

**UNIwersytet Ekonomiczny we Wrocławiu**  
**Wydział Zarządzania, Informatyki i Finansów**

---

Kierunek: Informatyka w Biznesie (IwB)

**PROJEKT ZALICZENIOWY Z CZĘŚCI ĆWICZENIOWEJ I  
LABORATORYJNEJ**

**Justyna Koziej (149679)**  
**Tomasz Lipowski (163304)**  
**Tomasz Połomski (163309)**

**INTEGRACJA APLIKACJI I SYSTEMÓW**

**Prowadzący:**

**inż. Paweł Szpak, mgr Paweł Kuczyński**

---

**WROCLAW 2016**

## 1. Spis treści

1. Spis treści .....	2
2. Użyte technologie.....	3
2.1. PHP – Technologia użyta do zaprojektowania HUB-a.....	3
2.2. JSON – Technologia użyta do odczytu danych z providerów .....	4
3. Architektura systemu.....	7
4. Opis HUB-a.....	8
5. Napotkane problemy .....	8
6. Adres do repozytorium .....	8
7. Podział pracy w grupie .....	9
8. Co byśmy zmienili gdybyśmy robili ten projekt jeszcze raz? .....	9

## 2. Użyte technologie

### 2.1. PHP – Technologia użyta do zaprojektowania HUB-a

PHP to skryptowy język programowania służący głównie do tworzenia dynamicznych **stron internetowych**. Historia tego języka ma swój początek w 1994 roku, kiedy to Rasmus Lerdorf stworzył go jako zestaw narzędzi do obsługi swojej własnej **strony WWW**. Prawdziwą popularność natomiast język PHP zdobył gdy W 1997 roku projektem zainteresowali się dwaj izraelscy programiści, Zeev Suraski i Andi Gutmans. Zdecydowali oni przepisać kod PHP od nowa, korzystając z pomocy istniejącej już wówczas społeczności PHP. W czerwcu 1998 roku ogłosili PHP 3.0 jako następcę PHP/FI. Najważniejszą cechą tej aplikacji była jej modułowość i łatwa możliwość rozszerzania funkcjonalności języka. Dziesięć lat od pierwszej wersji ukazała się piąta odsłona języka PHP.

PHP to język programowania, który się znacznie wyróżnia na tle innych, konkurujących z nim języków. Jedną z jego zalet jest to, że jest on darmowy, każdy może go pobrać zainstalować i używać bez żadnych ograniczeń. Język PHP jest łatwy w nauce i znacznie szybszy od innych języków. Co więcej, jest możliwość wstawiania kodu PHP bezpośrednio do dokumentu HTML co jest dużym ułatwieniem w tworzeniu skryptów. PHP pozwala na współpracę z wieloma bazami danych i świetnie działa z bazą MySQL. Ponad to interpreter PHP może działać jako moduł najpopularniejszego serwera Apache, co daje mu ogromną przewagę nad pozostałymi językami.

## 2.2. JSON – Technologia użyta do odczytu danych z providerów

**JSON** (JavaScript Object Notation) jest prostym formatem wymiany danych. Zapis i odczyt danych w tym formacie jest łatwy do opanowania przez ludzi. Jednocześnie, z łatwością odczytują go i generują komputery. Jego definicja opiera się o podzbiór języka programowania JavaScript, Standard ECMA-262 3rd Edition - December 1999. JSON jest formatem tekstowym, całkowicie niezależnym od języków programowania, ale używa konwencji, które są znane programistom korzystającym z języków z rodziny C, w tym C++, C#, Java, JavaScript, Perl, Python i wielu innych. Właściwości te czynią JSON idealnym językiem wymiany danych.

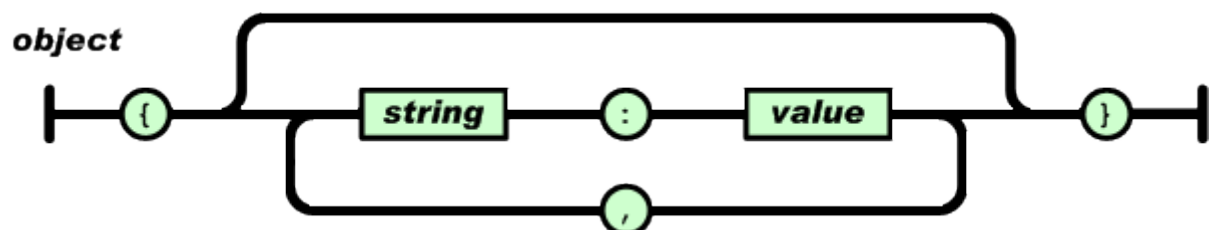
JSON powstał w oparciu o dwie struktury:

- Zbiór par nazwa/wartość. W różnych językach jest to implementowane jako *obiekt*, rekord, struktura, słownik, tabela hash, lista z kluczem, albo tabela asocjacyjna.
- Uporządkowana lista wartości. W większości języków implementuje się to za pomocą *tabeli*, wektora, listy, lub sekwencji.

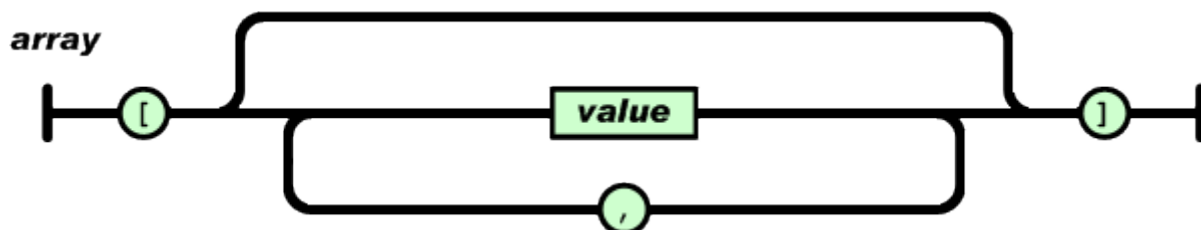
Wspomniane struktury danych są uniwersalne. Prawie wszystkie nowoczesne języki programowania posługują się nim w tej lub innej formie. Ma to sens, by format danych, który jest przenośny pomiędzy różnymi językami programowania opierał swoją budowę na wspomnianych strukturach.

W przypadku formatu JSON, przybierają one następujące formy:

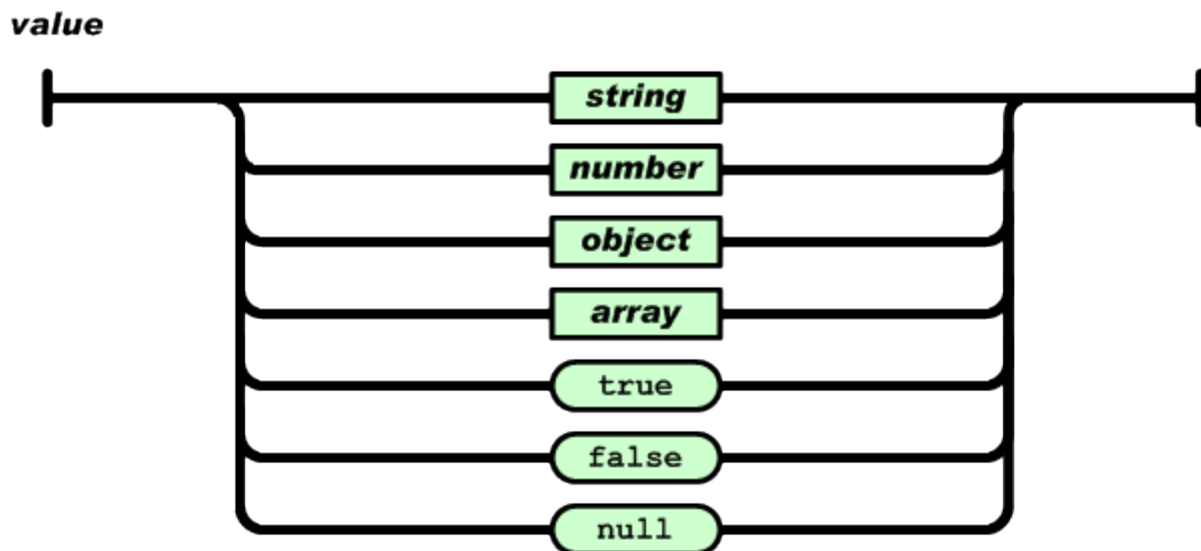
*Obiekt* jest nieuporządkowanym zbiorem par nazwa/wartość. Opis obiektu rozpoczyna { (lewa klamra) a kończy } (prawa klamra). Po każdej nazwie następuje : (dwukropek) oraz pary nazwa/wartość, oddzielone , (przecinkiem).



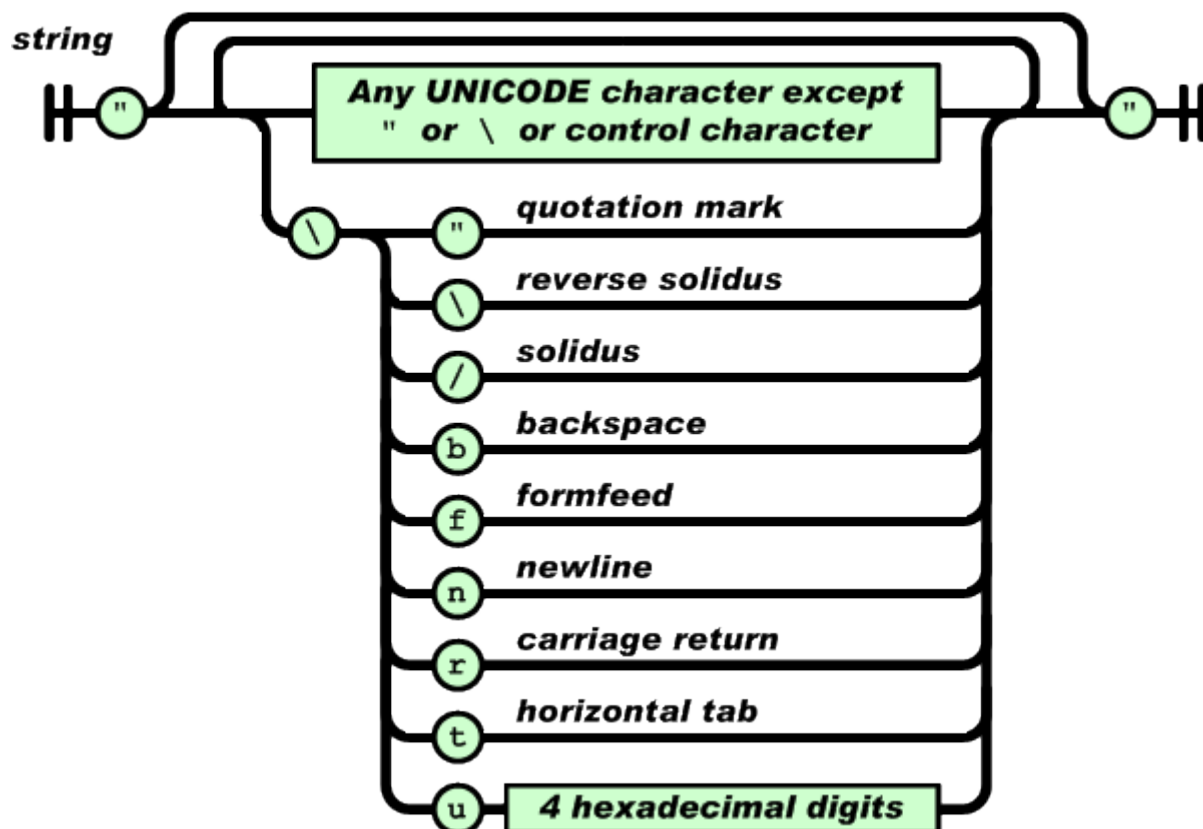
*Tabela* jest uporządkowanym zbiorem wartości. Opis tabeli rozpoczyna znak [ (lewy nawias kwadratowy) a kończy znak ] (prawy nawias kwadratowy). Poszczególne wartości rozdzielane są znakiem , (przecinek).



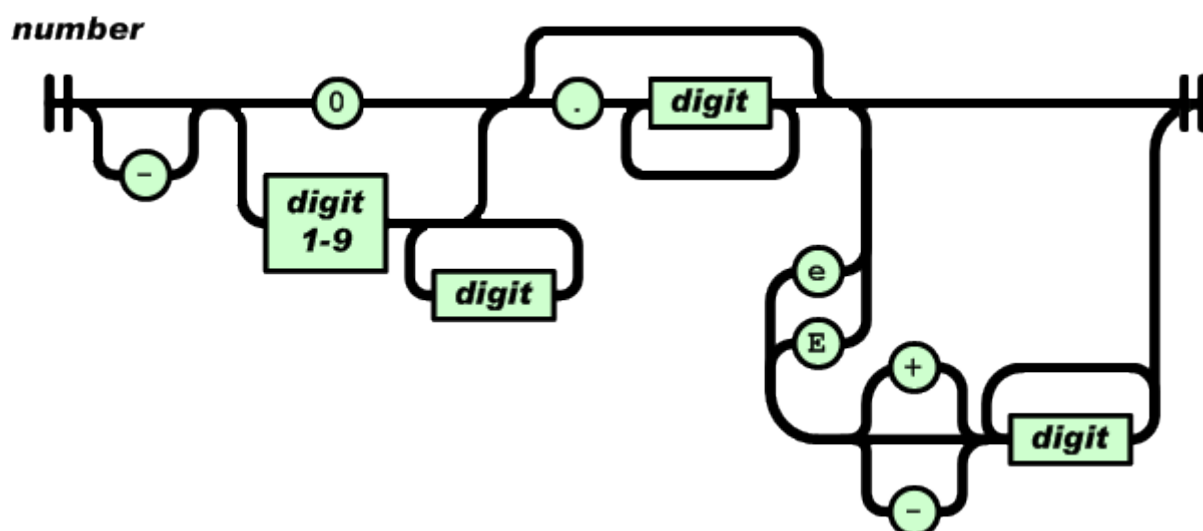
*Wartość* to *łańcuch znakowy*, którego początek i koniec oznacza podwójny cudzysłów, lub *liczba*, lub wartość true (prawda) lub false (fałsz) lub null, lub *obiekt* lub *tabela*. Struktury te można zagnieżdżać.



*Łańcuch znakowy* jest zbiorem zera lub większej ilości znaków Unicode, opakowanym w podwójne cudzysłowy, stosujących znak odwrotnego ukośnika jako początek sekwencji specjalnej (escape). Pojedynczy znak jest reprezentowany jako łańcuch jednoznakowy. Łańcuch znakowy JSON jest podobny do łańcucha znakowego C lub Java.

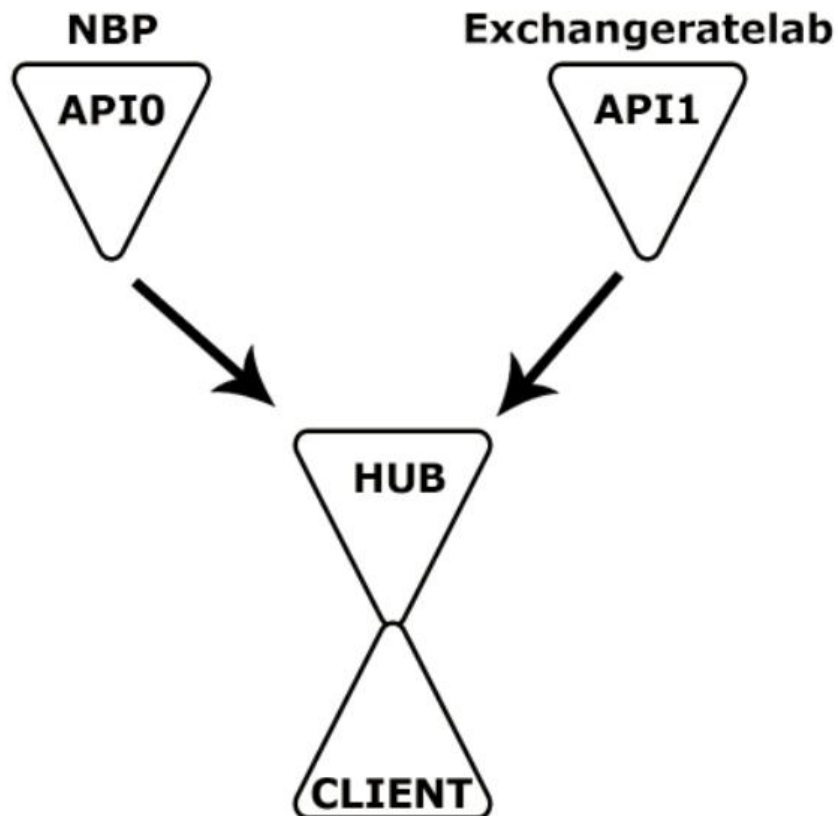


Liczby zapisywane w formacie JSON są bardzo podobne do liczb w języku C lub Java, poza tym wyjątkiem, że nie używa się formatów ósemkowych i szesnastkowych.



Wolne miejsce (spacje, znaki tabulatora, itp.) można wstawić między dowolną parę składowych. Poza kilkoma detalami dotyczącymi kodowania, na tym kończy się opis języka JSON.

### 3. Architektura systemu



Rysunek 1 Architektura systemu

System składa się z dwóch zewnętrznych providerów API: Narodowego Banku Polskiego oraz zagranicznego serwisu Exchangeratelab w wersji darmowej.

Następnym elementem systemu jest HUB/Client wykonany w technologii PHP który wczytuje dane generowane przez providerów API w formacie JSON, następnie przetwarza je na ujednolicone dane (wspólne dla obydwóch providerów).

Kończącym działaniem systemu jest wyświetlenie otrzymanych danych wyświetlenie ich przy użyciu kodu HTML w formie przyjaznej dla użytkownika końcowego.

## 4. Opis HUB-a

W pierwszej kolejności do HUB-a wczytywane są dwa źródła w formacie JSON z zewnętrznych providerów, odbywa się to po nieszyfrowanym połączeniu HTTP.

Następnie kursy walut pobrane z tychże źródeł są dekodowane funkcją *json\_decode*, która zakodowany ciąg JSON konwertuje do zmiennej PHP w przypadku Narodowego Banku Polskiego jest to zmienna *\$kursy\_nbp*, a w przypadku portalu Exchangeratelab zmienna nosi nazwę *\$kursy\_exchangeratelab*.

Kolejnym krokiem jest łączenie dwóch tablic w jednolitą pod zmienną *\$items* następnie połączone już ze sobą dwie tablice poddane są losowemu wymieszaniu danych jeśli chodzi o kolejność wyświetlania.

Nasz HUB zawiera również dwie funkcje (każda funkcja dla swojego providera), które to czynią najważniejszą rzecz w naszym HUB-ie tzn. konwertują tablice podpięte z API providerów do jednolitej formy przyjętej przez naszego HUB-a.

Jako że HUB łączy również funkcję klienta ostatnim krokiem jest wyświetlenie już przetworzonych danych w przyjaznej formie dla użytkownika końcowego korzystającego z naszego rozwiązania.

## 5. Napotkane problemy

Największym problemem w implementacji był dostęp do źródeł zewnętrznych (API). W pierwotnym założeniu mieliśmy korzystać z serwisu Allegro oraz Ebay, niestety o ile z serwisem Allegro nie było problemu z uzyskaniem dostępu do ich API, to Ebay robił duże problemy. Dlatego też postanowiliśmy skupić się na branży bankowej tzn. kursach walut czego przykładem jest obecny projekt.

Choć również w tym wypadku nie było to łatwe zadanie jeśli chodzi o znalezienie darmowego API w celu wykonania projektu. Narodowy Bank Polski udostępnia bez żadnych opłat dostęp do swojej bazy kursów walut, natomiast w wersji darmowej API Exchangeratelab możemy wykonać tylko 50 zapytań dziennie.

## 6. Adres do repozytorium

[https://github.com/justomtom/projekt\\_ips](https://github.com/justomtom/projekt_ips)



## 7. Podział pracy w grupie

Jeśli chodzi o podział pracy w naszej grupie nie był on sztywno obwarowany, każdy z uczestników projektu wychodził z własnymi pomysłami na forum gdzie były one dyskutowane oraz analizowane pod różnymi kątami.

Część projektowa jak i dokumentacja powstała wspólnymi siłami wszystkich członków grupy

## 8. Co byśmy zmienili gdybyśmy robili ten projekt jeszcze raz?

Gdybyśmy mieli do wykonania jeszcze raz taki sam projekt przygotowalibyśmy własne API ponieważ jest to mniej czasochłonne i pracochłonne biorąc pod uwagę to iż każdy z naszych członków zespołu pracuje zawodowo natomiast weekendy spędzamy na uczelni i ciężko nam się spotkać w ciągu tygodnia by działać nad zadaniami projektowymi.

Jednak na pewno wykonanie zadania korzystając z zewnętrznych dostawców było o wiele ciekawszym doświadczeniem oraz przede wszystkim mogliśmy pracować na „żywych organizmach”.

Jeśli chodzi o samą funkcjonalność projektu, jeśli byłby on pisany dla klienta komercyjnego na pewno skorzystalibyśmy z płatnych providerów, którzy dostarczają dane LIVE, do tego użylibyśmy technologii AJAX która pozwalałaby w czasie rzeczywistym wyświetlać klientowi końcowemu wyniki z odświeżając je co np. 1sek.