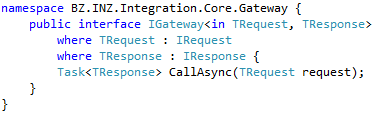
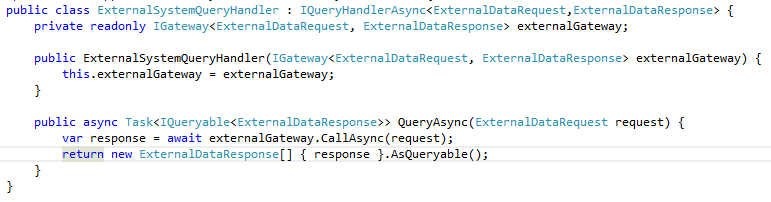
Wzorzec projektowy Gateway należy do grupy wzorców strukturalnych. Jest to wyspecjalizowana forma innego wzorca – Fasady. Jego głównym założeniem jest uproszczenie komunikacji z systemami zewnętrznymi, z naciskiem na wysyłanie i pobieranie danych.

Gateway separuje skomplikowany system komunikacji z zewnętrzną aplikacją od samego faktu wysłania bądź pobrania informacji. Wystawia on prosty interfejs, w którym definiowane są struktury wiadomości wychodzącej i przychodzącej. Cała logika stojąca za tymi procesami, znajduje się wewnątrz udostępnionych funkcji.

Takie podejście sprawia, że pracując w większych zespołach, nie wszyscy programiści muszą znać specyfikacje każdego z wykorzystywanych API, aby skomunikować się z wybranym systemem. Po zasłonięciu takiej komunikacji przez omawiany wzorzec muszą znać jedynie specyfikę wiadomości, które chcą wysłać i odebrać.

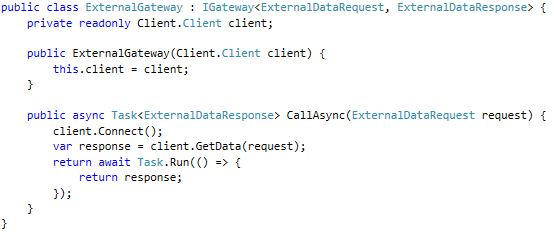
Kolejną korzyścią jaką niesie ze sobą jego implementacja, jest możliwie szybka zmiana sposobu komunikacji z daną aplikacją, bez ingerencji w komponenty, które z tej komunikacji korzystają. Wszystkie informacje o danym API znajdują się tylko w module dedykowanym do połączenia się z nim. Cała reszta komponentów zna tylko strukturę Gateway’a oraz typy wiadomości jakie wysyła i odbiera.





Rysunek . Interfejs IGateway

Rysunek .Wykorzystanie Gateway'a w praktyce.



Rysunek . Implementacja przykładowego Gateway'a

Punktem początkowym wzorca, jest generyczny interfejs IGateway. Jako parametry przyjmuje on dwa rodzaje typów danych. Pierwszym z nich jest typ implementujący interfejs IRequest. Specyfikuje on typ wiadomości wysyłanej do systemu zewnętrznego. Drugim parametrem jest klasa implementująca z kolei interfejs IResponse. Określa ona jakie dane zostaną otrzymane w odpowiedzi.

Następnie przedstawione jest użycie danego Gateway’a w praktyce. Na warstwie aplikacji, rolą jednego QueryHandlera jest pobranie danych z systemu zewnętrznego. Używa on do tego interfejsu IGateway z odpowiednimi parametrami definiującymi typy danych, jakie mając być wysyłane i odebrane. Implementacja interfejsu dostarczona jest przez konstruktor. Kontrola nad tym sprawowana jest poprzez odpowiedni moduł IoC z wykorzystaniem biblioteki Autofac. W metodzie QueryAsync wykorzystywane jest polecenie CallAsync, przy pomocy którego odbywa się komunikacja, z systemem zewnętrznym. Jak widać, metoda ta jest asynchroniczna. Istnieje wiec możliwość równoległego wysłania kilku zapytań do różnych systemów, a następnie poczekania na wszystkie odpowiedzi i zebranie ich razem. Daje to więc szerokie pole do optymalizacji komunikacji.