



# Regulamin szkolenia **Full Stack Developer**

Edycja 2

**Autor: Kamil Wołczyk**

Maj 2017

# Harmonogram

---

Sobota	Niedziela	Miesiąc	
07	08	październik	ZJAZD
14	15	październik	-
21	22	październik	ZJAZD
28	29	październik	-
04	05	listopad	ZJAZD
11	12	listopad	-
18	19	listopad	ZJAZD
25	26	listopad	-
02	03	grudzień	ZJAZD
09	10	grudzień	-
16	17	grudzień	ZJAZD
23	24	grudzień	Wigilia Bożego Narodzenia
30	31	grudzień	Sylwester
06	07	styczeń	Święto Trzech Króli
13	14	styczeń	ZJAZD
20	21	styczeń	-
27	28	styczeń	ZJAZD

*\*Początek kursu 07 października 2017r, zakończenie 28 stycznia 2018r*

Sesje zajęć w każdy dzień:

09:00 – 10:45

11:00 – 12:30

13:00 – 14:30

14:45 – 16:00

Podsumowując, 4 miesiące zajęć, po 2 zjazdy w każdym miesiącu, co daje łącznie 16 dni szkoleniowych.

W każdy dzień 6h warsztatów. **ŁĄCZNIE 96 godzin zajęć**

# Zalety

---

1. Całkowity kurs Full Stack Developmentu, czyli możliwość zdobycia wiedzy z pełnego zakresu aktualnie popularnych technologii wykorzystywanych w projektach IT. Dzięki temu zdobędziesz wiedzę potrzebną do zbudowania kompletnej aplikacji od A do Z – części serwerowej, bazy danych i warstwy klienta.
2. Kurs nie ma na celu nauczyć Cię bardzo dokładnie jednej technologii – to możesz zrobić samemu czytając książki i wykonując ćwiczenia. Szkolenie ma za zadanie pokazać wspólne wykorzystanie wszystkich komponentów i dokładnie zrozumieć ich sposób działania.
3. 100% czasu przeznaczony jest na warsztaty, na których najpierw dowiesz się jak dana biblioteka funkcjonuje, a równolegle będziesz implementował życiowy przykład. Finalnie uzyskasz pełną aplikację webową.
4. Zajęcia poprzez odpowiedni dobór tematów przygotowują Cię do uczestnictwa w procesie wytwarzania oprogramowania w zespole projektowym. Nie tylko poznasz języki programowania (C#, JavaScript, Typescript) i nowoczesne frameworki (.NET MVC, AngularJS), ale również dowiesz się o sposobie pracy w metodyce scrum, sposobach synchronizacji kodu w zespole dzięki narzędziu GIT, dobrych praktykach programowania, a także strategiach dostarczania gotowego produktu.
5. Nauka w małej grupie szkoleniowej. Dajemy gwarancję, że w grupie będzie od 5 do maksymalnie 10 osób. Dzięki temu nawiążesz bezpośredni kontakt z prowadzącym, który będzie nadzorował cały proces Twojej nauki.
6. Otrzymasz zadania do wykonania w domu pomiędzy zajęciami, których samodzielne zrealizowanie gwarantuje uporządkowanie wiedzy i odzwierciedli Twój postęp w zdobywaniu wiedzy. Prowadzący wykona code review Twojego zadania przed kolejnymi zajęciami.
7. Przy omawianiu technologii prowadzący wskaże źródła wiedzy (książki, blogi, tutoriale, kursy video), które pozwolą Ci poszerzyć wiedzę we własnym zakresie.
8. Zajęcia prowadzone są w weekendy (sobota i niedziela). Każdy dzień został podzielony na 4 bloki, pomiędzy którymi będą krótkie przerwy na posiłek i odpoczynek.
9. Przystępna cena. Jedynie 3000zł brutto za cały kurs.
10. Zaangażowanie popłaca! Firma HeadChannel Ltd umożliwia zatrudnienie najlepszym uczestnikom kursu.

# Płatność

---

Koszt pełnego kursu dla pojedynczego uczestnika to 3000zł brutto. Uwzględniając łączną liczbę godzin szkoleniowych (96h) koszt jednej godziny to około 31zł brutto.

Sposób płatności:

- 1 rata – płatność 3000zł przy podpisaniu umowy bezpośrednio na konto firmy Headchannel.
- 4 raty – płatność 4x750zł, płatne w trakcie trwania kursu, do 10 dnia każdego miesiąca szkoleniowego (odpowiednio: 10 października, 10 listopada, 10 grudnia 2017 oraz 10 stycznia 2018r).

Dla osób, które zapiszą się na kurs do dnia 1 września 2017r. Oferowana jest **cena promocyjna 2500zł** (4 raty po 625zł)

## Zapisy

---

Zgłoszenia na szkolenie przyjmowane są do 1 października 2017 roku. Maksymalna liczba uczestników drugiego cyklu warsztatów to 10 osób.

Zgłoszenia można wykonywać poprzez formularz kontaktowy umieszczony na stronie internetowej, drogą mailową oraz telefonicznie.

W przypadku większego zainteresowania, w terminie 02.10.2017 – 05.10.2017 zostaną przeprowadzone rozmowy z chętnymi osobami.

W sytuacji braku zainteresowania (poniżej 5 uczestników), organizator ma prawo odstąpić od realizacji szkolenia.

# Tematyka

---

## Dzień 1

1. Wprowadzenie do tworzenia oprogramowania – omówienie procesu wytwarzania oprogramowania, ról i odpowiedzialności członków zespołu. Omówienie zasad i sposobu pracy programisty.
2. Prezentacja nowoczesnej architektury systemów. Przedstawienie popularnych języków programowania, bibliotek, technologii i narzędzi wykorzystywanych w pracy programisty.
3. Przygotowanie środowiska pracy – instalacja narzędzi (wymagany system Windows). Zapoznanie się z programem Microsoft Visual Studio 2017 Community.
4. Wyjaśnienie działania kompilatora dla języka C#. Przedstawienie i opracowanie struktury solucji i projektu.

## Dzień 2

1. Poznanie bazowych aspektów języka C#: typy danych, zmienne i tablice. Nauka 'debuggowania' programu.
2. Tworzenie aplikacji konsolowej w systemie Windows. W tym:
  - Operatory arytmetyczne i logiczne
  - Warunki (if, switch)
  - Pętle (for, foreach, while)
  - Enkapsulacja (public, protected, private)
  - Metody
  - Enumeracje
  - Przestrzenie nazw (namespace)
3. Git – omówienie systemu kontroli wersji. Ćwiczenie podstawowego użycia system git: komendy commit, pull, push.
4. Zaawansowane aspekty pracy z Git. Tworzenie branchy, rozwiązywanie konfliktów, gitflow.

## Dzień 3

1. Przedstawienie metodologii Agile Scrum – ogólne aspekty i zasady wytwarzania oprogramowania. Prezentacja sposobu pracy w grupie.
2. Wprowadzenie do zagadnienia OOP (Object Oriented Programming). Pojęcie klasy i instancji klasy.
3. Dozwolone elementy zawartości klas na przykładzie programu rejestracji uczestników szkolenia
4. Dziedziczenie i polimorfizm

## Dzień 4

1. Kontynuacja nauki języka C# - OOP (Object Oriented Programming)
  - Abstrakcja: interfejsy
  - Abstrakcja: klasy abstrakcyjne
  - Przeciążanie metod

2. Dobre praktyki pisania kodu – wzorce projektowe, zasady SOLID

## **Dzień 5**

1. Zaawansowane aspekty C# i platformy .NET przydatne w codziennej pracy
  - Typy generyczne i kolekcje
  - Atrybuty
  - Interpolacja stringów
  - Wyrażenia lambda
  - Linq
2. Podział aplikacji na projekty. Ustawianie referencji do innych projektów. Pobieranie paczek z globalnego repozytorium poprzez narzędzie Nuget Package Manager
3. Wyjaśnienie zasad i działania protokołu http(s). 3 cechy tego protokołu.
4. Przedstawienie podejścia Microsoft do tworzenia aplikacji webowych. Omówienie ogólnych informacji związanych z IIS, ASP .NET WebForms i ASP .NET MVC. Teoretyczne przedstawienie działania frameworka ASP .NET MVC 5 (droga od „żądania” do „odpowiedzi”)

## **Dzień 6**

1. Stworzenie prostej aplikacji internetowej ASP .NET MVC 5
2. Routing po stronie serwera
3. Podział logiki na kontrolery i akcje. Przekazywanie parametrów do metod akcji.
4. Zwracanie dynamicznego widoku przy użyciu silnika Razor. Przekazywanie modelu do widoku

## **Dzień 7**

1. Wykorzystywanie gotowych komponentów html poprzez obiekt @Html w silniku Razor. Poznanie pojęcia Dependency Injection na przykładzie kontenera DI Autofac.
2. Budowanie formularzy. Wyjaśnienie mechanizmu Post-Back (nagłówek x-www-form-urlencoded). Mechanizm zapobiegania atakowi CSRF. Walidacja danych formularza.
3. Konfiguracja lokalnego serwera IIS i instalacja nowej aplikacji na serwerze.
4. Utworzenie kilku stron w aplikacji ASP .NET MVC. Wykorzystanie mechanizmu layout'u. Użycie partial view, action view. Przekierowania pomiędzy stronami (akcjami kontrolerów). Tworzenie linków w widokach (ActionLink, RouteLink)

## **Dzień 8**

1. Instalacja bazy danych MS SQL Server. Utworzenie relacyjnej bazy danych.
2. Język zapytań SQL. Budowanie kwerend za pośrednictwem SQL Management Studio. Zapytania: SELECT FROM, WHERE, ORDERBY, DISTINCT, JOIN, GROUPBY, UPDATE, DELETE
3. Podłączenie aplikacji do bazy danych za pośrednictwem ADO.NET
4. Stworzenie prostej biblioteki umożliwiającej pobieranie danych i mapowanie do obiektów z użyciem komponentu DataReader

## **Dzień 9**

1. Przedstawienie technologii ORM na przykładzie Entity Framework.
2. Implementacja połączenia do bazy danych z użyciem EF oraz podejścia Code First. Mechanizm migracji zmian struktury bazy danych.
3. Podstawy technologii Html5: struktura dokumentu i nowe znaczniki
4. Stylowanie aplikacji w oparciu o CSS3
  - Rodzaje selektorów
  - Nadawanie wyglądu elementom strony
  - Model pudełkowy
  - Pozycjonowanie komponentów
5. Preprocesory CSS na przykładzie SASS
6. Poznanie biblioteki Bootstrap. Stylowanie komponentów z użyciem biblioteki. Wykorzystanie kolumnowego układu kontenerów.

## **Dzień 10**

1. Pojęcie Responsive Web Design. Instalacja biblioteki Bootstrap poprzez użycie narzędzia Node Package Manager. Zastosowanie styli biblioteki Bootstrap. Zainstalowanie Visual Studio Code jako edytora JS, Css, Html.
2. Użycie 12-kolumnowego systemu szkieletu strony biblioteki Bootstrap
3. Nowoczesne podejście do tworzenia styli – preprocesory Less i Sass. Omówienie potrzeby stosowania preprocesorów. Wykorzystanie SASS do stworzenia stylu aplikacji. Użycie funkcjonalności variables, mixins oraz import.
4. Omówienie podejścia Single Page Application. Dodanie do aplikacji kontrolerów WebAPI2. Wyjaśnienie mechanizmu AJAX i omówienie popularnych formatów przesyłania danych: xml i json. Przedstawienie i użycie narzędzia PostMan do wykonywania zapytań http.

## **Dzień 11**

1. Podstawy języka javascript. Dodawanie skryptów inline do strony html. Tworzenie skryptów JS, w tym:
  - Zmienne
  - Warunki, pętle, operatory logiczne i arytmetyczne
  - Komponenty wyjścia: konsola i popupy
  - Funkcje
  - Enkapsulacja
  - Analiza aplikacji klienta z użyciem dodatku Chrome Developer Tools
2. Manipulacja drzewem DOM oraz zdarzenia w przeglądarce. Pojęcie callback. Użycie timeout i interval
3. Obiektowość w javascript. Wzorzec prototype
4. Dobre praktyki i wzorce projektowe stosowane w pisaniu aplikacji w języku java script. Wyjaśnienie jak działa event loop w silnikach JS.

## **Dzień 12**

1. Wprowadzenie do biblioteki jQuery. Dodawanie biblioteki do strony. Dynamiczne tworzenie komponentów drzewa DOM. Rejestrowanie handlerów zdarzeń okna.
2. Zapytania asynchroniczne http do WebAPI z wykorzystaniem biblioteki jQuery
3. Komponenty javascript biblioteki bootstrap

#### 4. Użycie nowych funkcji standardu HTML5

- Web storage
- Canvas
- Web worker
- Web socket

### Dzień 13

1. Omówienie potrzeby stosowania frameworków MVC w warstwie klienta. Porównanie dostępnych na rynku frameworków: AngularJS 1.6, AngularJS (obecnie 4), React, Knockout, VueJS.  
Przygotowanie środowiska deweloperskiego. Przedstawienie narzędzia NPM i Webpack.
2. Użycie AngularCLI do wygenerowania szablonu aplikacji. Omówienie struktury projektu. Wyświetlenie tekstu „hello, world” po stronie klienta. Uruchamianie aplikacji w wersji developerskiej i produkcyjnej
3. Wprowadzenie do języka typescript:
  - Zmienne i funkcje
  - Klasy i interfejsy
  - Enumeracje
  - Moduły
  - Dekoratory
4. Stworzenie struktury prostych komponentów wyświetlających informacje na stronie. Rodzaje data bindingu. Stylowanie komponentów. Dyrektywa \*ng-If/else i \*ng-For

### Dzień 14

1. Obsługa zdarzeń
2. Komunikacja komponentów: @Input, @Output, EventEmitter. Pojęcie transkluzji
3. Routing na stronie: tworzenie stanów, generowanie szablonu i menu, przekierowania z poziomu kontrolerów, odczytywanie parametrów stanu.
4. Dependency injection i serwisy

### Dzień 15

1. Formularze template-driven
2. Formularze model-driven (reactive forms)
3. Dyrektywy i pipe'y
4. Biblioteka RxJS i pojęcie observable. Wyjaśnienie potrzeby „obserwowania” zmian na obiektach.

### Dzień 16

1. Obsługa ajax w bibliotece AngularJS z użyciem RxJS
2. Angular bootstrap – gotowe komponenty bootstrap dla frameworka Angular
3. Zapewnienie jakości aplikacji. Omówienie potrzeby stosowania testów. Prezentacja podziału testów aplikacji.
4. Test driven design. Pisanie testów jednostkowych w C# i AngularJS



# Plan

Dzień 1	Sesja 1	Wprowadzenie
	Sesja 2	Architektura nowoczesnych systemów
	Sesja 3	Przygotowanie środowiska pracy
	Sesja 4	Działanie kompilatora
Dzień 2	Sesja 1	Podstawy języka C#
	Sesja 2	
	Sesja 3	Git
	Sesja 4	
Dzień 3	Sesja 1	Agile Scrum
	Sesja 2	Podstawy Object Oriented Programming w C#
	Sesja 3	
	Sesja 4	
Dzień 4	Sesja 1	Abstrakcja w OOP
	Sesja 2	
	Sesja 3	
	Sesja 4	Dobre praktyki – wzorce i zasady SOLID
Dzień 5	Sesja 1	Zaawansowane aspekty C# i .NET
	Sesja 2	Praca z wieloma projektami w jednej solucji
	Sesja 3	HTTP(s)
	Sesja 4	Przedstawienie IIS i .NET MVC
Dzień 6	Sesja 1	Podstawowa aplikacja .NET MVC
	Sesja 2	
	Sesja 3	
	Sesja 4	
Dzień 7	Sesja 1	Post-Back w MVC
	Sesja 2	Formularze i walidacja
	Sesja 3	Layout w projekcie
	Sesja 4	Partial view i action view
Dzień 8	Sesja 1	Tworzenie bazy danych
	Sesja 2	SQL
	Sesja 3	ADO.NET
	Sesja 4	Budowanie własnego mechanizmu pobierania danych z SQL
Dzień 9	Sesja 1	Entity Framework
	Sesja 2	Code First w Entity Framework
	Sesja 3	Html - podstawy
	Sesja 4	CSS - podstawy
Dzień 10	Sesja 1	Bootstrap – podstawy
	Sesja 2	RWD z użyciem Bootstrap
	Sesja 3	LESS i SASS
	Sesja 4	WebAPI2
Dzień 11	Sesja 1	Podstawy języka JavaScript
	Sesja 2	
	Sesja 3	
	Sesja 4	Dobre praktyki i wzorce projektowe w aplikacjach JS
Dzień 12	Sesja 1	jQuery
	Sesja 2	Ajax z użyciem jQuery
	Sesja 3	Komponenty js w bibliotece Bootstrap
	Sesja 4	Funkcje js w HTML5

Dzień 13	Sesja 1	Angular 4 – przygotowanie środowiska i język typescript
	Sesja 2	
	Sesja 3	
	Sesja 4	
Dzień 14	Sesja 1	Angular4 – elementy podstawowe
	Sesja 2	
	Sesja 3	
	Sesja 4	
Dzień 15	Sesja 1	Angular4 – formularze i dyrektywy
	Sesja 2	
	Sesja 3	
	Sesja 4	
Dzień 16	Sesja 1	Ajax w Angular 4
	Sesja 2	Angular bootstrap
	Sesja 3	Prezentacja testów jednostkowych
	Sesja 4	TDD w C# i AngularJS