IAS projekt

Raport

nr indeksów członków grupy: 149011

użyte technologie/ języki programowania: PHP/Laravel/MySQL/REST

schemat oraz opis architektury systemu: architektura systemu składa się z dwóch osobnych instancji baz danych (*myprovider1* i *myprovider2*). Obie bazy danych składają się z jednej encji, w których przechowywane są dane o książkach. Ponadto istnieją dwie aplikacje, które pobierają dane z bazy danych MySQL i udostępniają je w formacie JSON. Następnie w aplikacji integrującej (hub) pobierane są dane z obu aplikacji i integrowane są w formie nowej struktury danych. Podobnie jak w przypadku providerów, hub udostępnia całą kolekcję danych z obu aplikacji, w formie pliku JSON.

opis dostawców, struktura encji: architektura systemu składa się z dwóch osobnych instancji baz danych (*myprovider1* i *myprovider2*). W obu bazach danych przechowywane są dane na temat książek. Obie bazy składają się z jednej encji różniącej się między sobą nazwami kolumn. W bazie *myprovider1* znajdują się dane w języku angielskim (nazwy kolumn po angielsku) oraz rekordy również po angielsku. Natomiast w bazie *myprovider2* znajdują się dane w encji o polskich nazwach kolumn wraz z polskimi tytułami książek.

opis huba, przebieg integracji encji, struktura wynikowa encji: tak jak zostało opisane wyżej integracja danych w hubie polega na pobraniu z obu aplikacji providerów danych w formacie JSON. Następnie w klasie *ItemController* została napisana logika zamiany struktury danych z obu providerów do struktury huba. Ujednolicona struktura huba wygląda następująco: *provider_item_id, tytul, autorzy, opis, strony, wydawca*. Następnie po pobraniu kolekcji danych z aplikacji obu providerów i ujednolicenia danych według nowej struktury huba, dane udostępniane są w formacie JSON.

napotkane problemy:

- Zmiana ścieżek w plikach konfiguracyjnych: Apache (httpd.conf), Apache (httpd-ssl.conf), Apache (httpd-xampp.conf), phpMyAdmin (config.inc.php)
- podwójny "return" (return(items);) w metodzie index() w klasie ItemController.php
- problem przy pushu zmian do repozytorium

adres do repozytorium: https://github.com/santiarzo/IAS_santiarzo

podział pracy w grupie: projekt realizowany był przez jedną osobę

dodatkowo: Co byśmy zmienili gdybyśmy robili ten projekt jeszcze raz?:

- Prawdopodobnie zrobiłbym integrację danych przychodzących w różnych formatach (np. JSON i XML) oraz huba który integrowałby dane o różnym formacie
- Ewentualnie spróbował bym napisać podobną aplikację w Javie "od zera", ponieważ ciężko jest się połapać w kodzie Laravela bez wcześniejszej styczności z tym

frameworkiem oraz z PHP. Jest wiele klas, jak np. VerifyCsrfToken.php które np.odnoszą się do bezpieczeństwa całej aplikacji a które utrudniają ogólny ogląd na cały framework oraz tylko niezbędne klasy wraz z metodami potrzebne to zakodowania providerów oraz huba.