Tu zacznij

Serwisy WCF

Zadanie 1

Zadanie 2

Zadanie 3

Interfejs API z CRUD

Wykorzystanie MSSQL

API CRUD

Problematyka ćwiczeń

(legacy) aplikacjach, z wykorzystaniem protokołu SOAP.

• 1. Integracja serwisów rozproszonych w aplikacjach

Serwisy WCF / SOAP

Wideo

Zagadnienia

Opisane zagadnienia to:

- Architektury
- Serwis Serwisy WCF

Architektury

pośrednictwem protokołu sieciowego. Systemy rozproszone mogą być przydatne do skalowania przepustowości, skalowania zespołów i przekazywania (potencjalnie drogich zadań lub) odpowiedzialności innym komponentom.

Architektura rozproszona oznacza po prostu, że komponenty systemu są wdrażane osobno i działają poprzez komunikację za

Temat porusza integracje serwisów rozproszonych w technologii WCF/ASMX jako przykład integracji serwisów w "dojrzałych"

Architektura monolityczna

Architektury monolityczne są aplikacjami "wszystko w jednym". Tutaj próbujesz rozdzielić obawy i rozłożyć bazę kodu na komponenty funkcjonalne (Klasy). Zasadniczo budujesz pojedynczy artefakt sieciowy, a następnie próbujesz rozłożyć aplikację na warstwy.

W większości przypadków będziesz mieć warstwę prezentacji, następnie warstwę logiki biznesowej, a na koniec warstwę dostępu do danych. Ideą tej segregacji jest praca z dowolnym elementem architektury niezależnym od tego pod spodem lub powyżej (MVC - omówione niżej). Przykładem takiej aplikacji może być projekt w całości oparty o Laravel.

Monolityczna architektura jest wygodna dla małych zespołów, dlatego wiele startupów wybiera to podejście podczas tworzenia aplikacji. Elementy monolitycznego oprogramowania są ze sobą powiązane i współzależne, co pomaga w samodzielnym oprogramowaniu. Ta architektura jest tradycyjnym rozwiązaniem do budowania aplikacji, ale niektórzy programiści uważają ją za przestarzałą. Uważamy jednak, że architektura monolityczna jest idealnym rozwiązaniem w niektórych okolicznościach.

Wady:

Ta architektura doprowadza do ogromnych problemów z szybkimi odpowiedziami na połączenia. Innym problemem jest trudność w zarządzaniu projektem i bazą kodu, gdy musisz budować, testować lub wdrażać, w zasadzie bawisz się całą bazą kodu, zajmuje to sporo pracy, aby każda funkcja została uruchomiona. Pojedyncze wdrożenie może zawierać komponenty dostępu do danych, komponenty logiki biznesowej, usługi internetowe itd.

Istotną kwestią są koszty związane z infrastrukturą. W przypadku, gdy pojedynczy komponent jest obciążony i wymaga skalowania, trzeba będzie dodać zasoby dla **całej aplikacji**. Oznacza to, że źle działająca część Twojej architektury oprogramowania może doprowadzić do upadku całej struktury lub też płacisz *ciężarówką* \$\$\$ za jej utrzymanie.

Architektura SOA (Service oriented architecture)

Naturalnym przejściem od aplikacji monolitycznej jest zastosowanie architektury opartej na usługach (SOA) (np. Projekt składający się z x liczby małych aplikacji opartych o WCF). Przy takim podejściu rozdzielisz swoją aplikację na mniejsze moduły. Wszystkie usługi działałyby wówczas z warstwą agregacyjną, którą można nazwać magistralą.

Ta architektura ma dobry sposób na oddzielenie i komunikację. Każda usługa może komunikować się przy użyciu standardowej warstwy przetwarzania biznesowego. Ta warstwa oddziela wewnętrzne i zewnętrzne elementy systemu.

Wady:

Ta warstwa agregacji (SOA Bus) jest największym wyzwaniem, z którym trzeba się zmierzyć. Problem polega na dodaniu logiki operacyjnej do magistrali. W miarę jak ta warstwa staje się coraz większa, a do systemu dodawane jest coraz więcej komponentów, pojawiają się problemy z łączeniem systemów.

wypadła z łaski architektów. Ludzie zaczęli spoglądać w stronę monolitycznych architektur lub przenieśli się w stronę Architektury Mikrousług.

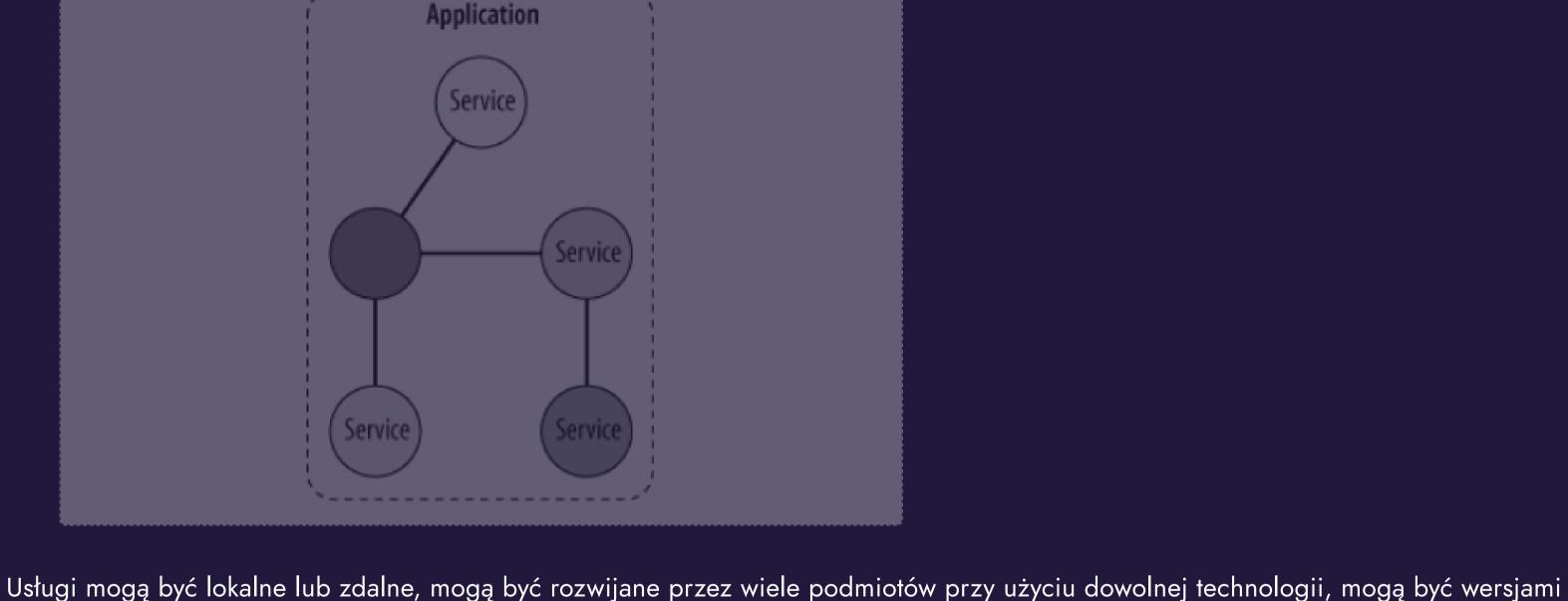
Przy takiej architekturze, można było uzyskać 200 lub 500 odpowiedzi i nic pomiędzy nimi. Nadmiar agregacji sprawia, że ta architektura

Serwis

do obiektów, od komponentów do serwisu. Zorientowanie na usługi (SO) jest abstrakcyjnym zbiorem zasad i najlepszych praktyk w zakresie budowania aplikacji zorientowanych na usługi. Załącznik A zawiera zwięzły przegląd i przedstawia w zarysie motywację do stosowania tej metodologii.

Serwis to jednostka funkcjonalna wystawiona na działanie świata. Pod tym względem jest to kolejny krok ewolucyjny w długiej podróży od funkcji

Aplikacja zorientowana na usługi agreguje usługi w jedną aplikację logiczną, podobnie jak aplikacja zorientowana na komponenty agreguje komponenty, a aplikacja obiektowa agreguje obiekty. Aplikacja zorientowana serwisowo:



przez pośrednika, takiego jak Windows Azure AppFabric Service Bus.

niezależnymi, a nawet mogą być wykonywane na różnych liniach czasowych. Wewnątrz usługi znajdują się koncepcje, takie jak języki, technologie, platformy, wersje i frameworki, jednak pomiędzy usługami dozwolone są tylko zalecane patrole komunikacyjne. Klient usługi jest tylko stroną korzystającą z jej funkcjonalności. Klientem może być dosłownie wszystko, na przykład formularze Windows, WPF

lub klasa Silverlight, strona ASP.NET, lub inna usługa. Klienci i usługi współdziałają poprzez wysyłanie i odbieranie wiadomości. Wiadomości mogą być przesyłane bezpośrednio z klienta do usługi lub

W przypadku WCF, wiadomości są zazwyczaj wiadomościami SOAP. Komunikaty te są niezależne od protokołów transportowych - w przeciwieństwie do usług internetowych, usługi WCF mogą komunikować się za pomocą różnych transportów (nie tylko HTTP). Klienci WCF mogą współdziałać z usługami innymi niż WCF, a usługi WCF mogą współdziałać z klientami spoza WCF. Jeśli jednak rozwiniesz zarówno klienta, jak i

Ponieważ tworzenie usługi jest nieprzejrzyste z zewnątrz, usługa WCF zazwyczaj eksponuje metadane opisujące dostępną funkcjonalność i możliwe sposoby komunikowania się z usługą. Metadane są publikowane w sposób predefiniowany, neutralny technologicznie, np. przy użyciu WSDL (Web Services Description Language) przez HTTP-GET lub branżowego standardu wymiany metadanych przez dowolny protokół. Klient spoza WCF może zaimportować metadane do swojego rodzimego środowiska jako typy rodzime. Podobnie, klient WCF może zaimportować metadane usługi innej niż WCF i wykorzystać je jako natywne klasy i interfejsy CLR.

usługę, zazwyczaj możesz tak skonstruować aplikację, aby obie strony wymagały WCF w celu wykorzystania zalet specyficznych dla WCF.

Biblioteka WCF jest zestawem do tworzenia oprogramowania do rozwijania i wdrażania usług w systemie Windows. WCF zapewnia środowisko

Serwisy WCF

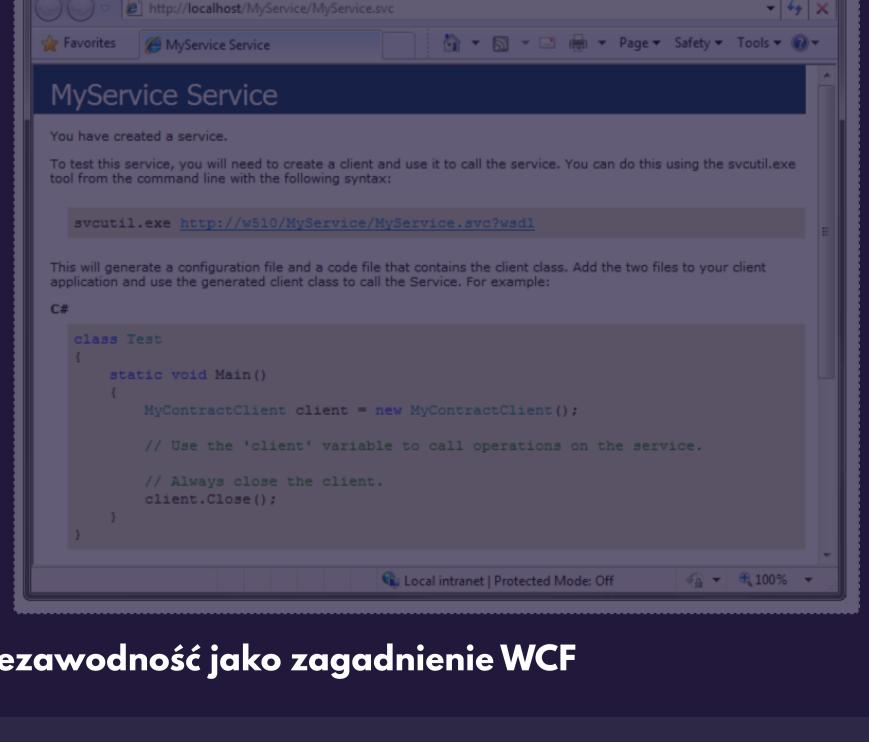
uruchomieniowe (runtime) dla usług sieciowych, pozwalając na wykorzystanie typów Common Language Runtime (CLR) jako usług oraz wykorzystanie innych usług jako typów CLR. Chociaż w teorii można budować usługi bez WCF, w praktyce usługi budowane z wykorzystaniem tej technologii są znacznie łatwiejsze w zarządzaniu. WCF jest zestawem standardów branżowych definiujących interakcje usługowe, konwersje typów i zarządzania różnymi protokołami sieciowymi. WCF zapewnia zatem interoperacyjność usług. Usługa jest jednostką funkcjonalności wystawioną na działanie globalne. Zorientowanie na usługi (Service Oriented) jest abstrakcyjnym zbiorem

zasad i najlepszych praktyk budowania aplikacji. Aplikacja zorientowana na usługi agreguje usługi w jedną logiczną aplikację, podobnie jak aplikacja zorientowana obiektowo agreguje obiekty. Komunikacja (przesyłanie danych) może wystepować w formacie XML albo JSON.

Zalety zastosowania WCF: o programowanie klienta serwisu webowego SOAP,

o projektowanie rozproszonego serwisu webowego SOAP,

- o szybkie udostępnianie danych z bazy w oparciu o podejście sieciowe REST WCF DataServices,
- o optymalizacja i bezpieczeństwo web serwisów SOAP.
- Przykładowy serwis:



miejsca przeznaczenia, nadawca będzie o niej wiedział.

odpowiednio dostosować przepływ wiadomości.

Niezawodność jako zagadnienie WCF Niezawodność komunikatu nie gwarantuje jego dostarczenia. Stanowi ona jedynie gwarancję, że jeśli wiadomość nie dotrze do

WCF i inne technologie zorientowane na usługi rozróżniają pomiędzy niezawodnością transportu a niezawodnością komunikatów. Niezawodność

transportowa (taka jak oferowana przez TCP/IP) zapewnia gwarantowane doręczenie pakietów w trybie "point-to-point" na poziomie pakietu sieciowego. Niezawodność transportu nie jest odporna na awaryjne połączenia sieciowe lub różne inne problemy komunikacyjne. Niezawodność wiadomości, jak sama nazwa wskazuje, dotyczy niezawodności na poziomie wiadomości, w zależności od tego, ile pakietów jest potrzebnych do dostarczenia wiadomości. Niezawodność wiadomości zapewnia pełną gwarancję dostarczenia i uporządkowania wiadomości, niezależnie od tego, ilu pośredników jest zaangażowanych. Niezawodność wiadomości opiera się na standardzie branżowym dotyczącym

niezawodnej komunikacji opartej na wiadomościach, który utrzymuje sesję na poziomie transportu i wspiera próby w przypadku awarii transportu,

takich jak awaria połączenia bezprzewodowego. Zajmuje się ona automatycznie zatorami, buforowaniem wiadomości i kontrolą przepływu i może