**ООО «Системы МАСП»**

**Back-end часть проекта MapBul**

**ОПИСАНИЕ СТРУКТУРЫ**

Исполнитель: Крючков А. И.

2016 г.

[ОБЩЕЕ ОПИСАНИЕ 5](#_Toc458005027)

[БИБЛИОТЕКА БАЗЫ ДАННЫХ 6](#_Toc458005028)

[Общая структура проекта 6](#_Toc458005029)

[Описание объектов 7](#_Toc458005030)

[*WEB*-СЛУЖБА 11](#_Toc458005031)

[Общая структура проекта 11](#_Toc458005032)

[Взаимодействие c мобильным приложением 12](#_Toc458005033)

[Описание объектов 13](#_Toc458005034)

[Класс *WebService* 13](#_Toc458005035)

[Класс *MySqlRepository* 16](#_Toc458005036)

[Класс *JsonResult* 16](#_Toc458005037)

[*WEB*—ИНТЕРФЕЙС 18](#_Toc458005038)

[Общая структура проекта 18](#_Toc458005039)

[*MapBul.Web.Controllers* 19](#_Toc458005040)

[*MapBul.Web.Models* 20](#_Toc458005041)

[*MapBul.Web.Auth* 21](#_Toc458005042)

[*MapBul.Web.Helpers* 22](#_Toc458005043)

[*MapBul.Web.Repository* 23](#_Toc458005044)

[*MapBul.Web.ExternalRequest* 23](#_Toc458005045)

[Свойства и зависимости сборки 24](#_Toc458005046)

[Описание объектов 25](#_Toc458005047)

[Пространство имен *MapBul.Web.Controllers* 25](#_Toc458005048)

[Пространство имен *MapBul.Web.Models* 26](#_Toc458005049)

[Пространство имен *MapBul.Web.Auth* 27](#_Toc458005050)

[Пространство имен *MapBul.Web.Repository* 28](#_Toc458005051)

[Пространство имен *MapBul.Web.ExternalRequest* 28](#_Toc458005052)

[БИБЛИОТЕКА СОВМЕСТНО ИСПОЛЬЗУЕМЫХ ТИПОВ *MapBul.SharedClasses* 30](#_Toc458005053)

[Структура проекта 30](#_Toc458005054)

[Описание объектов 31](#_Toc458005055)

[Пространство имен *MapBul.SharedClasses* 31](#_Toc458005056)

[Пространство имен *MapBul.SharedClasses.Constants* 31](#_Toc458005057)

[*WEB*-ПРИЛОЖЕНИЕ *MapBul.XIsland* 33](#_Toc458005058)

[Структура проекта 33](#_Toc458005059)

[*MapBul.XIsland.Controllers* 34](#_Toc458005060)

[*MapBul.XIsland.Models* 35](#_Toc458005061)

[*MapBul.XIsland.Helpers* 36](#_Toc458005062)

[*MapBul.Web.Repository* 37](#_Toc458005063)

[Свойства и зависимости сборки 38](#_Toc458005064)

[Описание объектов 39](#_Toc458005065)

[Пространство имен *MapBul.XIsland.Controllers* 39](#_Toc458005066)

[Пространство имен *MapBul. XIsland.Models* 39](#_Toc458005067)

[Пространство имен *MapBul. XIsland.Repository* 40](#_Toc458005068)

[Пространство имен *MapBul.XIsland.Helpers* 40](#_Toc458005069)

[ЗАПУСК ПРИЛОЖЕНИЙ 41](#_Toc458005070)

[ПРОГРАММНЫЕ СРЕДСТВА 41](#_Toc458005071)

[РЕКВИЗИТЫ ДОСТУПА 42](#_Toc458005072)

[*RDP* 42](#_Toc458005073)

[*FTP* 42](#_Toc458005074)

[*MySQL* 42](#_Toc458005075)

# ОБЩЕЕ ОПИСАНИЕ

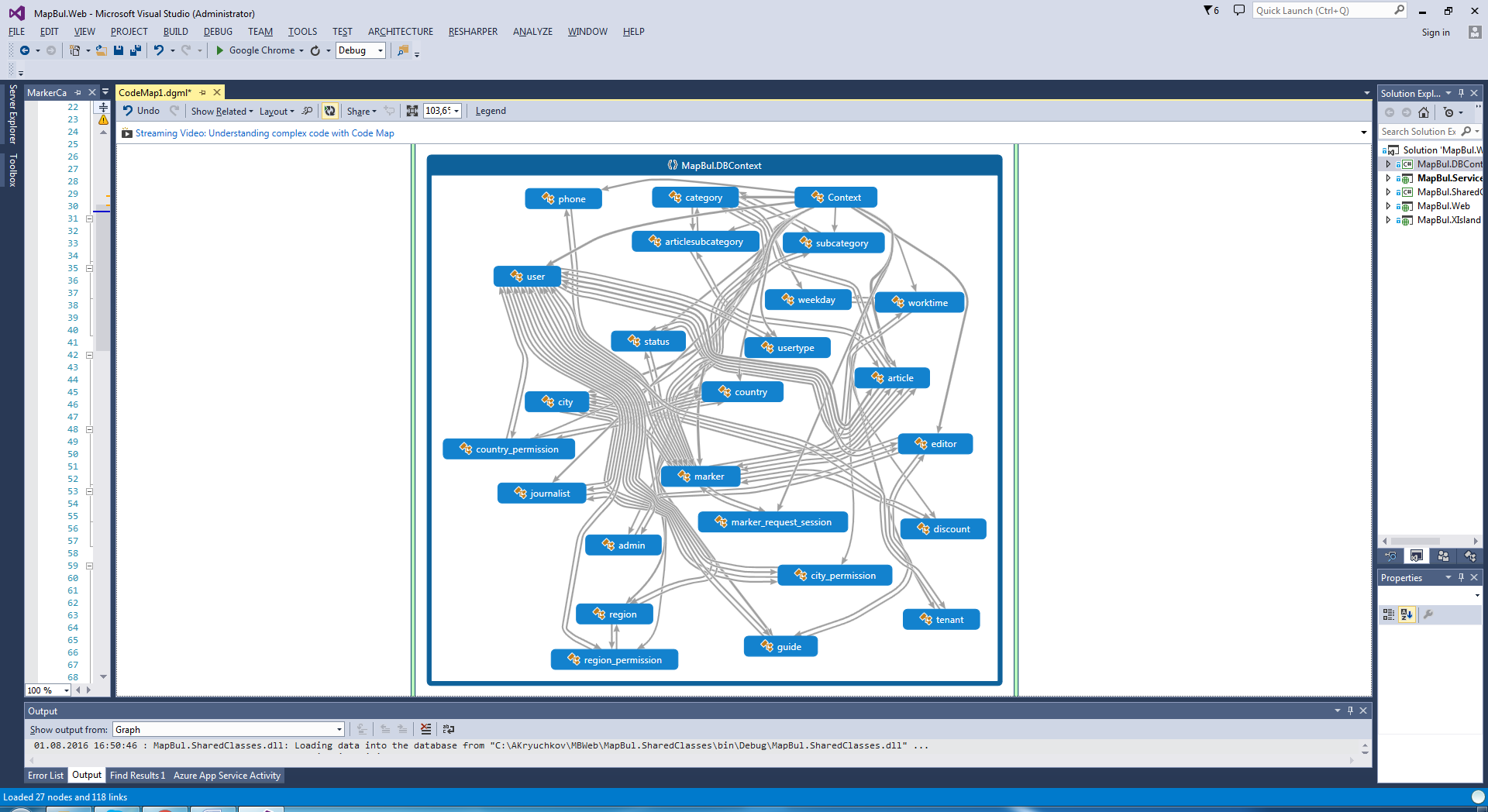
Серверная часть приложения содержит в себе пять отдельных проектов:

* Библиотека, предоставляющая методы для работы с базой данных (*MapBul*.*DBContext*)
* Проект *web*-службы, осуществляющей взаимодействие с клиентским приложением (*MapBul.Service*)
* Проект *web*-приложения, предоставляющего доступ к кабинету администратора, журналиста, редактора и гида (*MapBul.Web*)
* Библиотека совместно используемых классов (*MapBul.SharedClasses*)
* Web-приложение страницы сервиса *XIsland* (*MapBul.XIsland*)

# БИБЛИОТЕКА БАЗЫ ДАННЫХ

## Общая структура проекта

Сборка включает в себя классы-модели для таблиц базы данных. Модели и контекст базы данных сгенерированы с помощью *EntityFramework Code First.*



Зависимости сборки:

* *EntityFramework*—объектно-ориентированная технология доступа к данным. Производит сопоставление реляционных данных с объектами.
* *MySql*.*Data*.*Entity*.*EF6*—расширение *EntityFramework* для СУБД *MySQL*.
* *System*—библиотека стандартных типов.

Целевой файл сборки— *MapBul*.*DBContext.dll*, подключаемый в качестве зависимости в проекты *MapBul.Service, MapBul.XIsland* и *MapBul.Web*. Сборка не подписана, развертывание закрытое.

## Описание объектов

Проект *MapBul.DBContext* содержит модели для всех таблиц базы данных приложения и контекст базы данных.

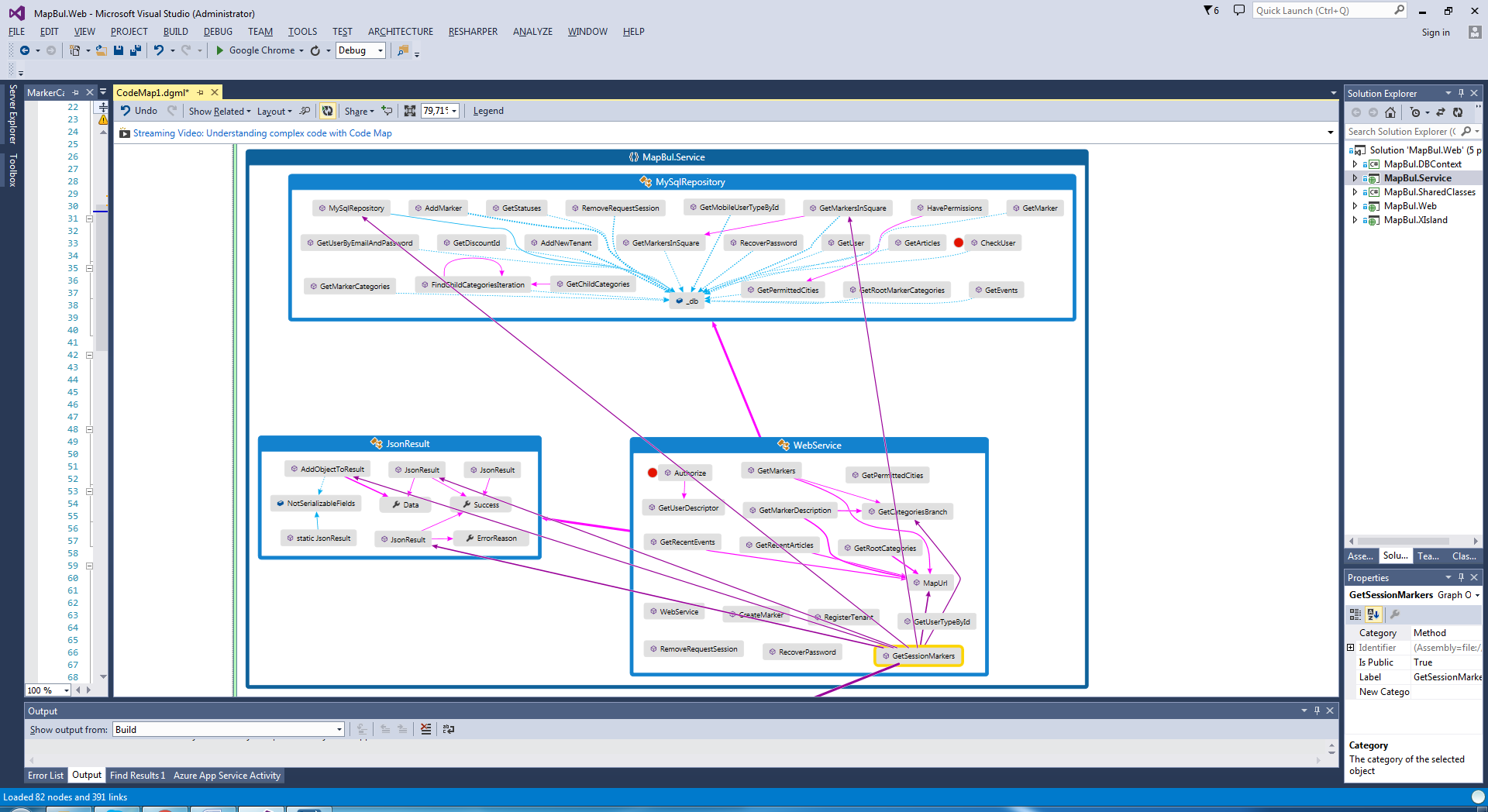
Набор моделей следующий:

* *Admin*—модель одноименной таблицы базы данных, содержит список администраторов. Внешний ключ на таблицу *User.* Содержит поле, показывающее, является ли администратор суперадминистратором.
* *Article*—модель одноименной таблицы базы данных, содержит список статей. Каждая статья привязана к записи в таблице *User*, *City*, *Status*, *Marker* и *Category.*
* *Articlesubcategory*—Модель одноименной таблицы базы данных. Таблица для развязки отношения многие ко многим между таблицами *Articles* и *Category.*
* *Category*—модель одноименной таблицы базы данных, содержит список категорий. Категории образуют древовидную структуру любой степени вложенности. В таблице хранится два отдельных дерева категорий: для маркеров и для статей.
* *City*—модель одноименной таблицы базы данных, содержит список городов. Для каждого города хранится идентификатор места от *Google Maps API*. Каждый город привязан к стране.
* *City\_permission*—модель одноименной таблицы базы данных. Таблица необходима для выставления прав для пользователей на деятельность в определенном городе. Содержит внешние ключи на таблицу городов и пользователей (*User*).
* *Country*—модель одноименной таблицы базы данных. Таблица содержит список стран. Для каждой страны хранится идентификатор места от *Google Maps API*.
* *Country\_permission*—модель одноименной таблицы базы данных. Таблица необходима для выставления прав для пользователей на деятельность в определенной стране. Содержит внешние ключи на таблицу стран и пользователей (*User*).
* *Discount*—модель одноименной таблицы базы данных, Таблица содержит словарь возможных скидок. На записи в данной таблице ссылается таблица маркеров.
* *Editor*—модель одноименной таблицы базы данных. Таблица содержит описатели редакторов. Каждая запись таблицы ссылается на запись в таблице пользователей.
* *Guide*— модель одноименной таблицы базы данных. Таблица содержит описатели гидов. Каждая запись таблицы ссылается на запись в таблице пользователей.
* *Journalist*— модель одноименной таблицы базы данных. Таблица содержит описатели журналистов. Каждая запись таблицы ссылается на запись в таблице пользователей.
* *Marker*—модель одноименной таблицы базы данных. Содержит по одной записи для каждого маркера. Каждая запись привязана к записи в таблице *User*, *City*, *Status*, *Marker* и *Category.*
* *Marker\_request\_session*—модель одноименной таблицы базы данных. Таблица содержит идентификатор сессии, и ссылку на таблицу маркеров. Таблица используется для сокращения количества передаваемых маркеров с сервера на устройство. Для каждой сессии для каждого запрошенного маркера создаются записи, которые указывают, какие маркеры уже были переданы приложению. По окончании сессии записи с данным идентификатором удаляются.
* *Phone*—модель одноименной таблицы базы данных. Таблица содержит список телефонов для маркеров. У одного маркера может быть несколько телефонов. Основной телефон помечается «1» в поле *Primary*.
* *Region*—модель одноименной таблицы базы данных. На данный момент таблица не используется.
* *Region\_permission*—модель одноименной таблицы базы данных. На данный момент таблица не используется.
* *Status*—модель одноименной таблицы базы данных. В данной таблице хранятся все возможные статусы для маркеров и статей.
* *Subcategory*— Модель одноименной таблицы базы данных. Таблица для развязки отношения многие ко многим между таблицами *Marker* и *Category.*
* *Tenant*— модель одноименной таблицы базы данных. Таблица содержит описатели жителей. Каждая запись таблицы ссылается на запись в таблице пользователей.
* *User*—модель одноименной таблицы базы данных. Представляет собой абстрактный тип данных, содержащий общие поля для всех типов пользователей. Для каждого пользователя создается запись в данной таблице.
* *Userype*—модель одноименной таблицы базы данных. Таблица содержит все типы пользователей в системе.
* *Weekday*—модель одноименной таблицы базы данных. Таблица представляет собой справочник дней недели.
* *WorkTime*—модель одноименной таблицы базы данных. В таблице содержатся времена работы маркеров.

Контекст базы данных представлен в классе *Context*. Класс наследуется от *DbContext.* Свойства класса представляют собой коллекции объектов классов-моделей.

# *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.*

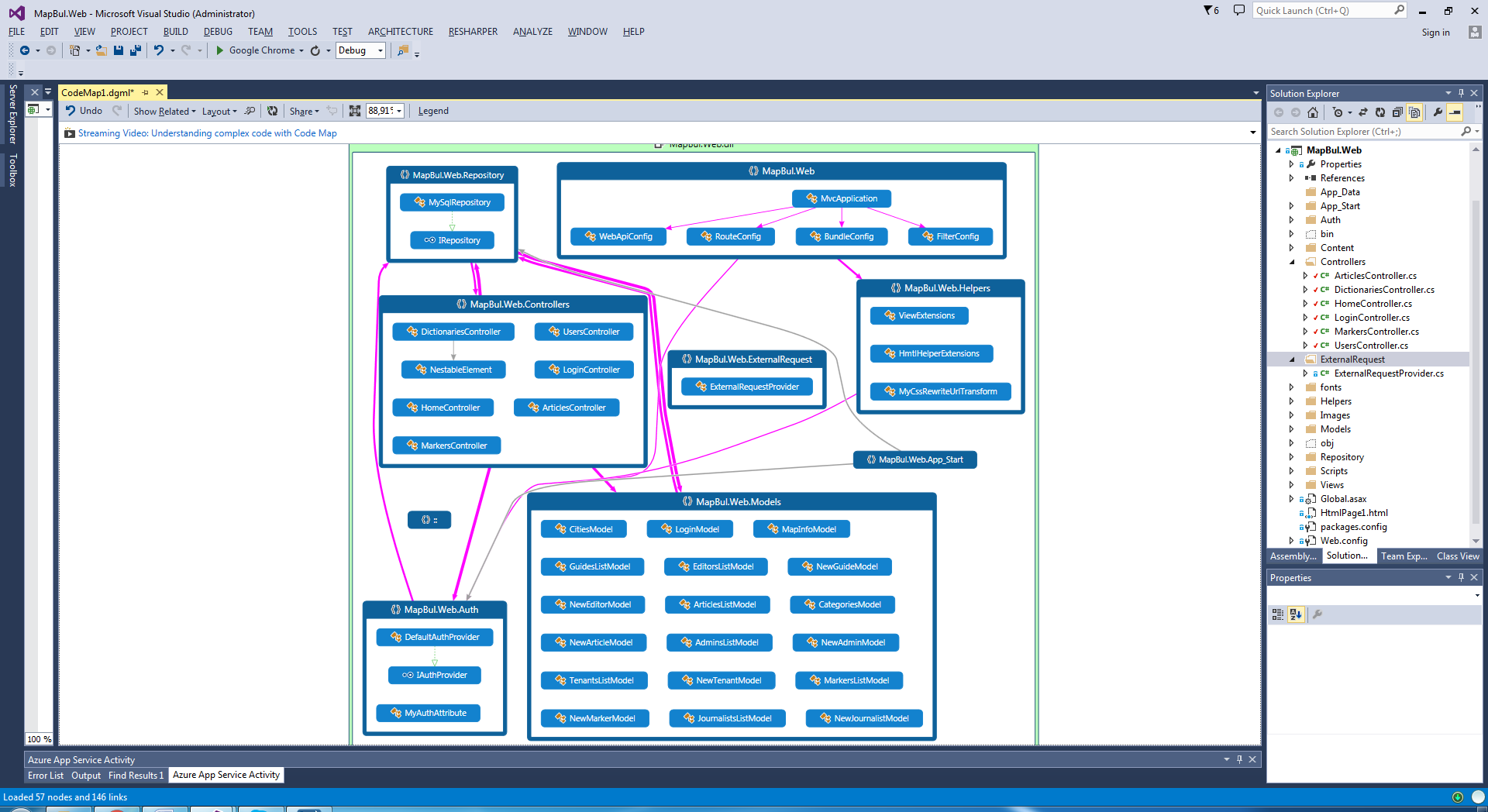
### Класс *JsonResult*

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

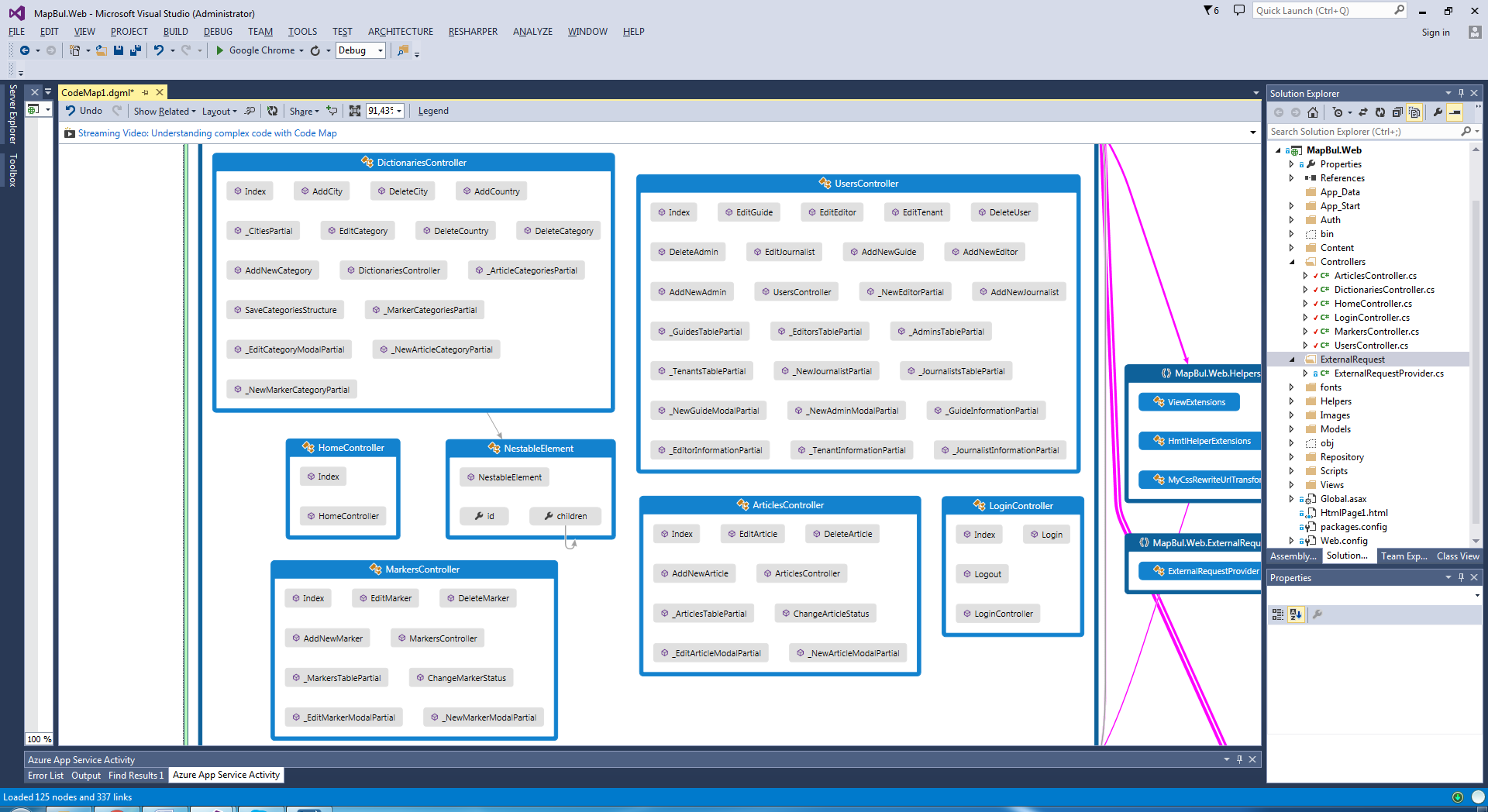
* + *success*—Флаг успешного выполнения запроса.
  + *errorReason*—строка причины ошибки.
  + *Data*—коллекция данных результата.
  + Конструкторы по данным для отсылки и причине ошибки.
  + *NotSerializableFields*—список названий полей, которые не должны передаваться приложению.
  + *AddObjectToResult(object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

# *WEB*—ИНТЕРФЕЙС

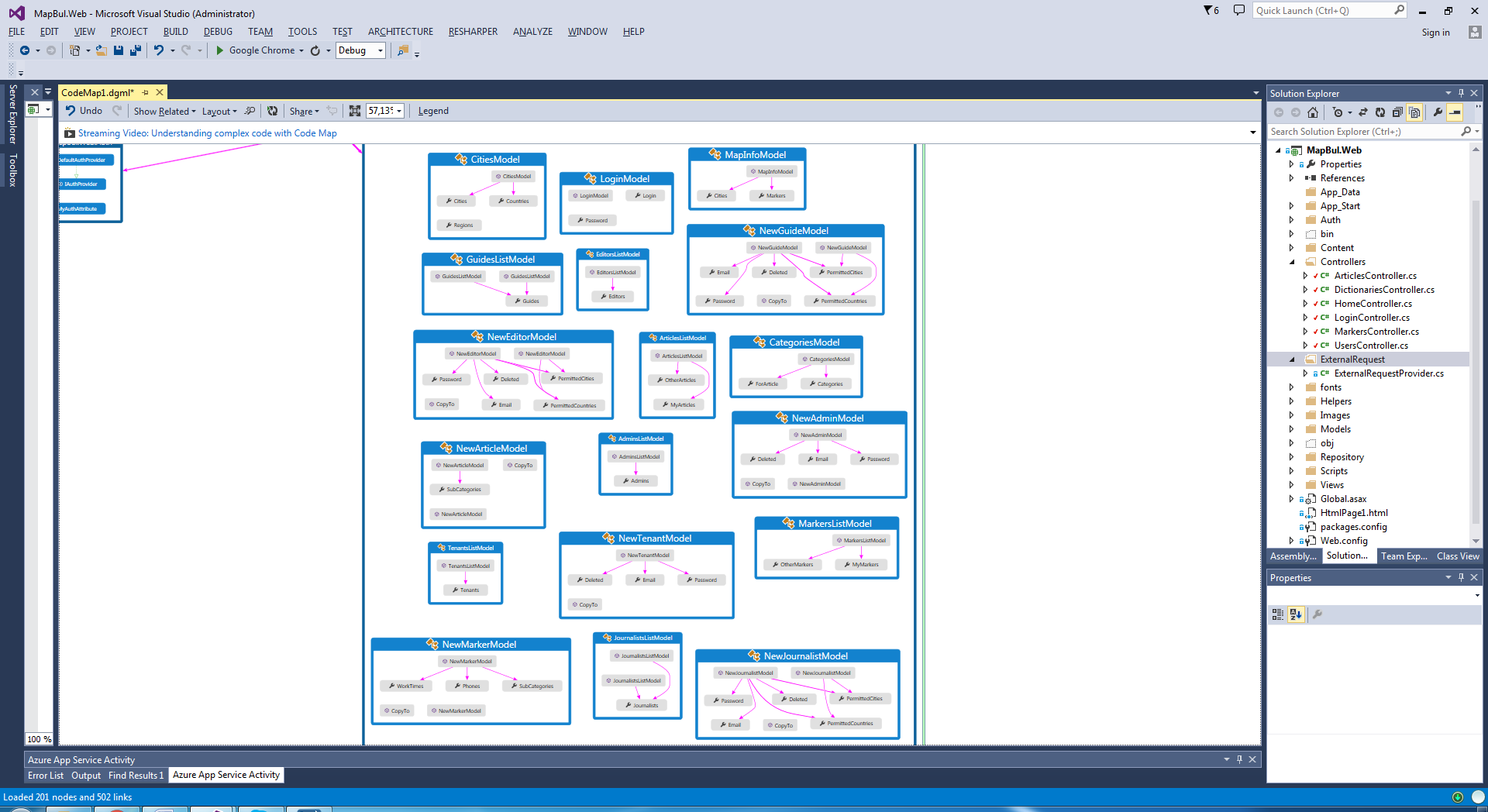
## Общая структура проекта



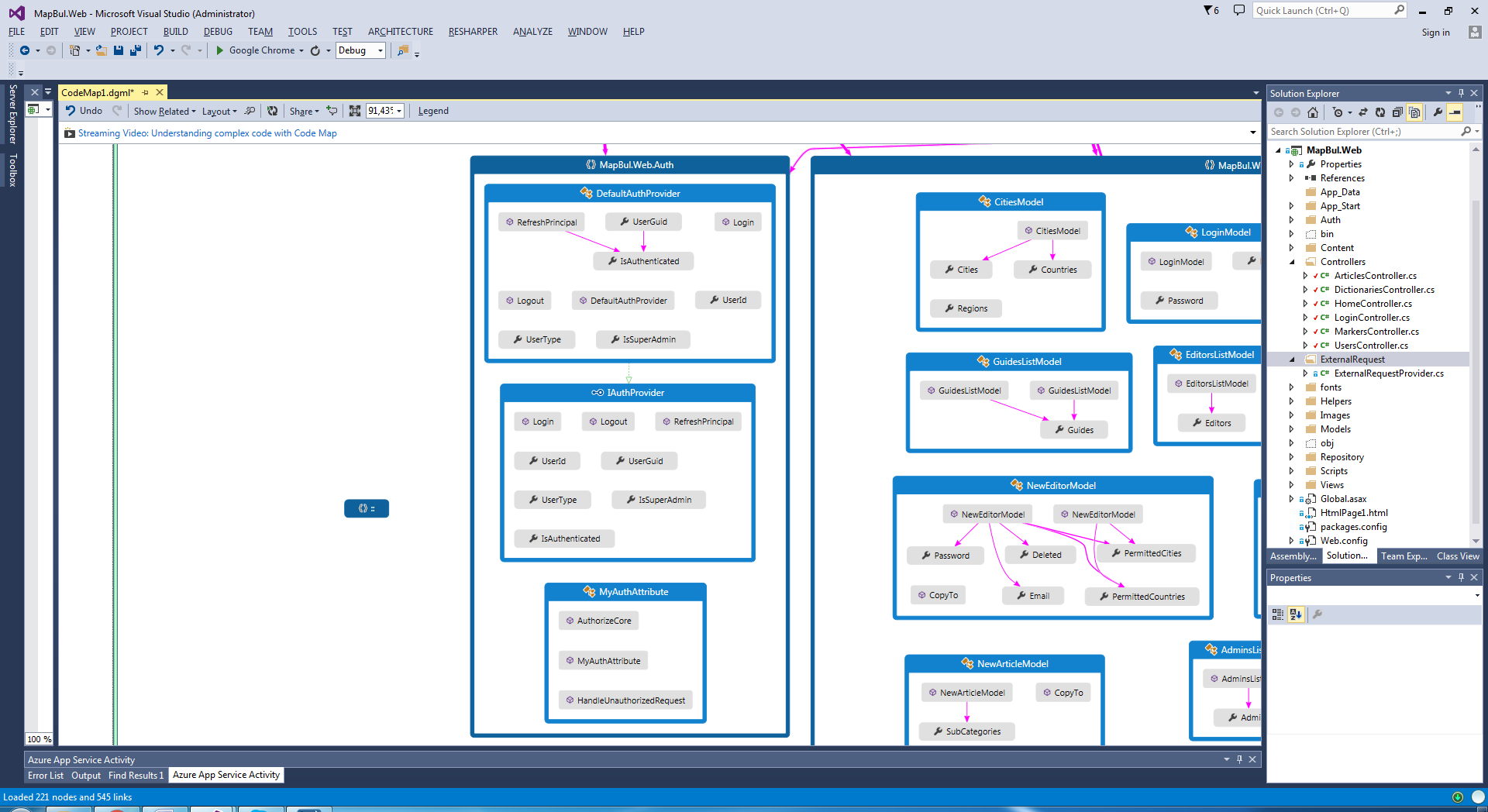
### *MapBul.Web.Controllers*



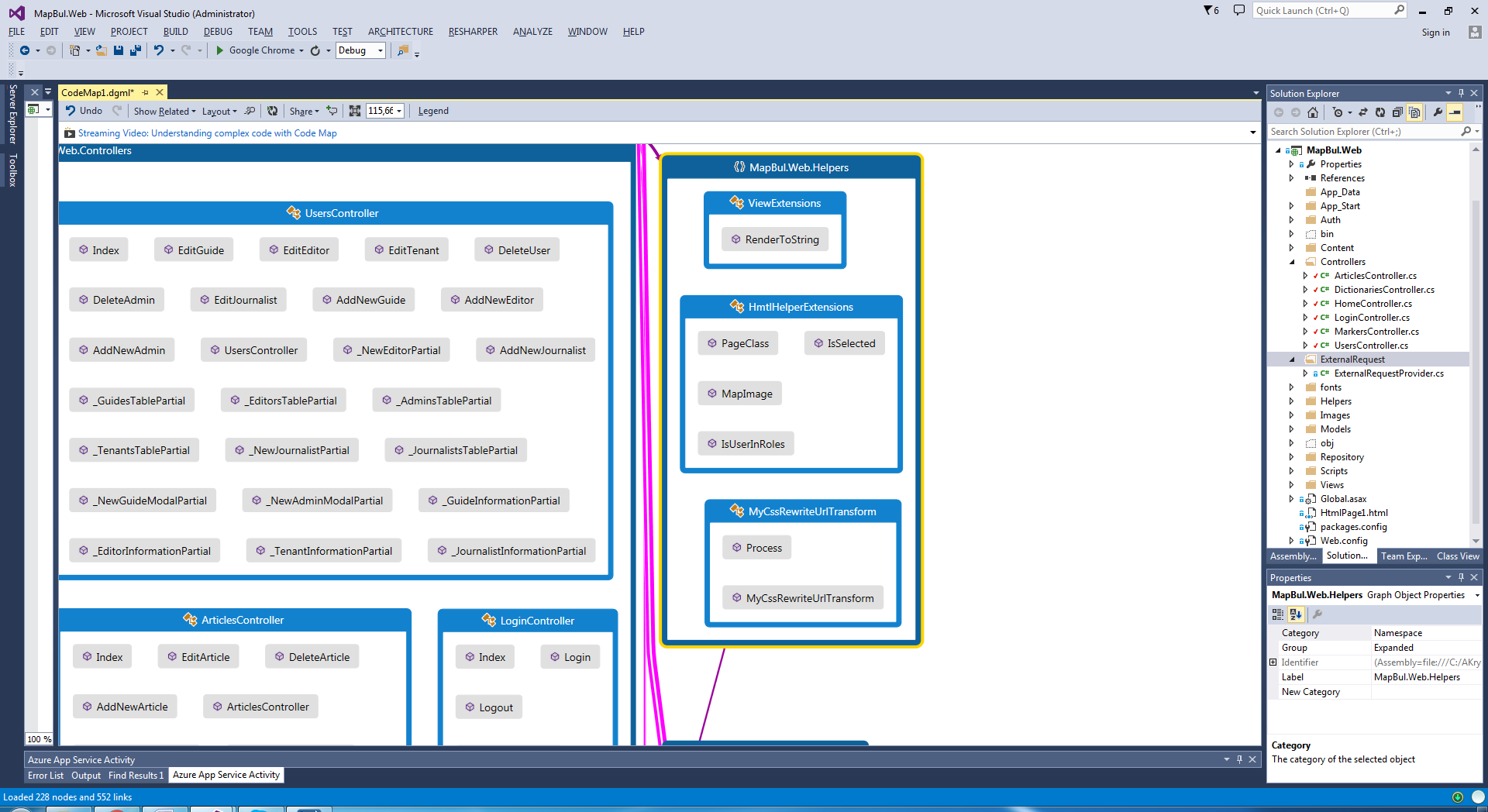
### *MapBul.Web.Models*



### *MapBul.Web.Auth*



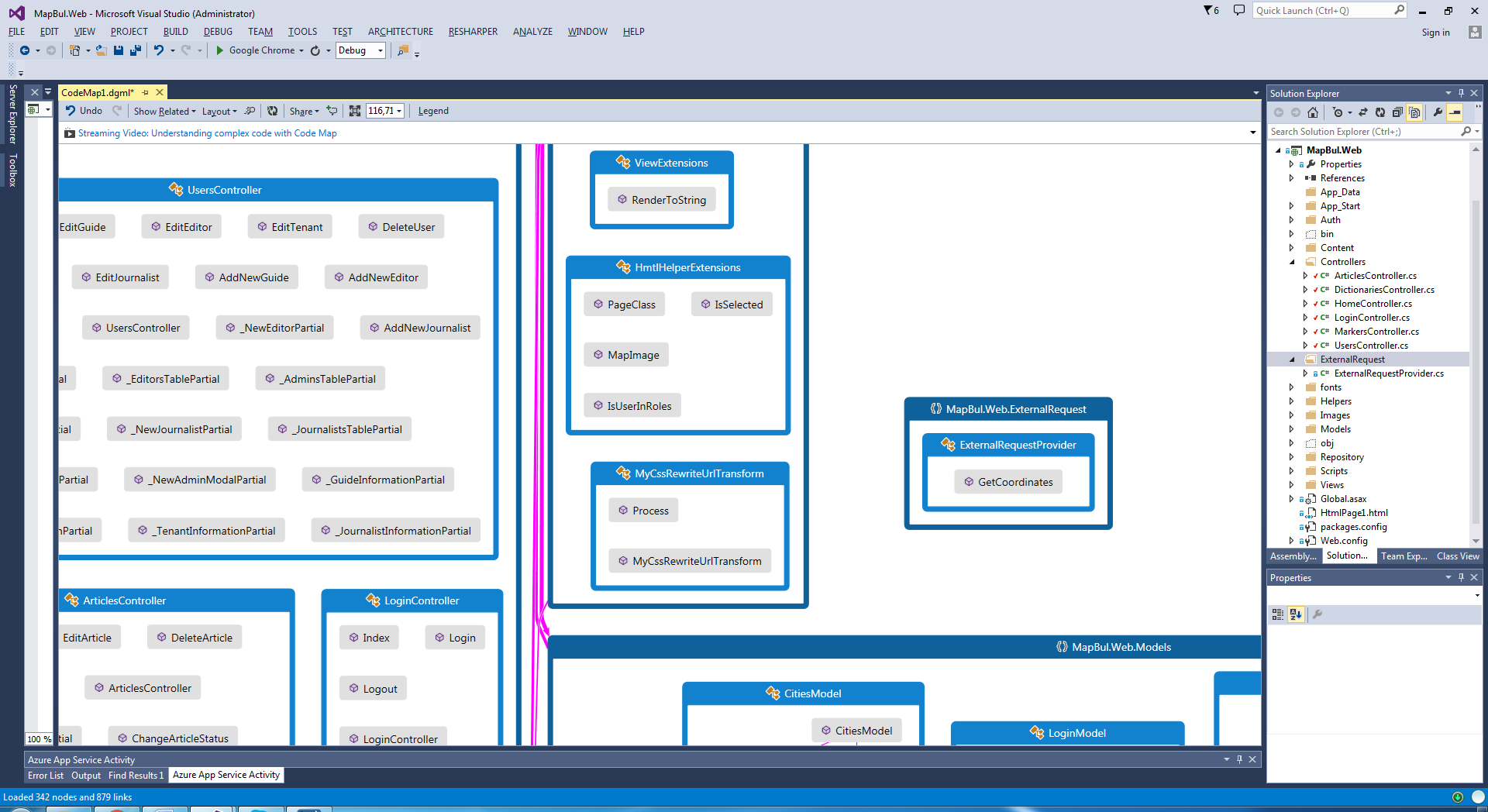
### *MapBul.Web.Helpers*



### *MapBul.Web.Repository*

### 

### *MapBul.Web.ExternalRequest*



### Свойства и зависимости сборки

Сборка реализует Веб-интерфейс для кабинета пользователей и кабинета администратора. Проект реализован с использованием паттерна *MVC.*

Зависимости сборки:

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

Помимо зависимостей приложения, используются следующие сторонние *javaScript* библиотеки:

* *jQuery*—зависимость библиотеки *Bootstrap* (<https://jquery.com/>).
* *Bootstrap—*набор инструментов для внешней оболочки Web-приложений (<http://www.getbootstrap.com>).
* *Inspinia Responsive Admin Theme—*набор элементов управления и отображения информации на основе библиотеки *Bootstrap.* (<https://wrapbootstrap.com/theme/inspinia-responsive-admin-theme-WB0R5L90S>).

Целевой файл сборки—*MapBul.Web.dll,* развертываемый на *IIS* сервере. Сборка не подписана.

## Описание объектов

Сборка состоит из следующих пространств имен:

* *Controllers.* В данном пространстве имен описаны все классы-контроллеры. Методы контроллеров соответствуют запрашиваемым страницам и действиям в приложении.
* *Auth*. Содержит классы, реализующие логику авторизации партнеров и администраторов в приложении.
* *Models*. Здесь описаны модели данных для представлений.
* *Helpers*. В данном пространстве имен содержатся классы расширения для стандартных классов *MVC.* Методы классов реализуют типичные манипуляции с данными для проекта.
* *Repository*. Классы в данном пространстве имен реализуют *CRUD* функции для таблиц базы данных.
* *ExternalRequest*. Содержит класс, реализующий запросы к внешним сервисам, например, *API Google Maps*.

### Пространство имен *MapBul.Web.Controllers*

В данном пространстве имен описаны следующие классы:

* *ArticlesController*—контроллер раздела статей. Методы данного класса обеспечивают отображение страниц раздела, обработку запросов на создание, изменение и удаление статей.
* *DictionariesController*—контроллер раздела словарей. Методы данного класса обеспечивают отображение страниц раздела, обработку запросов на создание, изменение и удаление стран, городов и категорий статей и маркеров.
* *HomeController*—контроллер главной страницы. Содержит один метод, возвращающий экран с картой.
* *LoginController*—контроллер страницы авторизации. Методы обеспечивают отображение страницы авторизации, обработку входа и выхода из системы.
* *MarkersController*—контроллер раздела маркеров. Методы данного класса обеспечивают отображение страниц раздела, обработку запросов на создание, изменение и удаление маркеров.
* *UsersController*—контроллер раздела пользователей. Методы данного класса обеспечивают отображение страниц раздела, обработку запросов на создание, изменение и удаление пользователей.

### Пространство имен *MapBul.Web.Models*

В данном пространстве имен описаны следующие классы-модели:

* *ArticlesListModel—*модель списка статей.
* *NewArticleModel—*модель новой статьи или события. Класс наследуется от типа *article*.
* *CitiesModel—*модель списка стран и городов.
* *CategoriesModel—*модель списка категорий.
* *MapInfoModel—*модель главной страницы сайта.
* *LoginModel—*модель авторизации пользователей.
* *MarkersListModel—*модель списка маркеров.
* *NewMarkerModel—*модель нового маркера.
* *TenantsListModel—* модель списка жителей.
* *JournalistsListModel—*модель списка журналистов.
* *GuidesListModel—*Модель списка гидов.
* *EditorsListModel*—модель списка редакторов.
* *AdminsListModel—*модель списка администраторов.
* *NewEditorModel—*модель нового редактора. Класс наследуется от типа *editor*.
* *NewJournalistModel—*Модель нового журналиста. Класс наследуется от типа *journalist*.
* *NewAdminModel—*Модель нового администратора. Класс наследуется от класса *admin*.
* *NewGuideModel—*модель нового гида. Класс наследуется от типа *guide*.
* *NewTenantModel—* модель нового жителя. Класс наследуется от типа *tenant.*

### Пространство имен *MapBul.Web.Auth*

В данном пространстве имен содержатся классы, реализующие функционал авторизации пользователей. Здесь описаны следующие классы:

Интерфейс *IAuthProvider*. Интерфейс содержит следующие сигнатуры методов и свойств:

* *Login*—метод авторизации в приложении
* *Logout*—метод выхода из приложения
* *RefreshPrincipal()—*метод обновления куки авторизации.
* *IsAuthenticated*—свойство, определяющее, авторизован ли текущий пользователь.
* *CurrentUserName*—свойство, определяющее имя текущего пользователя.
* *UserType—*свойство, возвращающее тип авторизованного пользователя.
* *UserGuid—*свойство, возвращающее идентификатор текущего пользователя.
* *UserId—*свойство, возвращающее ID текущего пользователя.
* *IsSuperAdmin*—свойство, определяющее, является ли пользователь суперадминистратором.

Класс *DefaultAuthProvider.* Наследует интерфейс *IAuthProvider.* В классе описана реализация методов интерфейса с использованием авторизации через формы.

### Пространство имен *MapBul.Web.Repository*

Данное пространство имен содержит два объекта:

* Интерфейс *IRepository*, содержащий сигнатуры всех *CRUD* методов базы данных
* Класс *MySqlRepository*, реализующий данный интерфейс.

### Пространство имен *MapBul.Web.ExternalRequest*

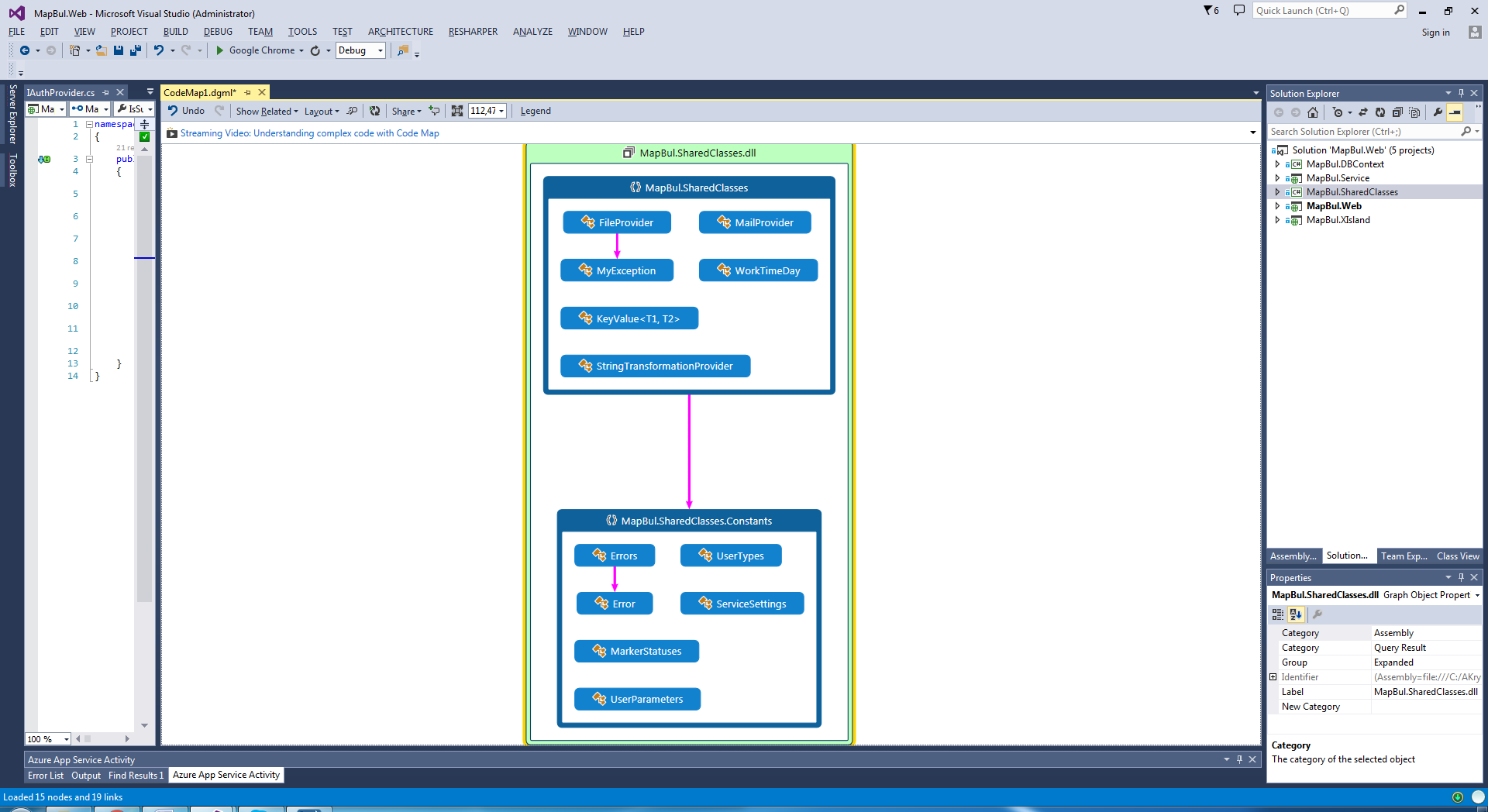
Данное пространство имен содержит один класс с одним методом, с помощью которого происходит определение координат точки по ее адресу.

Страницы *Web*-приложения описаны в представлениях. Все представления находятся в папке *View.* Представления разбиты по категориям.

Скрипты и таблицы стилей для страниц сайта находятся в папках *Scripts* и *Content*.

# БИБЛИОТЕКА СОВМЕСТНО ИСПОЛЬЗУЕМЫХ ТИПОВ *MapBul.SharedClasses*

## Структура проекта



## Описание объектов

### Пространство имен *MapBul.SharedClasses*

В данном пространстве имен определены классы, используемые в других проектах решения.

* FileProvider—провайдер работы с файловой системой
* MailProvider—провайдер работы с электронной почтой
* MyException—исключение, описывающее возможные особые состояния в системе
* WorkTimeDay—сериализуемый класс для передачи времени работы в мобильное приложение
* KeyValue—generic тип ключ-значение
* StringTransformationProvider—провайдер методов работы со строковыми данными.

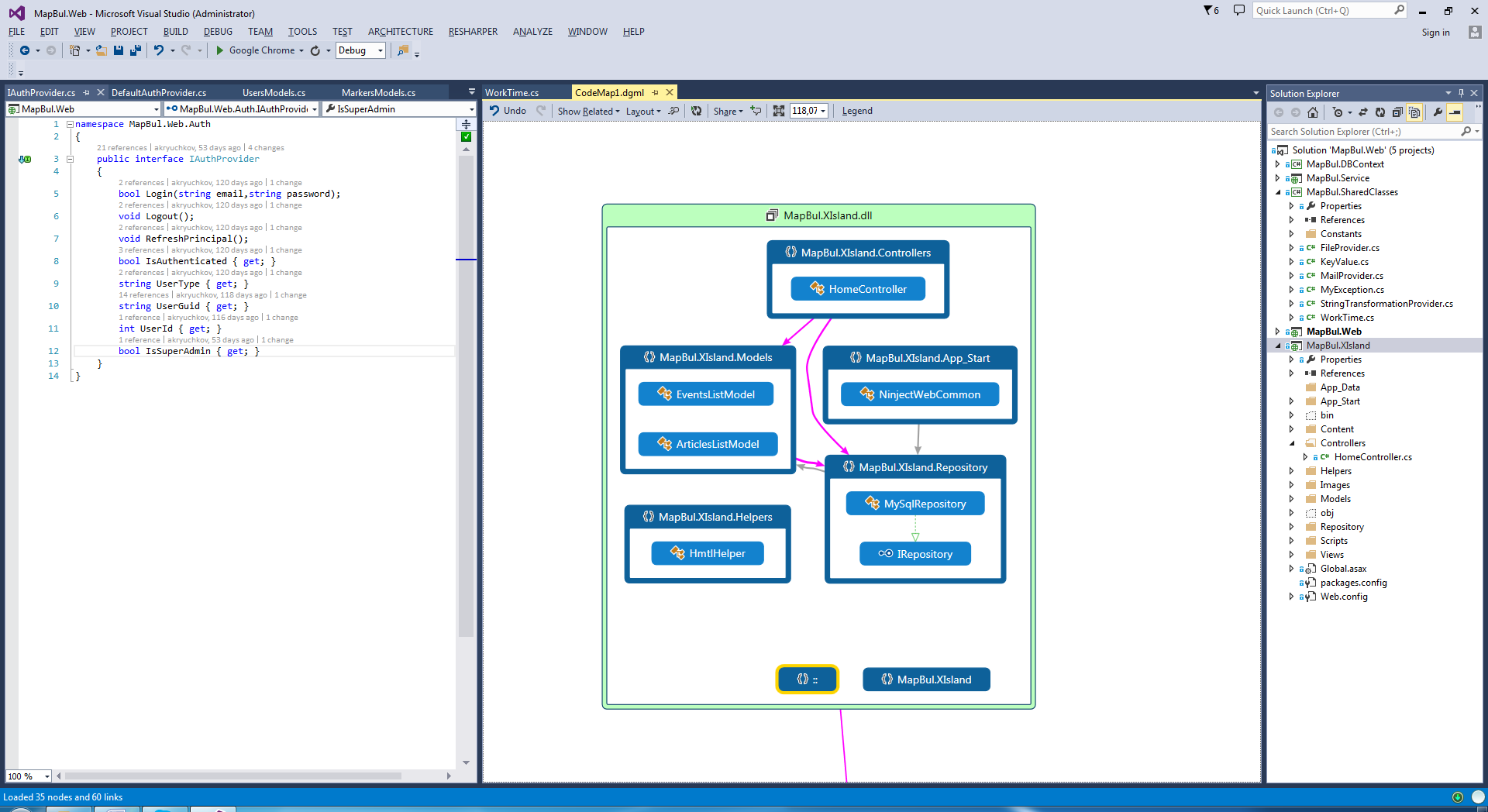
### Пространство имен *MapBul.SharedClasses.Constants*

В данном пространстве имен определены классы, содержащие строковые и численные константы системы. В пространстве имен реализованы следующие классы:

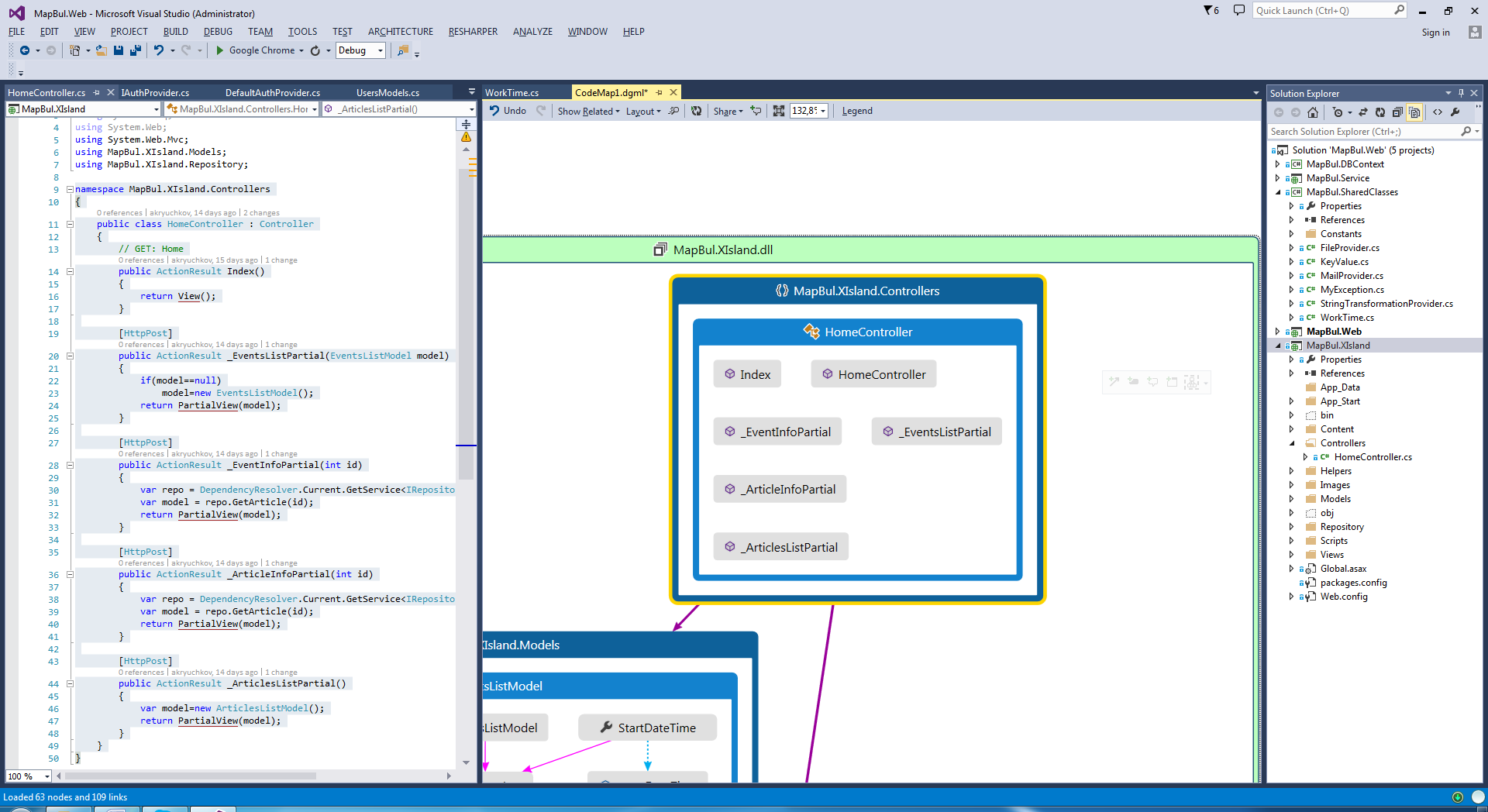
* *Errors*—Список возможных ошибок *CRUD* операций с данными в БД.
* *Error—*Описатель ошибки. Содержит номер ошибки и ее описание.
* *UserTypes*—Класс, содержащий строковые константы, являющиеся сокращениями названия типов пользователей. Значения вынесены в отдельный класс для того, чтобы не использовать в коде «магические» значения.
* *ServiceSettings*—Класс, содержащий константные значения настроек системы, например, количество статей, передаваемых за раз в приложение.
* *MarkerStatuses*—Класс, содержащий строковые константы, описывающие возможные статусы маркеров и статей.
* *UserParameters*

# *WEB*-ПРИЛОЖЕНИЕ *MapBul.XIsland*

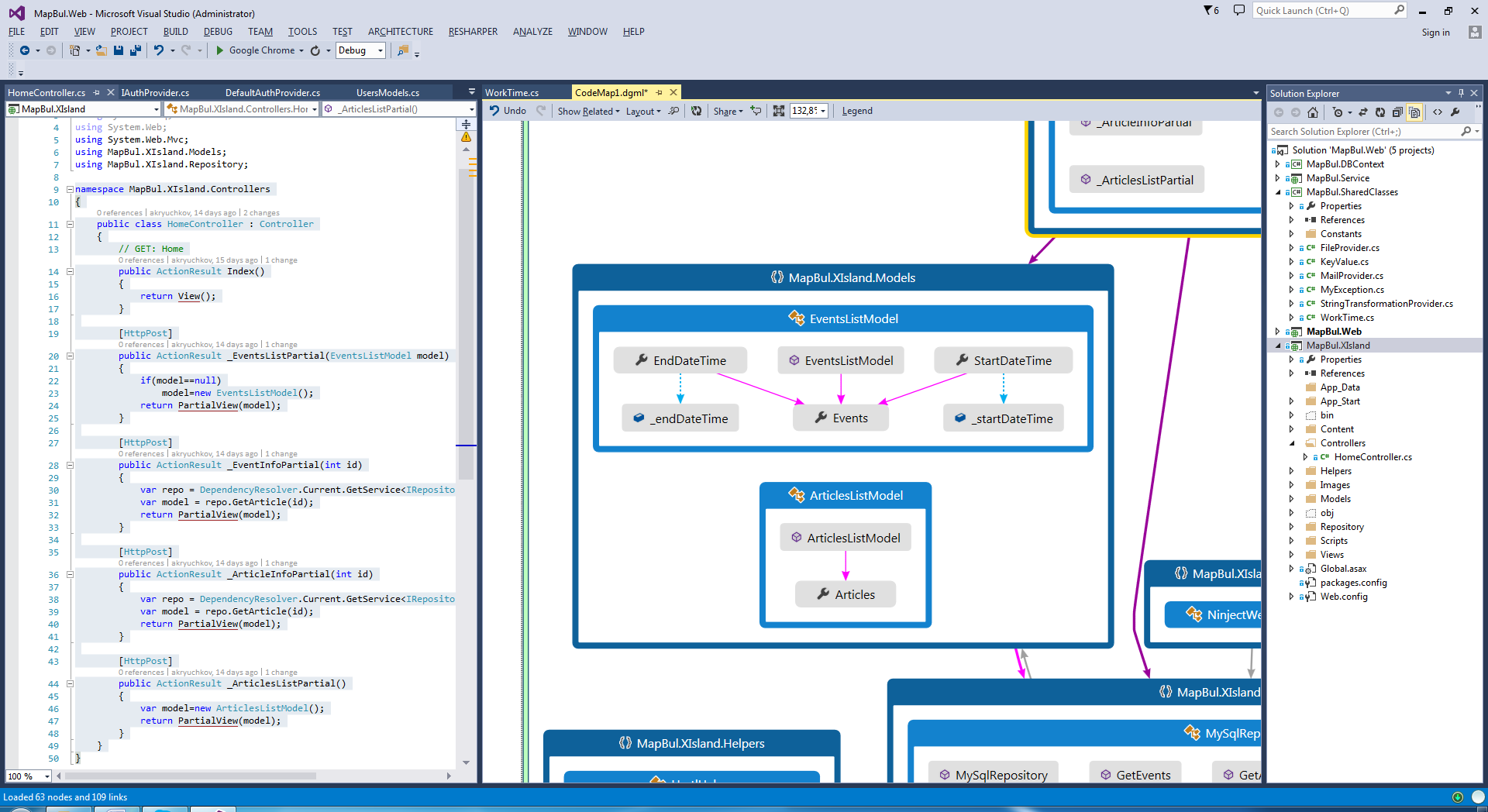
## Структура проекта



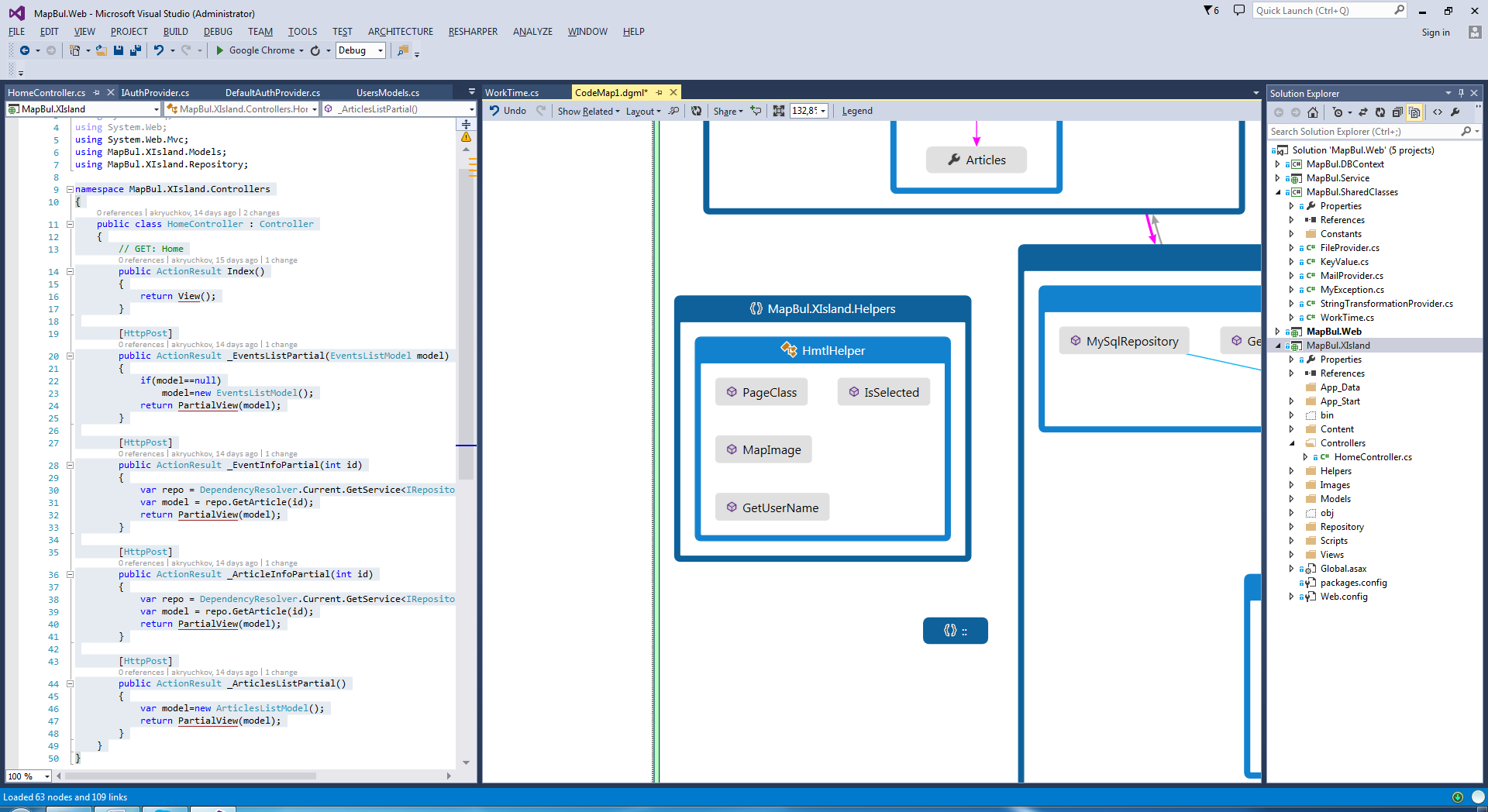
### *MapBul.XIsland.Controllers*



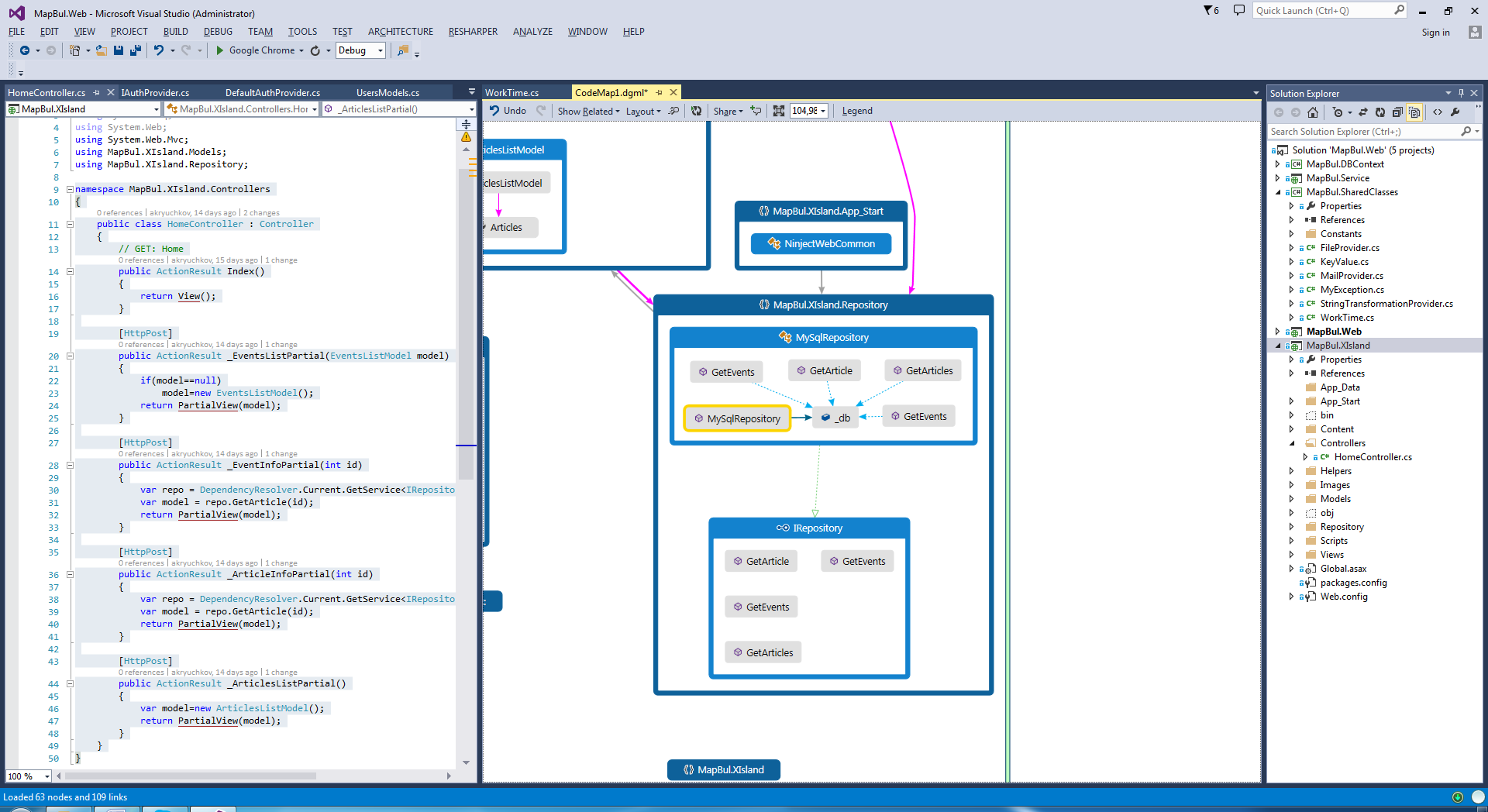
### *MapBul.XIsland.Models*



### *MapBul.XIsland.Helpers*



### *MapBul.Web.Repository*



### Свойства и зависимости сборки

Сборка реализует Веб-приложение домашней страницы *XIsland*. Проект реализован с использованием паттерна *MVC.*

Зависимости сборки:

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

Помимо зависимостей приложения, используются следующие сторонние *javaScript* библиотеки:

* *jQuery*—зависимость библиотеки *Bootstrap* (<https://jquery.com/>).

Целевой файл сборки—*MapBul.XIsland.dll,* развертываемый на *IIS* сервере. Сборка не подписана.

## Описание объектов

Сборка состоит из следующих пространств имен:

* *Controllers.* В данном пространстве имен описаны все классы-контроллеры. Методы контроллеров соответствуют запрашиваемым страницам и действиям в приложении.
* *Models*. Здесь описаны модели данных для представлений.
* *Helpers*. В данном пространстве имен содержатся классы расширения для стандартных классов *MVC.* Методы классов реализуют типичные манипуляции с данными для проекта.
* *Repository*. Классы в данном пространстве имен реализуют *CRUD* функции для таблиц базы данных.

### Пространство имен *MapBul.XIsland.Controllers*

В данном пространстве имен описаны следующие классы:

* *HomeController*—контроллер главной страницы. Содержит методы, возвращающие главную страницу сайта и частичные представления.

### Пространство имен *MapBul. XIsland.Models*

В данном пространстве имен описаны следующие классы-модели:

* *EventsListModel—*модель списка событий. Содержит даты фильтрации событий и список событий.
* *ArticlesListModel—*модель списка статей.

### Пространство имен *MapBul. XIsland.Repository*

Данное пространство имен содержит два объекта:

* Интерфейс *IRepository*, содержащий сигнатуры всех *CRUD* методов базы данных
* Класс *MySqlRepository*, реализующий данный интерфейс.

### Пространство имен *MapBul.XIsland.Helpers*

Данное пространство имен описывает классы расширения для стандартных классов *MVC*. В пространстве имен описаны следующие классы:

* *HtmlHelper*—класс, реализующий дополнительные методы для класса *Html.*

# ЗАПУСК ПРИЛОЖЕНИЙ

Решение содержит пять проектов. Проекты *Web, Service* и *XIsland* являются *Web*-приложениями, развертываемыми на *IIS* сервере. Для развертки этих приложений необходимо построить целиком все решение и скопировать для каждого приложения на сервер содержимое следующих папок: *bin*, *content*, *fonts*, *images*, *scripts*, *views*, а также файлы: *background*.*jpg*, *favicon*.*ico*, *packages*.*config*, *precompiledApp*.*config*, *Web*.*config*. Далее, следует настроить соответствующим образом сервер *IIS*.

Проекты *SharedClasses* и *DBContext* являются библиотеками классов, целевые файлы сборки необходимо скопировать в папку *bin* *Web-*приложений.

В случае сбоев, для сброса состояния приложения достаточно перезагрузить *IIS* сервер.

# ПРОГРАММНЫЕ СРЕДСТВА

Приложение написано на языке C#, его исполнение происходит на сервере *IIS.* Соответственно, для запуска приложения необходимо следующее программное обеспечение: *.NET Framework 4.5 (*[*https://www.microsoft.com/ru-ru/download/details.aspx?id=30653*](https://www.microsoft.com/ru-ru/download/details.aspx?id=30653)*), IIS-server 7.5 (*[*https://www.microsoft.com/ru-ru/download/details.aspx?id=1038*](https://www.microsoft.com/ru-ru/download/details.aspx?id=1038)*).*

Для доступа к базе данных *MySQL* используется программа *MySQLWorkbench (*[*https://www.mysql.com/products/workbench/*](https://www.mysql.com/products/workbench/)*)*

# РЕКВИЗИТЫ ДОСТУПА

## *RDP*

IP адрес сервера: 185.76.145.214

Имя пользователя: Администратор

Пароль: qg>R&&;2

## *FTP*

IP адрес сервера: 185.76.145.214

Имя пользователя: mapbul

Пароль: mapbulftp

## *MySQL*

IP адрес сервера: 185.76.145.214

Имя пользователя: mapbul

Пароль: mapbulsql