# MittoAgSMS pregled rešenja

Autor Nikola Dukanac

Januar 2018.

Na osnovu zahteva, generisana je webapi2 .net framework 4.5 aplikacija koja omogućuje 4 servisa:

1. slanje sms poruka
2. Filtriranje sms poruka
3. Statistika slanja sms poruka
4. prikaz podataka o zemljama.

Svi projekti i fajlovi unutar Visual Studio solution-a su raspoređeni u okviru nekoliko solution foldera:

**Business Logic** – sloj na kome se implementira poslovna logika

MittoAgSMS.BusinessLogic

MittoAgSMS.BusinessLogic.Abstractions

**DataModel** – sloj na kome se implementira poslovni model, model podataka

MittoAgSMS.BusinessModel

MittoAgSMS.DataModel

**Repositories -** sloj apstrakcije izvora podataka

MittoAgSMS.DataAccessLayer

MittoAgSMS.DataAccessLayer.Abstractions

**Services** -sloj na kome se razvijaju servisi koji koriste repository i druge servise

MittoAgSMS.Services

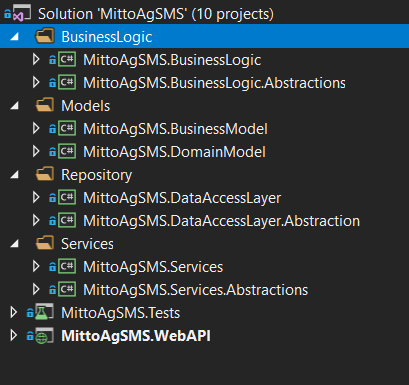
MittoAgSMS.Services.Abstractions

**WebApi** – Sloj na kome se razvijaju kontroleri i Service Api metode.

MittoAgSMS.WebAPI

MittoAgSMS.Tests - mesto gde su unit testovi

Izgled Solution Explorera je prikazan na slici u nastavku.

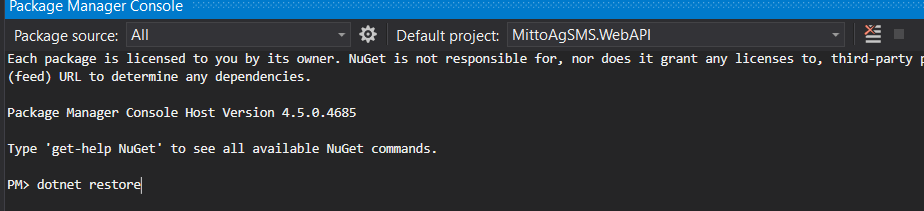


## Preduslovi

Pre pokretanja aplikaije, neophodno je imati NuGet Package Manger i NuGet CLI (Command Line Interface). Njih je moguće instalirati na više načina.

Ukoliko je na računaru omogućen internet saobraćaj ka adresama nuget.org i chocolaty.org, moguće je instalirati NuGet CLI putem alata chocolatey.

Potrebno je pre pokretanja aplikacije uraditi restore Nuget paketa komandom **dotnet restore**



U razvoju su korišćene sledeći Nuget paketi:

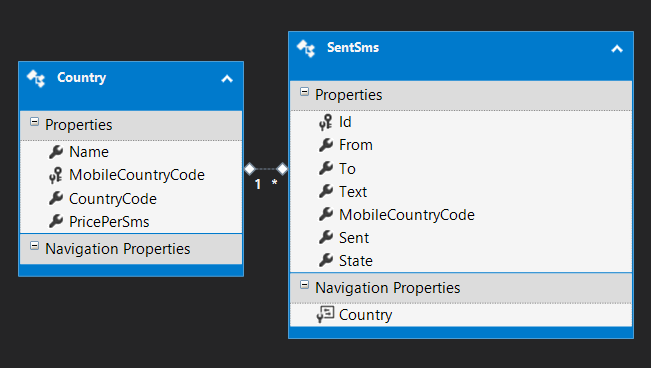
Autofac.Webapi2 - služi za konfigurisanje IoC kontejnera. U Global.asax je registrovan putem metode IocConfig.Configure();

Automapper - služi za mapiranje poslovnog modela u model podataka i obratno. U Global.asax je registrovna putem metode AutoMapperConfig.Initialize();

Serilog i Serilog.Sinks.File - služi za logovanje. U ovom projektu je dodata i uloga Mokovanja slanja sms poruke. To je implementirano u servisu SendSmsServiceMock. Logovanje je efinisano u folderu C:/SMS\_log

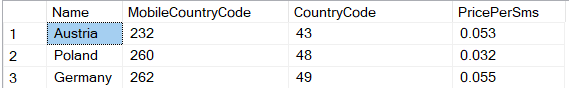
EntityFramework - apstrakovanje baze podataka. U ovom projektu je korišena baza SQL Server 2014.

Model ima 2 tabele a evo i prikaz modela podataka:



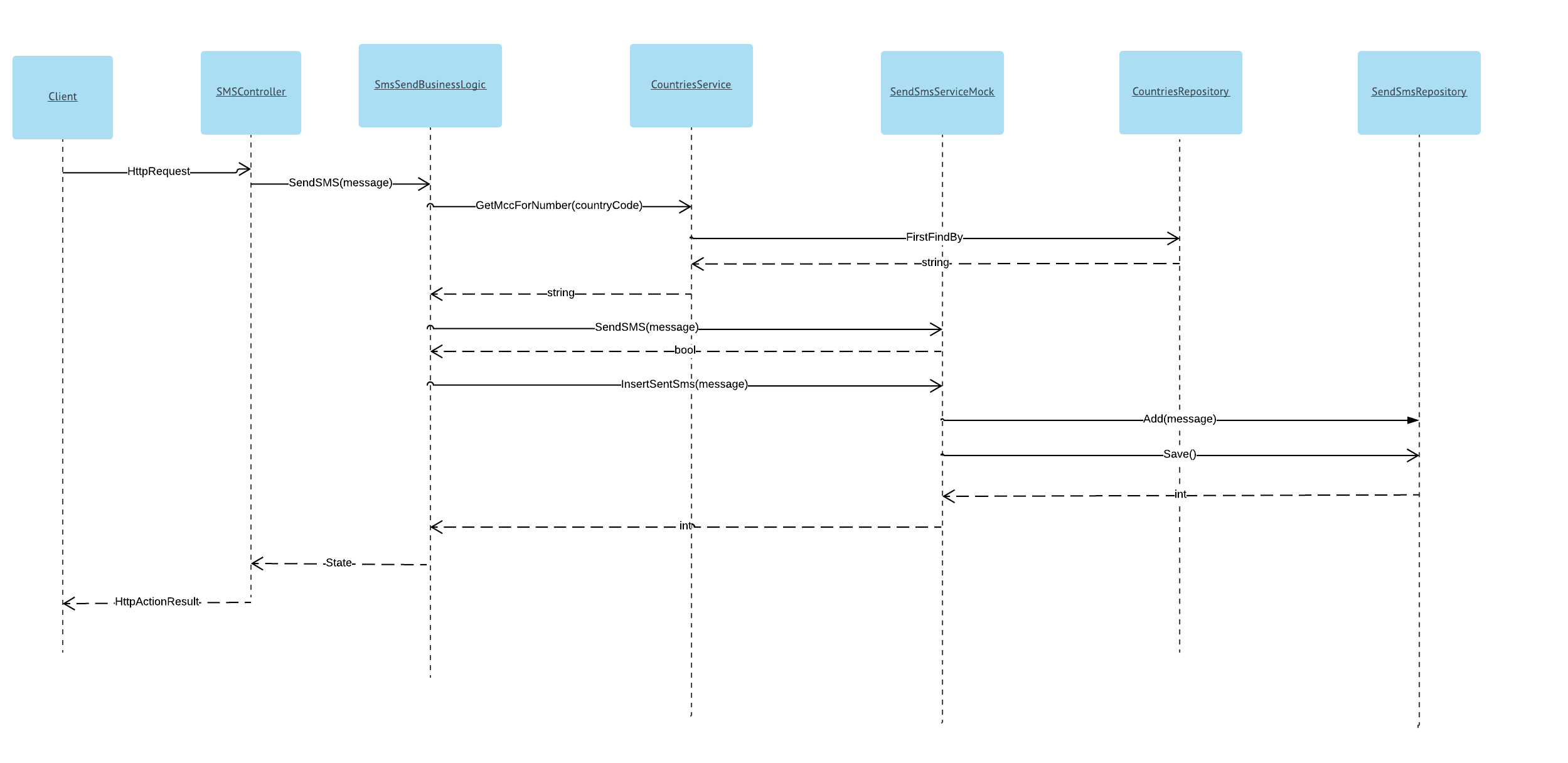
MobileCountryCode je spoljni ključ u tabeli SentSms a primarni u tabeli Country.

Tabela Country inicijalno sadrži sledeće podatke:

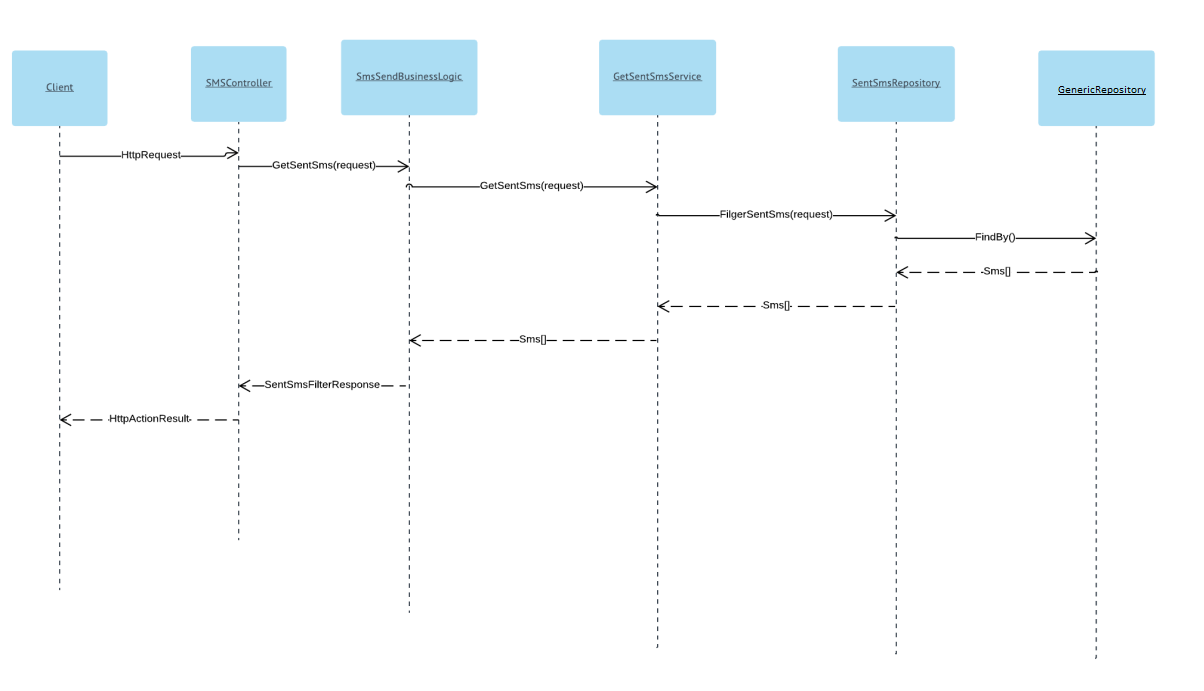


Moq - moking frejmvork koji mokuje servise i njihove pozive i služi za izolovanje klasa u testovima. U projektu Tests se nalaze Unit testovi biznis logike.

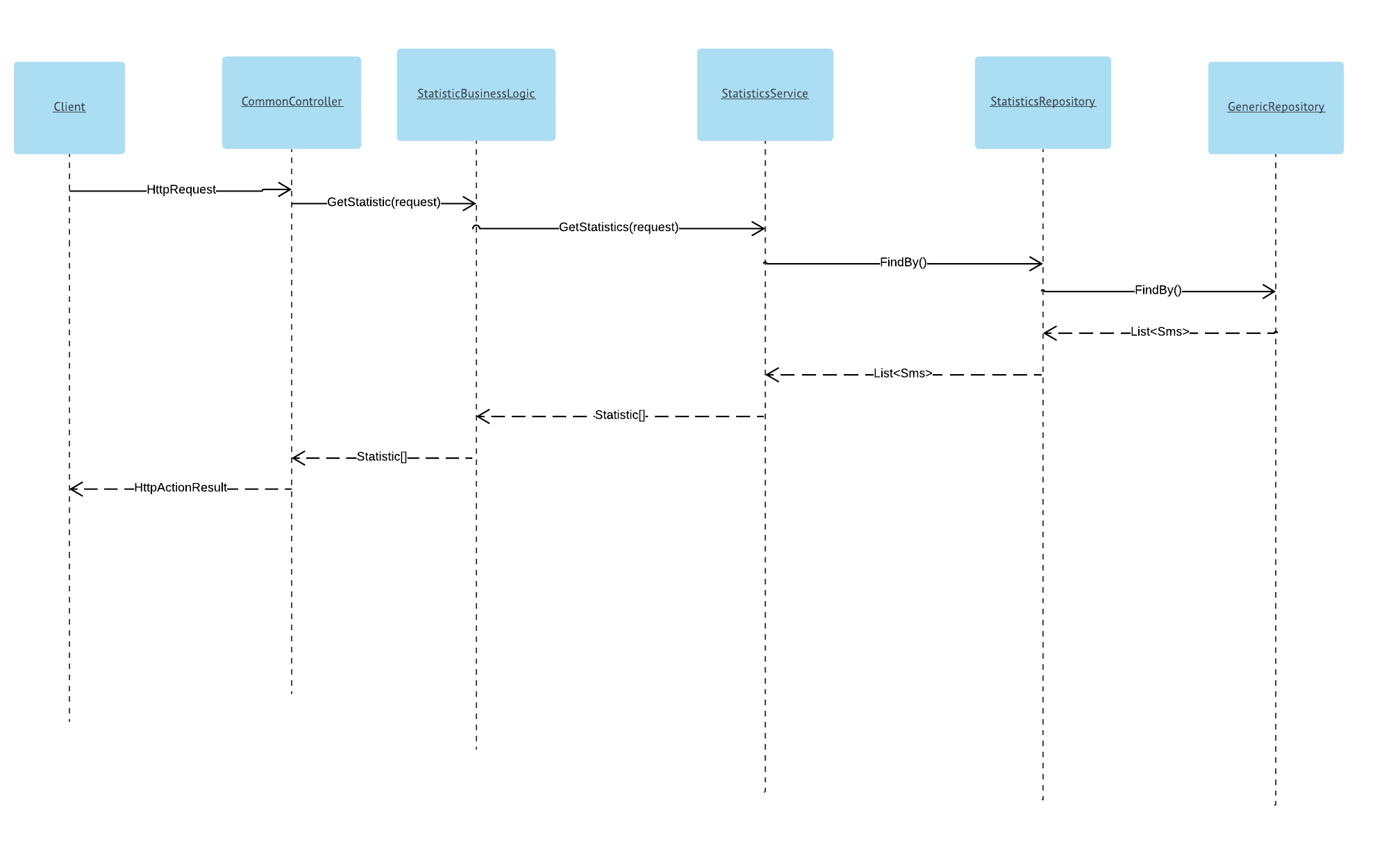
## Sequence dijagrami rešenja



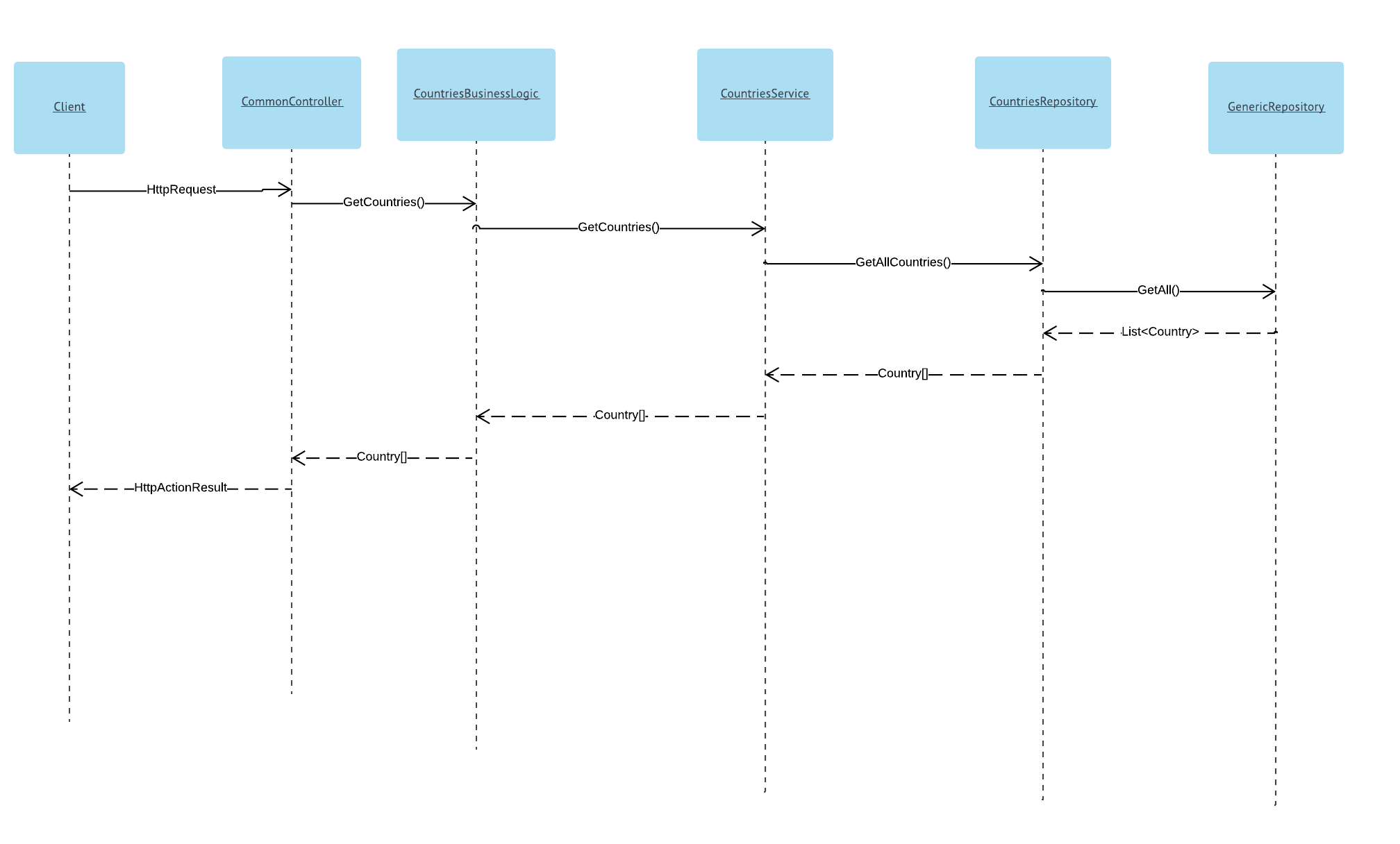
Metod sendSMS za slanje poruka



Metod za filtirranje poruka

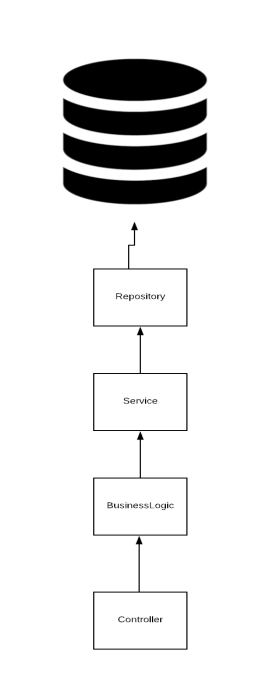


Metod za preuzimanje statistike

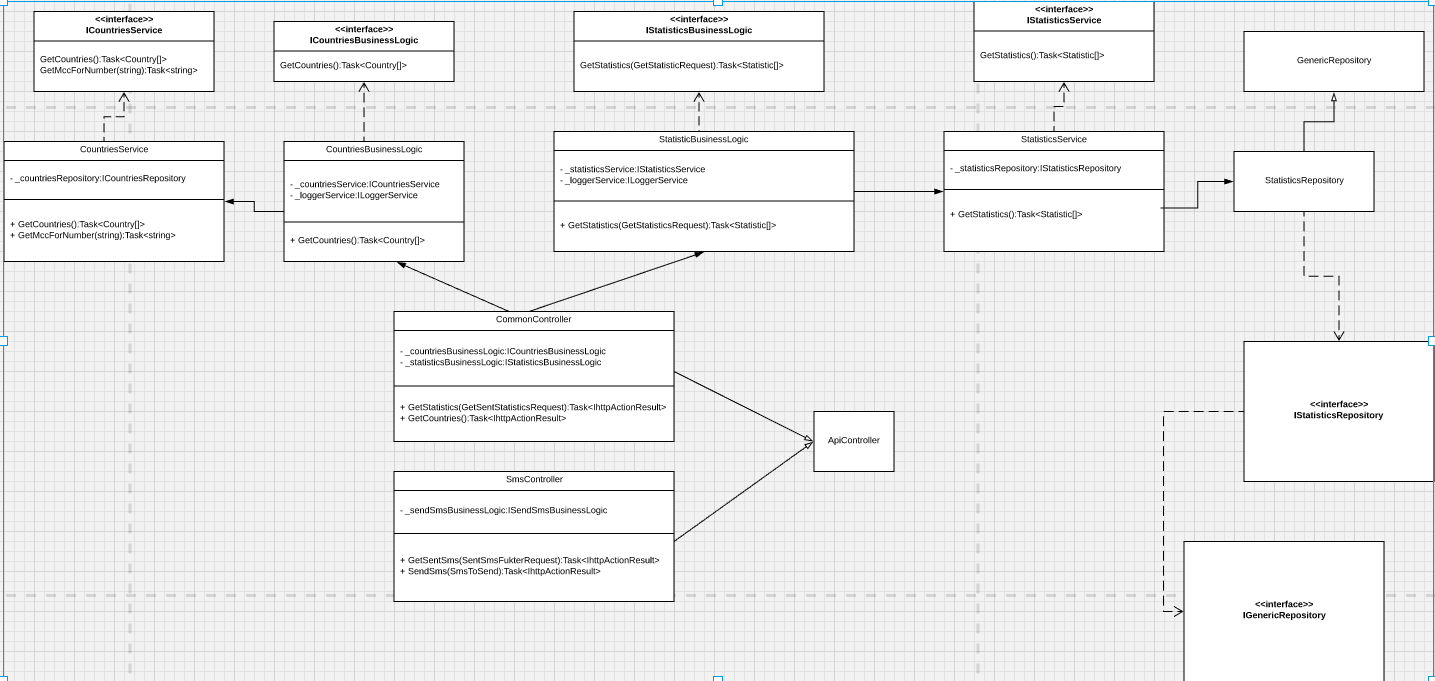


Metod koji vraća sve zemlje

Za implementaciju je korišćena troslojna arhitektura:



## Dijagram klasa



Opis funkcionalnosti

Omogućeno je definisanje formata odgovora u samom queryju - xml ili json. To je urađeno u klasi WebApiConfig.cs

config.Routes.MapHttpRoute(

name: "SMS",

routeTemplate: "{controller}/{action}.{ext}");

config.Routes.MapHttpRoute(

name: "Common",

routeTemplate: "{action}.{ext}",

defaults: new { controller = "Common" });

U istoj klasi je deifnisano i serijalizovanje datuma u requestu da bude UTC

new Newtonsoft.Json.JsonSerializerSettings

{

DateTimeZoneHandling = Newtonsoft.Json.DateTimeZoneHandling.Utc,

DateFormatHandling = Newtonsoft.Json.DateFormatHandling.IsoDateFormat

};

Pošto je ograničeno polje dužine sms poruke 160 karaktera u bazi, implementirano je deljenje poruke na više poruka i slanje istih. To je urađeno putem metode SendSmsBusinessLogic.SplitIntoChunks(). Ova metoda uzima tekst poruke i stringova dužine 160 karaktera ili kraće. Kompleksnost algoritma je O(n).

## Dodatna unapređenja

Za verziju 2 je potrebno dodati nove module:

**Caching** - modul za keširanje

**Auth** - modul za autentifikaciju i autorizaciju

**HealthCheck** - modul za proveru zdravlja servisa

**Swagger** - alat za generisanje dokumentacije