Został stworzony przez firmę Google. Jego głównym celem jest wspomaganie tworzenia i rozwoju aplikacji internetowych na pojedynczej stronie (SPA). Opisywana biblioteka aktywnie wykorzystuje wzorzec MVC (Model-View-Controller) aby ułatwić rozwój i testowanie aplikacji. AngularJS został stworzony z przekonaniem, że programowanie deklaratywne powinno być używane do budowy interfejsów i łączenia komponentów oprogramowania, podczas gdy programowanie imperatywne znajduje zastosowanie w logice biznesowej. Poniższe opracowanie ma za zadanie przybliżyć tematykę AngularJS oraz omówić komponenty składowe biblioteki. Podjęta zostanie próba implementacji aplikacji SPA, a także testów weryfikujących działanie pod względem funkcjonalnym. Na koniec zostaną wyciągnięte wnioski oraz postawione zostanie pytanie o przyszłość technologii Angular.

2. Cel projektu

Celem projektu jest pełna analiza infrastruktury i elementów składowych systemu AngularJS. Możliwości platformy zostaną przetestowane, zilustrowane na przykładowej aplikacji a następnie poddane ocenie i porównaniu. Na początku opisane zostanie środowisko w jakim implementowane i uruchamiane będą przykłady i aplikacja stworzone na potrzeby opracowania. Następnie wymienione zostaną wszystkie wykorzystane do implementacji technologie. Przybliżone zostaną podstawowe elementy ramy projektowej takie jak: moduły, kontrolery, filtry czy dyrektywy. Nakreślony zostanie projekt przykładowej aplikacji. Na koniec odbędzie się właściwa faza implementacji.

3. Środowisko implementacyjne

Implementacja została w pełni opracowana w środowisku *Microsoft Visual Studio*. Jest to zintegrowane środowisko produkcyjne (IDE) zaprojektowane przez firmę *Microsoft*. Jest używane do produkcji programów komputerowych na platformę Windows. Za jego pomocą można produkować zarówno kod natywny jak i zarządzany przez platformę .NET. Oprócz pełnego wsparcia dla platformy .NET, Visual Studio oferuje wiele narzędzi i dodatków usprawniających proces implementacji dla wielu innych technologii – również dla AngularJS. Edytor narzędzia jest w pełni zintegrowany z technologią – zapewnia odpowiednie podświetlenie i weryfikację składni użytych filtrów oraz dyrektyw. Instrukcję w języku Javascript są podpowiadane za pomocą narzędzia Intellisense. Wbudowany menedżer pakietów wspomaga instalację i utrzymywanie dodatkowych bibliotek i zależności AngularJS. Wbudowane mechanizmy przyspieszają poruszanie się/tworzenie/zmianę plików .js, .css oraz .html, na których opiera się Framework AngularJS.

4. Wykorzystane technologie

- *Javascript* – Jest interpretowanym, wieloparadygmatowym językiem wysokiego poziomu. Posiada dynamiczne, słabe typowanie. Oprócz HTML oraz CSS jest jedną z głównych technologii www dnia dzisiejszego. Jest wykorzystywany do tworzenia dynamicznych, interaktywnych stron internetowych. Wszystkie nowoczesne przeglądarki internetowe posiadają wsparcie języka bez konieczności instalowania dodatków czy wtyczek. Wszystko za sprawą wbudowanego silnika Javascript. AngularJS wykorzystuje język Javascript do oprogramowania logiki biznesowej.

- *HTML* – Jest standardowym językiem znaczników przeznaczonym do tworzenia stron/aplikacji internetowych. Przeglądarki internetowe otrzymują dokument HTML od serwera www i przekształcają zawarte w nim znaczniki w funkcjonalne, multimedialne strony internetowe. HTML jest językiem opisu struktury strony internetowej. Elementy AngularJS takie jak dyrektywy czy filtry, integrują się ze znacznikami HTML i odpowiednio rozszerzają ich funkcjonalność.

- *CSS*- Jest językiem stylów wykorzystywanym do opisywania prezentacji dokumentu zapisanego za pomocą znaczników. Chociaż wykorzystuje się go głównie do wyznaczenia wizualnego stylu interfejsów użytkownika wykonanych w HTML, może być również stosowany wraz z dokumentami XML, SVG, XUL. Powstał głównie z potrzeby separacji zawartości od warstwy prezentacji (wygląd, czcionki, kolory...). Taka separacja usprawnia dostęp do zawartości, zwiększa elastyczność oraz kontrolę nad ustawieniami prezentacji. Dużą zaletą css jest możliwość wykorzystania istniejącej bazy opisu stylów do większej liczby dokumentów. Co redukuje odpowiednio powtarzalną liczbę instrukcji.

- *jQuery* - Międzyplatformowa biblioteka oparta o język Javascript. Stworzona w celu uproszczenia tworzenia skryptów po stronie klienta. Składnia jQuery jest zaprojektowana w taki sposób by możliwie jak najbardziej ułatwić nawigowanie po strukturze dokumentu *DOM*, tworzenie animacji, obsługę zdarzeń, manipulację arkuszami stylów oraz przetwarzaniem żądań typu AJAX. W znaczący sposób skraca liczbę instrukcji pozwalających na manipulację różnymi zasobami. Dzięki ujednoliconej składni, nie wymaga odrębnego zestawu kodu na potrzeby obsługi różnych przeglądarek internetowych.

- *Jasmine* - Jest Frameworkiem nastawionym na testowanie zachowań. W przeciwieństwie do innych bibliotek testów, nie polega na przeglądarek, strukturze dokumentu *DOM* czy jakimkolwiek frameworku js. Dzięki temu nadaje się doskonale do testowania stron internetowych, projektów Node.js lub dowolnej aplikacji, która działa w miejscu, w którym można uruchomić Javascript. Dzięki mechanizmom integracji biblioteki z AngularJS, konfiguracja zestawów testowych oraz sama faza testowania przebiega bardzo szybko.

-*Kendo-UI* – Zestaw kontrolek, widgetów i narzędzi usprawniających projektowanie i implementację webowego interfejsu użytkownika. Komponenty *Kendo-UI*. Pomagają w szybki sposób zaimplementować dowolny widok, redukując ilość instrukcji *Javascript*, *HTML* czy *CSS*. Można je zintegrować ze strukturami AngularJS a także rozszerzać ich funkcjonalność za pomocą mechanizmów biblioteki.

5. Wprowadzenie do AngularJS

AngularJS jest szkieletowym systemem Javascript umożliwiającym implementowanie aplikacji typu SPA z pomocą architektury MVC. System ten umożliwia używanie standardowych i wypróbowanych technik programistycznych, które tradycyjnie stosowano po stronie serwera. System zapewnia spójną, skalowalną strukturę, która znacznie upraszcza tworzenie dużych i skomplikowanych aplikacji.

Podstawowym pojęciem, na którym opiera się cały system AngularJS, jest wzorzec projektowy o nazwie MVC (*Model-View-Controller*). Wzorzec ten, opisuje sposób rozdzielenia logicznych składników dużych aplikacji. Pomaga programiście zdecydować, jak i gdzie rozdzielić różne obowiązki. Zgodnie z założeniami MVC aplikację należy podzielić na trzy moduły:

- Model – Warstwa, która pośredniczy w komunikacji z widokiem oraz kontrolerem. Stanowią go zazwyczaj dane, które najczęściej są pobierane z serwera.
- Widok – Interfejs, który jest widoczny dla użytkownika, i z którym użytkownik wchodzi w interakcję. Interfejs ten jest dynamiczny, tzn aplikacja generuje go na podstawie aktualnego modelu.
- Kontroler – Stanowi warstwy logiki biznesowej i prezentacji. Do jego zadań należy pobieranie danych i podejmowanie decyzji dotyczących sposobu prezentowania modelu.

Dzięki zastosowaniu reguły separacji obowiązków, wzorzec MVC przynosi szereg korzyści:

- Każda część odpowiada tylko za jedną rzecz. Model to dane, widok to interfejs użytkownika, a kontroler to logika biznesowa. Dzięki takiemu rozdziałowi obowiązków wybór odpowiedniego miejsca dla pisanego kodu oraz znalezienie wcześniej napisanych fragmentów kodu nie sprawiają trudności.
- Każda część jest maksymalnie niezależna od pozostałych. Dzięki temu elementy aplikacji są nadajacymi się do wielokrotnego użytku i łatwymi w obsłudze serwisowej modułami.

Aplikacja AngularJS to metaszkielec do tworzenia aplikacji jednostronicowych. Tworzenie i obsługa aplikacji przy użyciu systemu szablonowego działającego po stronie klienta i Javascriptu mogą być żmudne i uciążliwe. AngularJS pozwala pozbyć się niepotrzebnych elementów i zapewnia podstawową strukturę, dzięki czemu programista może skoncentrować się na istocie swojej aplikacji. Program składa się z mniejszej ilości kodu niż analogiczny program napisany przy użyciu czystego Javascriptu i biblioteki jQuery. Także w porównaniu z innymi frameworkami AngularJS wymusza napisanie mniejszej ilości szablonowego kodu oraz sprawia, że kod jest bardziej przejrzysty dzięki przesunięciu logiki do niewidocznych komponentów wielokrotnego użytku.

6. Filary aplikacji AngularJS

Istnieje jeszcze jeden ważny podział aplikacji AngularJS. Podział na moduły, kontrolery, filtry i dyrektywy. Są to tzw filary aplikacji AngularJS. Elementy te są odpowiedzialne za podzielenie aplikacji na logiczne warstwy abstrakcji a także za sterowanie jej przepływem.

Moduły w systemie AngularJS służą do zapisywania fragmentów kodu realizującego określone zadanie pod pewną nazwą. Można je porównać do pojęcia pakietu, wykorzystywanego przez niektóre technologie kompilowane. W module można zdefiniować odrębne kontrolery, usługi i inne struktury AngularJS. Są to funkcje i kod o dostępności w obrębie całego modułu. Ponadto moduł może być zależny od innych modułów, które są wówczas jego zależnościami definiowanymi podczas tworzenia egzemplarza tego modułu. Oznacza to, że AngularJS znajdzie moduł o określonej nazwie i dopilnuje, by wszystkie zdefiniowane w nim funkcje, kontrolery, usługi itd. były dostępne dla kodu zdefiniowanego w tym module. Moduły to nie tylko kontenery na kod JavaScript, ale również punkty uruchomieniowe aplikacji. Można wybrać, który moduł ma być głównym punktem wejściowym do aplikacji, przekazując jego nazwę do dyrektywy ng-app.

Kontroler aplikacji AngularJS jest zwykłą funkcją. Jego instancja zostaje utworzona w momencie napotkania przez interpreter dyrektywy ng-controller. Dyrektywa ng-controller szuka kontrolera o podanej przez nas nazwie, jeżeli go nie znajdzie AngularJS rzuci w konsoli wyjątek.

Kontrolery pełnią w AngularJS rolę podstawowej jednostki wykonawczej. Zawierają funkcje JavaScript wykonujące lub wyzwalające większość działań związanych z interfejsem użytkownika. Niektóre zadania kontrolera w systemie AngularJS:

- Pobieranie danych z serwera do interfejsu użytkownika.
- Wybieranie partii danych do pokazania użytkownikowi.
- Implementacja logiki prezentacyjnej odpowiedzialnej np. za wyświetlanie elementów, części interfejsu użytkownika, styl tych składników itd.
- Interakcja z użytkownikiem, np. obsługa kliknięć i sprawdzanie poprawności otrzymanych danych tekstowych.

Filtry AngularJS służą do przetwarzania danych i formatowania wartości w celu przedstawienia ich użytkownikowi. Stosuje się je do wyrażeń w kodzie HTML lub bezpośrednio do danych w kontrolerach i usługach. W większości przypadków filtr wykorzystuje się jako ostatni poziom formatowania danych z magazynu, aby przystosować je do pokazania użytkownikowi.

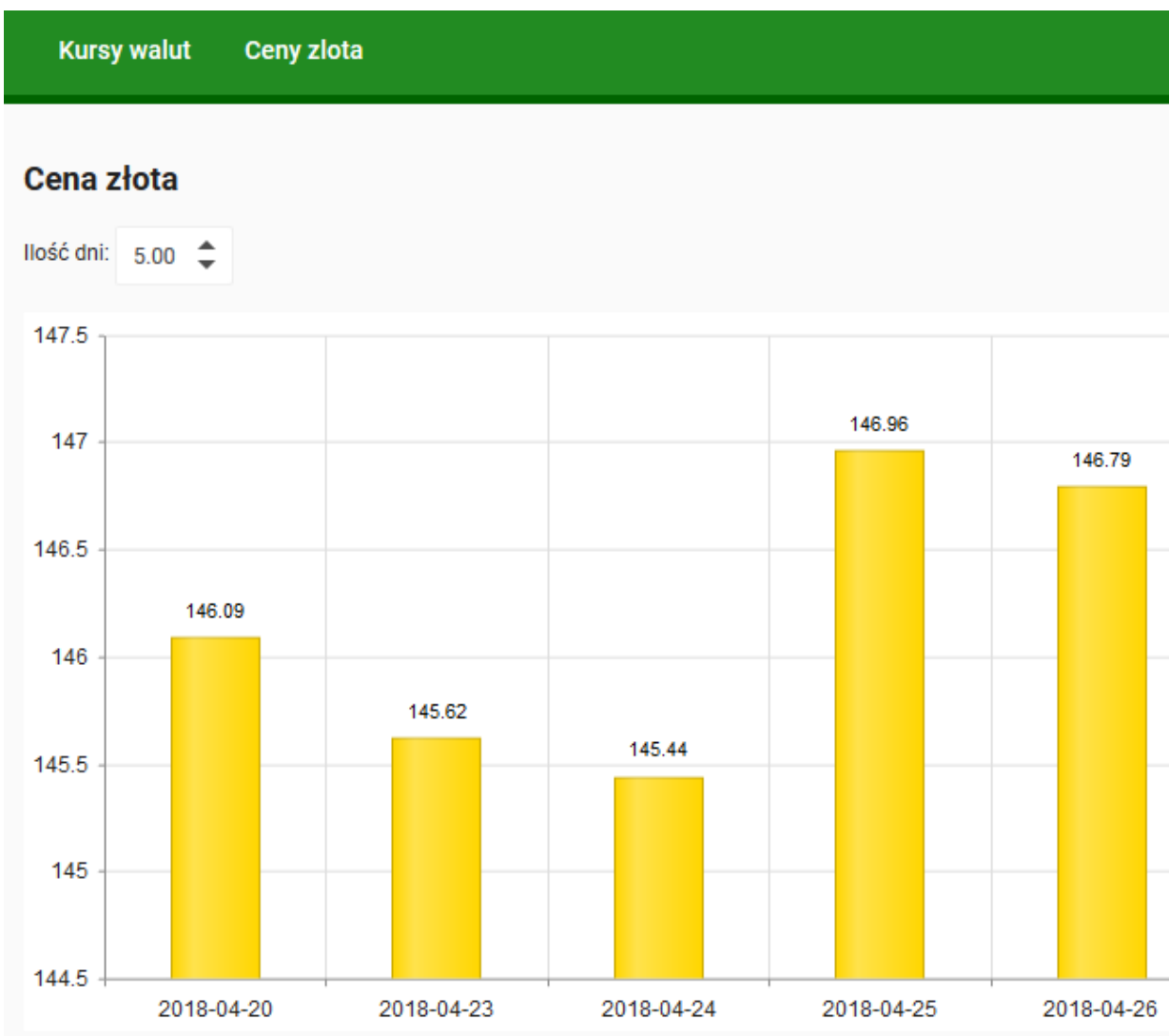
Dyrektywy to konstrukcje, które pozwalają na wzbogacenie funkcjonalności, zmodyfikowanie sposobu działania interfejsu użytkownika lub stworzenie całkowicie nowych komponentów wykorzystujących HTML i JavaScript. Dyrektywy mają własną logikę renderowania i logikę biznesową (skąd pobierać dane i jak reagować na działania użytkownika).

7. Projekt aplikacji

W zakresie projektu zaliczeniowego zrealizowana została aplikacja typu Web w technologii AngularJS. Wykonany program posiada graficzny interfejs użytkownika, przy pomocy, którego użytkownik wprowadza i przetwarza dane oraz jest w stanie obserwować wyniki zaistniałych w czasie działania operacji.

Kursy walut Ceny złota		
Kursy walut		
Tabela: A ▼		
currency ↑	code	mid
bat (Tajlandia)	THB	0.1102
dolar amerykański	USD	3.4827
dolar australijski	AUD	2.6338
dolar Hongkongu	HKD	0.4438
dolar kanadyjski	CAD	2.7097

Zadaniem aplikacji jest prezentacja kursów walut z bieżącego dnia oraz ewidencja cen złota za dni poprzednie (wyrażone w PLN). System pobiera dane z serwisu Narodowego Banku Polskiego poprzez wystawiony interfejs REST. Format wymiany danych to JSON. Interfejs zapewnia możliwość ustalenia badanej tabeli kursów oraz ilości dni biorących udział w analizie cen złota. Wyniki wyszukiwania są prezentowane w przyjaznej formie – tabeli oraz wykresu słupkowego.



8. Implementacja aplikacji

Na aplikację składa się jeden moduł organizujący w logiczny sposób komponenty podzielone wg domeny. Wewnątrz modułu pracują trzy kontrolery. Dwa spośród nich odpowiadają za logikę biznesową odpowiednich podstron. Trzeci odpowiada za implementację wspólnej logiki dla wszystkich widoków i zagadnień związanych z routingiem.

Sercem aplikacji jest podstrona *Index.html*. Zawiera ona dyrektywę *ng-view*, która współpracuje ściśle z implementacją routingu po stronie klienta. Kliknięcie odpowiedniej zakładki w górnym panelu nawigacyjnym powoduje uruchomienie mechanizmu routingu i załadowanie odpowiedniego widoku w miejsce wystąpienia dyrektywy *ng-view*.

```
<md-toolbar id="mainToolbar" ng-controller="indexController as indCtrl">
<ul kendo-menu>
<li><a href="#/exchangerates">Kursy walut</a></li>
<li><a href="#/goldvalue">Ceny złota</a></li>
</ul>
</md-toolbar>

<div class="body-content">
<div ng-view="">
</div>
</div>
```

W zakładce kursów walut, po wybraniu odpowiedniej pozycji z rozwijanej listy „Tabela”, model właściwości AngularJS jest automatycznie ustawiany a następnie wywoływane jest zdarzenie obsługujące komunikację z serwerem NBP. Po otrzymaniu zestawu danych z API, tworzony jest obiekt frameworka KendoUI a dane są dynamicznie wstawiane do tabeli.

```
<kendo-grid options="exCtrl.mainGridOptions" k-data-source="exCtrl.data">
</kendo-grid>

self.inputValue = "";
self.getExchangeRates = function () {
var baseUrl = "http://api.nbp.pl/api/exchangerates/tables/";
var url = baseUrl + self.inputValue + "?format=json";
$http.get(url).success(function (data) {
self.rates = data[0].rates;
self.data = new kendo.data.DataSource({
data: self.rates
});
}).error(function () {
alert("Api error");
});
}
```

W zakładce cen złota po wybraniu odpowiedniej pozycji z rozwijanej listy „Ilość dni”, model właściwości AngularJS jest automatycznie ustawiany a następnie wywoływane jest zdarzenie obsługujące komunikację z serwerem NBP. Po otrzymaniu zestawu danych z API, tworzony jest obiekt frameworka KendoUI a dane są automatycznie propagowane do kontrolki wykresu

słupkowego. Kontrolka wykresu jest automatycznie inicjalizowana danymi, które przybyły ze zdalnego serwera a silnik KendoUI tworzy graficzną reprezentację wykresu.

```
<divkendo-chart
    k-title="{ align: 'left' }"
k-series-defaults="{
type: 'column',
labels: {
visible: true,
background: 'transparent'
}
}"
k-series="[{
field: 'cena',
categoryField: 'data',
colorField: 'userColor'
}]"
k-data-source="genCtrl.data"
style="height: 400px;">
</div>

self.inputValue = 0;
self.getGoldValues = function () {
var baseUrl = "http://api.nbp.pl/api/cenyzlota/last/";
var url = baseUrl + self.inputValue + "?format=json";
$http.get(url).success(function (data) {
for (var i = 0; i < data.length; i++)
data[i].userColor = "#ffd600";

self.rates = data;
self.data = newkendo.data.DataSource({
data: self.rates
});
}).error(function () {
alert("Api error");
});
}
```

9. Podsumowanie

Główne założenia sprawozdania zostały zrealizowane. Udało się w możliwie wyczerpujący sposób ukazać podstawowe działanie poszczególnych komponentów systemu AngularJS. Wyszczególnione przykłady potwierdzają założenia z jakimi budowane są systemy oparte o tę technologię.

10. Bibliografia

1. Shyam Seshadri, Brad Green, AngularJS: Up and Running. Enhanced Productivity with Structured Web Apps, O'Reilly, 2014
2. Adam Freeman, AngularJS. Profesjonalne techniki, Apres, 2015