Pogo Rent 3000

# Background

This is a the first of several articles about building robust GIS applications build on .Net and Bing Maps.

The aim of this first article is to describe the implementation of Pogo Rent 3000. The idea of this project is to provide a solid implementation of a GIS project based on Bing Maps. The techniques and framework used are well known and provide a robust foundation for GIS applications.

We strongly believe that using standard formats for communication with GIS backend is something that is of importance and cannot be stated enough.

# Application structure

The application is structured the simplest way possible with providing a robust platform to extend the system. We build a classic 3-tier application.

## UI Tier

On top we have Bing Maps V8, it is one of the best map widget out there as today. It provides a straight forward, documented API. More over a set of spatial function [*list some]* and a *n* modules. A module is functionality you can add on to the web client. For example, GeoJSON support. You can also write your own modules. *[Another article?]*

## Service Tier

As service tier we have a Asp.NET Web API. This tier is boosted with Net Topology Suit which provide a rich set of GIS operations. It also provide GeoJSON support. With this in place we have the possibility to provide OGC formats.

## Data Tier

The database is a SQL Server where we use the spatial datatypes SQL Server provides. Using the spatial formats in SQL Server gives us better quality (i e spatial constraints), ability to spatial filter etc. We will go into SQL Server and spatial in another article deeper.

# Wrap up

With the techniques and framework in place, it is pretty straight forward to implement. The implementation is as simple as possible and it provides possibility to extend the PogoRent3000 system.