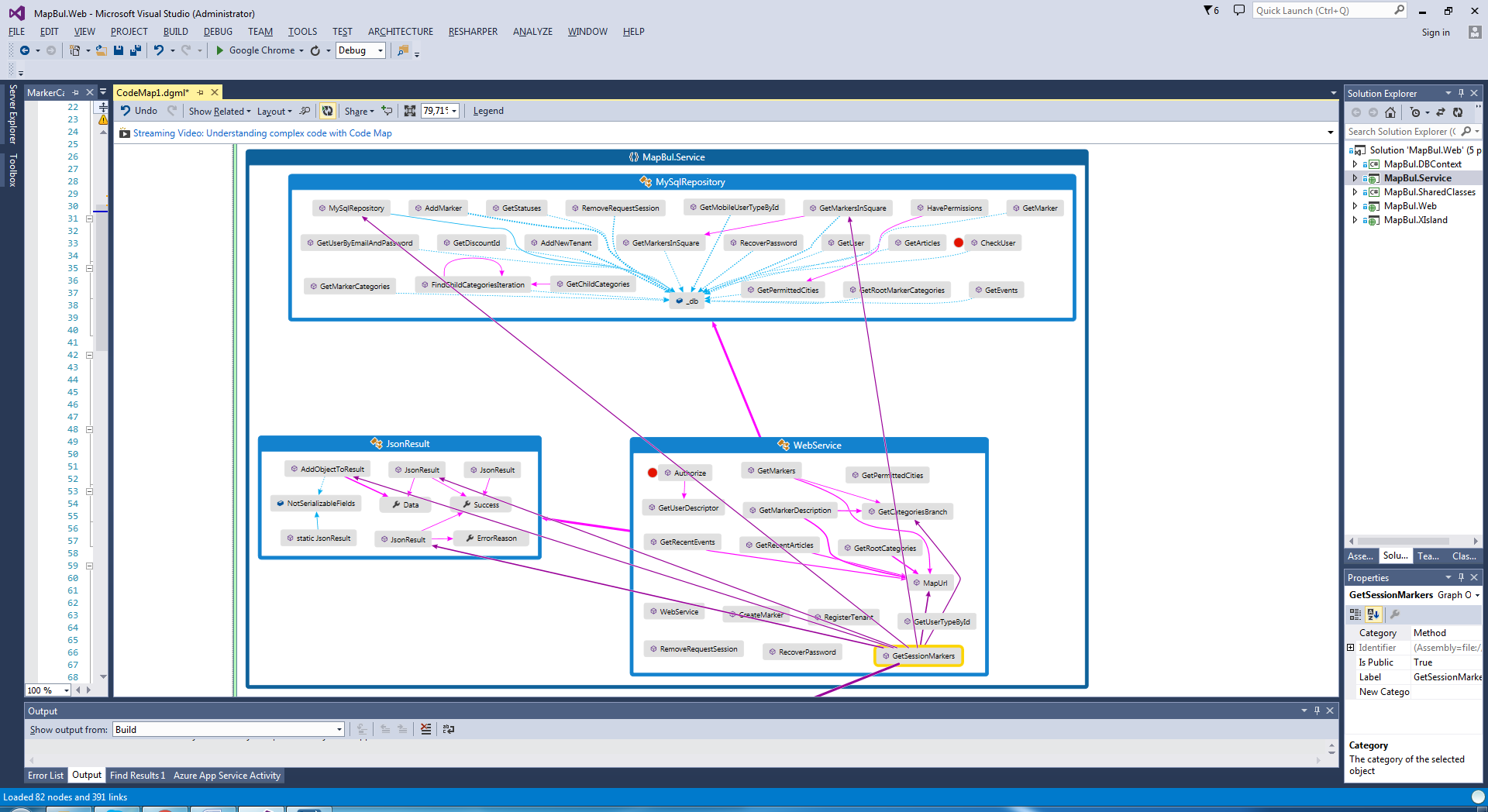
# *WEB*-СЛУЖБА

## Общая структура проекта



Сборка реализует функционал веб-службы для мобильного приложения. Включает 3 класса, основной класс *WebService* описывает логику методов, вызываемых мобильным приложением. Остальные классы являются вспомогательными.

Зависимости сборки:

* *DataBase*—библиотека типов для работы с базой данных приложения.
* *EntityFramework*—зависимость библиотеки *DataBase.*
* *MySql*.*Data*.*Entity*.*EF6*—зависимость библиотеки *DataBase.*
* *Newtonsoft*.*Json—*библиотека сериализации объектов в JSON и десериализации из него.
* *System*—библиотека стандартных типов.

Целевой файл сборки—*MapBul.Service.dll*, публикуемый на *IIS*. Сборка не подписана.

### Взаимодействие c мобильным приложением

Взаимодействие приложения и сервера построено с помощью *SOAP* веб-сервиса.

Данные от приложения передаются в соответствующие поля метода.

Данные от сервера представлены в формате *JSON* следующей структурой:

{

“*success*”:”<*true*/*false*>”,

“*errorReason*”:”<текст сообщения об ошибке, если она есть>”,

“*data*”:[{“<название поля>”:”<значение поля>”},{}…..]

}

## Описание объектов

### Класс *WebService*

В данном классе реализована большая часть алгоритмов взаимодействия и обработки информации. Методы данного класса вызываются мобильным приложением. Класс наследуется от *System.Web.Services.WebService*. Класс имеет следующие *Private* методы:

* + *private List<int> GetCategoriesBranch(category category)* — Метод возвращает список ИД категорий от данной до корневой. Входной параметр—категория, для которой необходимо вернуть ветку категорий.
  + *private string MapUrl(string filePath)—*Метод возвращает полный URL до файла. Входной параметр—относительный путь к файлу в том виде, в котором он лежит в БД.
  + *private object GetUserDescriptor(user user)—*Метод возвращает описатель пользователя в зависимости от его типа. Входной параметр—объект класса *User*, для которого необходим описатель.

*Public* методы реализуют обмен информацией с мобильным приложением. В классе описаны следующие *Public* методы:

* + *public string Authorize(string email, string password) —* Метод проверки адреса электронной почты и пароля пользователя. Входные параметры: адрес электронной почты и пароль соответственно. Адрес обращения: <http://185.76.145.214/Service/webservice.asmx?op=Authorize>
  + *public string GetUserTypeById(int id) —* Метод возвращает тип пользователя в виде строки по его ID.Входной параметр—идентификатор пользователя. Адрес обращения: [http://185.76.145.214/Service/webservice.asmx?op=GetUserTypeById](http://185.76.145.214/Service/webservice.asmx?op=GetUserTypeById%20)
  + *public string GetMarkers(double p1Lat, double p1Lng, double p2Lat, double p2Lng) —* Метод возвращает список маркеров в указанном прямоугольнике. Входные параметры являются координатами левого верхнего и правого нижнего угла прямоугольника. Адрес обращения: <http://185.76.145.214/Service/webservice.asmx?op=GetMarkers>
  + *public string GetSessionMarkers(double p1Lat, double p1Lng, double p2Lat, double p2Lng,string sessionId) —* Метод возвращает список непереданных в данной сессии маркеров в указанном прямоугольнике.Входные параметры—координаты прямоугольника и идентификатор сессии. Адрес обращения: <http://185.76.145.214/Service/webservice.asmx?op=GetSessionMarkers>
  + *public string RemoveRequestSession(string sessionId) —* метод удаляет данные сессии. Вызывается в начале новой загрузки маркеров. Входной параметр – идентификатор сессии. Адрес обращения: [http://185.76.145.214/Service/webservice.asmx?op=RemoveRequestSession](http://185.76.145.214/Service/webservice.asmx?op=Authorize)
  + *public string GetMarkerDescription(int markerId) —* Метод возвращает подробное описание маркера. Входной параметр – идентификатор маркера. Адрес обращения: [http://185.76.145.214/Service/webservice.asmx?op=GetMarkerDescription](http://185.76.145.214/Service/webservice.asmx?op=Authorize)
  + *public string GetRootCategories()* — Метод возвращает список корневых категорий маркеров. Адрес обращения: [http://185.76.145.214/Service/webservice.asmx?op=GetRootCategories](http://185.76.145.214/Service/webservice.asmx?op=Authorize)
  + *public string GetRecentArticles(bool refresh = false, DateTime? existingDateTime = null) —* Метод загрузки новых статей в мобильное приложение. Параметр *refresh* показывает необходимость загрузки только что добавленных статей. При наличии параметра *existingDateTime* загружаются 15 статей старше указанной даты. Адрес обращения: [http://185.76.145.214/Service/webservice.asmx?op=GetRecentArticles](http://185.76.145.214/Service/webservice.asmx?op=Authorize)
  + *public string GetRecentEvents(bool refresh = false, DateTime? existingDateTime = null) —* Метод загрузки новых событий в мобильное приложение. Параметр *refresh* показывает необходимость загрузки только что добавленных событий. При наличии параметра *existingDateTime* загружаются 15 событий старше указанной даты. Адрес обращения: [http://185.76.145.214/Service/webservice.asmx?op=GetRecentEvents](http://185.76.145.214/Service/webservice.asmx?op=Authorize)
  + *public string CreateMarker(…)—*метод добавления нового маркера. Входные параметры представляют собой описание маркера. Адрес обращения: [http://185.76.145.214/Service/webservice.asmx?op= CreateMarker](http://185.76.145.214/Service/webservice.asmx?op=%20CreateMarker%20)
  + *public string GetPermittedCities(string userGuid)—* Метод возвращает список городов, на которые у указанного пользователя есть права. В метод передается идентификатор пользователя. Адрес обращения: [http://185.76.145.214/Service/webservice.asmx?op=GetPermittedCities](http://185.76.145.214/Service/webservice.asmx?op=GetPermittedCities%20)
  + *public string RegisterTenant(…)—*Метод регистрации нового жителя.Адрес обращения: [http://185.76.145.214/Service/webservice.asmx?op=RegisterTenant](http://185.76.145.214/Service/webservice.asmx?op=Authorize)
  + *public string RecoverPassword(string email) —* Метод восстановления пароля. Новый пароль высылается на почту. Входной параметр—адрес электронной почты. Адрес обращения: [http://185.76.145.214/Service/webservice.asmx?op=RecoverPassword](http://185.76.145.214/Service/webservice.asmx?op=RecoverPassword%20)

### Класс *MySqlRepository*

Данный класс реализует *CRUD* методы взаимодействия с базой данных. Методы данного класса вызываются при необходимости из класса *WebService.*

### Класс *JsonResult*

Данный класс является контейнером для данных ответа сервера приложению. Класс написан для удобства сериализации данных ответа и для типизации ответов. Поля и методы класса:

* + *success*—Флаг успешного выполнения запроса.
  + *errorReason*—строка причины ошибки.
  + *Data*—коллекция данных результата.
  + Конструкторы по данным для отсылки и причине ошибки.
  + *NotSerializableFields*—список названий полей, которые не должны передаваться приложению.
  + *AddObjectToResult(object o, int index)*—метод добавляет поля параметра-объекта в поле *Data*. Используется для удобства формирования ответа.