

Pracownik	Przełożony		Data rozmowy
Paweł Tymczyk	Szymon Standarski		26.03.2021

Fakty		Konsekwencje	Oczekiwanie
Bardzo słaba znajomość języka JavaScript	<p>Brak standardowych podstaw, które są wymagane przez na stanowisko Junior developera. Brak znajomości zagadnień (prototypy, primitive root object, łańcuch prototypów i wiele innych z pojęć z tego poziomu). Nie identyfikuje część typów prymitywnych oraz złożonych. Nie kojarzy, ale nie umie korzystać z podstawowych metod wbudowanych w prototyp array. Wiedza teoretyczna jest bardzo mała. Brakuje podstaw – tych naprawdę podstawowych. Nie zna zastosowania arrow functions. Nie wie jak destrukuryzować obiekty. Nie umie zastosować spread operatora oraz rest operatora. Nie wie jak działa event loop oraz nie umie przewidzieć kolejności wywoływania kodu synchronicznego. Zadania praktycznego tak naprawdę samodzielnie nie zrobił żadnego. Umie posiłkować się dokumentacją lecz nie czyta kodu dokładnie. Myli działanie event loop z mechanizmami angulara. Błędnie korzysta z większości metod wbudowanych jak map, filter itp...</p> <p>Nie zna różnicy pomiędzy strict vs loose equality. Nie zna pojęcia memoizacja i nie rozumie jak działa.</p>	<p>W przypadku gdy Pracownik nie osiągnie oczekiwanego poziomu wiedzy (Junior), konsekwencją będzie rozwiązanie umowy.</p>	<p>Będzie znał podstawy języka JavaScript i podstawowe pojęcia z nim związane np. hoisting, scope, this, prototypy, łańcuch prototypów itp</p> <p>Będzie znał wszystkie rodzaje typów danych wbudowanych w język JavaScript</p> <p>Będzie znał podstawowe metody wbudowane w prototyp Array. Będzie umiał sprawnie ich używać.</p> <p>Będzie znał sposoby destrukuryzacji obiektów.</p> <p>Będzie wiedział czym są spread i rest operatory. Czym są i do czego można ich używać.</p> <p>Będzie wiedział czym jest event loop</p> <p>Będzie wiedział jak działa asynchroniczność w języku JavaScript oraz jak z nią pracować. Będzie znał podstawowe metody na obiekcie Prototype oraz jak pracować z async/await</p> <p>Będzie umiał sprawnie poruszać się w dokumentacji i tam wyszukiwać informacji.</p> <p>Będzie znał podstawowe wzorce projektowe występujące podczas programowanie w języku</p>
Bardzo słaba znajomość języka TypeScript	<p>Bardzo słabo wychodzi mu definiowanie typów nawet tych prostych. Potrzebuje pomocy podczas analizowania kodu. Nie umie wytłumaczyć oraz zastosować typów generycznych. Słaba podstawowa wiedza.</p>	<p>W przypadku gdy Pracownik nie osiągnie oczekiwanego poziomu wiedzy (Junior), konsekwencją będzie rozwiązanie umowy.</p>	<p>Będzie znał podstawowe typy z języka Typescript i będzie potrafił z nimi pracować.</p> <p>Będzie znał typy generyczne oraz jak ich używać.</p> <p>Będzie znał podstawowe wbudowane typy generyczne w język TypeScript np. Partial, Readonly</p>

<p>Bardzo słaba znajomość frameworka Angular</p>	<p>Nie wie i nie zna mechanizmów detekcji zmian. Nie wie czym są zmiany w sposób immutable i jakie dają korzyści w codziennej pracy. Nie wie czym jest pipe jednak zna przykład wbudowanych – do formatowania dat. Wie, że można stworzyć własnego pipe. Kojarzy CLI – podstawowe polecenia, do generowania projektu bądź komponentu – jednak nie zna go dokładnie. Myli idee interfejsów i ich przeznaczenie. Wie, że dekorator Pipe jest niezbędny. Teoretycznie zna podział komponentów na logiczne oraz prezentacyjne, ale nie nazywa tego fachowo i w rzeczywistości nie rozumie kluczowych różnic. Wie i zna sposób przekazywania danych poprzez dekorator @Input i więcej nie zna. Po nakierowaniu wie, że można użyć do tego serwisu. Nie zna idei serwisów i nie wie do czego są wykorzystywane. Korzystał z nich w pracy codziennej, ale pomimo tego nie rozumie ich przeznaczenia. Wie, że logikę związana z serwerem trzyma w serwisach. Zna przeznaczenie modułów. Tworzy własne moduły, aczkolwiek wiedza bardzo podstawowa. Zna elementy składniowe modułów jak imports, exports, ale nie zna innych – providers, bootstrap itp. Kojarzy czym są testy jednostkowe, ale słabo umie to wytłumaczyć. Nie umie odróżniać testów jednostkowych. Nie zna testów integracyjnych oraz e2e, automatycznych oraz property based. Wie czym jest mockowanie. Nie wie czym jest stub oraz piramida testów. Nie wie czym jest TDD. Nie zna podejścia AAA. Umie (ze sporą pomocą) budować dyrektywy aczkolwiek nie umie wytłumaczyć czym są i do czego powinny.</p>	<p>W przypadku gdy Pracownik nie osiągnie oczekiwanego poziomu wiedzy (Junior), konsekwencją będzie rozwiązanie umowy.</p>	<p>Będzie znał różnice pomiędzy dyrektywami a komponentami. Będzie potrafił powiedzieć gdzie lepiej jest użyć komponent a gdzie lepiej dyrektywę. Będzie potrafił opowiedzieć jak działa mechanizm detekcji zmian. Będzie potrafił napisać własny pipe angularowy. Będzie wiedział czym dokładnie są testy jednostkowe i do czego służą. Będzie potrafił wymienić rodzaje testów używanych do pracy z aplikacjami frontendowymi.</p>
---	--	--	--

<p>Bardzo słaba wiedza z zakresu HTML i CSS</p>	<p>Nie umie powiedzieć wprost w jaki sposób wyśrodkować element na stronie. Nie korzysta z grida, ale nie umie wytłumaczyć jak działają poszczególne dyrektywy. Nie wie jak wybierać poszczególne wiersze bądź kolumny grida. Nie wie czym jest semantyczny HTML – nie identyfikuje tagów. Nie zna znaczników jak video, audio. Nie zna problemów z display: inline-block – pojawianie się pustej przestrzeni. Nie zna podstawowych zachowań dla flex'a. Nie zna float: left, float: right oraz zachowania kontentu przy stosowaniu ich. Nie wie jak zaimplementować popularne mechanizmy, które są używane w aplikacjach – przyklejanie navbara do góry okna, menu rozwijane, ...itd. Nie zna różnic pomiędzy position: fixed, absolute, static, relative. Wie jak zarządzać wartościami z-index w aplikacji w sposób skalowalny i wygodny. Korzystał z sassa. Nie korzystał z petl, korzystał z instrukcji warunkowych. Zna podstawy działania, ale słabo to tłumaczy. Nie wie jakie właściwości powinniśmy animować. Nie rozumie i nie zna zastosowania selektorów css.</p>	<p>W przypadku gdy Pracownik nie osiągnie oczekiwanego poziomu wiedzy (Junior), konsekwencją będzie rozwiązanie umowy.</p>	<p>Będzie potrafił wymienić większość znaczników dostępnych w HTML5 (zwłaszcza tych które doszły w wersji 5) Zapozna się z flex layout i grid layout. Będzie potrafił zastosować jeden z ww. layoutów na dowolnym przykładzie testowym.</p>
<p>Bardzo słaba wiedza z zakresu biblioteki RxJS</p>	<p>Myli RxJs z NgRx. Nie zna różnic BehaviourSubject vs Subject. Nie zna ReplySubject oraz innych. Myli podstawowe pojęcia związane z biblioteką RxJs. Wie jak można zaimplementować caching ostatniej wartości, ale nie mówi tego wprost. Słabo zna pojęcia związane z programowaniem reaktywnym (obserwator, strumień, observable) oraz tych związanych już z biblioteką rxjs (operatory, pipe). Nie zna podstawowych operatorów oraz ich przeznaczenia. Bardzo duże braki w podstawach. Nie zna programowania reaktywnego oraz paradygmatów.</p>	<p>W przypadku gdy Pracownik nie osiągnie oczekiwanego poziomu wiedzy (Junior), konsekwencją będzie rozwiązanie umowy.</p>	<p>Będzie potrafił omówić różnice pomiędzy RxJS i NgRx. Będzie potrafił omówić i użyć większość operatorów i typów rozszerzających obiekt Subject, dostępnych w RxJS</p>

<p>Bardzo słaba ogólna wiedza na temat WEBu</p>	<p>Nie wie czym jest PWA. Nie wie do czego służy mechanizm CORS. Nie zna technik takich jak JSONP. Nie rozumie na czym polega atak XSS i nie wie, że angular przed nim chroni domyślnie. Nie wie czym jest sanitizing.</p> <p>Nie wiem czym jest MITM attack. Nie wie czym jest phishing. Nie wie czym jest service worker oraz web worker.</p>	<p>w przypadku gdy Pracownik nie osiągnie oczekiwanego poziomu wiedzy (Junior), konsekwencją będzie rozwiązanie umowy</p>	<p>Będzie potrafił wyjaśnić jak działa mechanizm CORS, co nam daje i jak należy z nim pracować. Będzie potrafił opowiedzieć czym jest PWA i RWD. Będzie wiedział czym sa web workery i service workery i jaka jest między nimi różnica. Będzie potrafił wymienić i użyć różnych metod przechowywania danych na urządzeniu użytkownika.</p>
<p>ustalenia, plan działań:</p> <p>Paweł otrzyma opiekuna w osobie Sebastiana Zdunka. Bedą się kontaktować raz na tydzień w celu omówienia zadań i zakresu tematów do nauki. Dodatkowo podczas spotkań Sebastian będzie weryfikował czy w poprzednim tygodniu Paweł przyswoił wiedzę z którą miał się zapoznać. Opiekun może Pawłowi przekazać tematy do nauki ale mogą to być także krótkie zadania algorytmiczne które będą wykorzystywały omawiane technologie. Dodatkowo co dwa tygodnie będą organizowane spotkania z Team Leaderem w celu omówienia postępu w nauce. Raz na miesiąc będzie organizowane spotkanie w którym wezmą udział: Monika Sobiecka (HRBP), Paweł Tymczyk, Sebastian Zdunek (opiekun) i Szymon Standarski (Team Leader) w celu omówienia postępu nauki. Po 3 miesiącach nastąpi kolejna weryfikacja wiedzy. W takim samym schemacie w jakim była przeprowadzona przed programem naprawczym. Na tej rozmowie zostanie zweryfikowany progress w</p>			