|  |
| --- |
| codeplex |
| Web Management Studio |
| Product overview |

|  |
| --- |
| Алексей Ширшов  16.09.2009 |

Оглавление

[1. Страницы 3](#_Toc241478344)

[2. Модели представлений 3](#_Toc241478345)

[2.1. Описание 3](#_Toc241478346)

[2.2. Функции 4](#_Toc241478347)

[2.2.1. Добавление, редактирование, удаление 4](#_Toc241478348)

[2.2.2. Список моделей, поиск 5](#_Toc241478349)

[3. Персистентные модели 6](#_Toc241478350)

[3.1. Описание 6](#_Toc241478351)

[3.2. Функции домена 7](#_Toc241478352)

[3.2.1. Синхронизация с источником 7](#_Toc241478353)

[3.2.2. Добавление таблиц в источник 7](#_Toc241478354)

[3.2.3. Добавление сущностей из источника 7](#_Toc241478355)

[3.3. Функции модели 7](#_Toc241478356)

[3.3.1. Создание, редактирование, удаление модели 7](#_Toc241478357)

[3.3.2. Список моделей, поиск 7](#_Toc241478358)

[4. Контроллер страницы 7](#_Toc241478359)

[4.1. Описание 7](#_Toc241478360)

[5. Класс контроллера страницы 8](#_Toc241478361)

[5.1. Описание 8](#_Toc241478362)

[6. Контроллеры представлений 8](#_Toc241478363)

[6.1. Описание 8](#_Toc241478364)

[6.2. Функции 10](#_Toc241478365)

[6.2.1. Создание, редактирование, удаление 10](#_Toc241478366)

[7. Шаблоны 11](#_Toc241478367)

# 1. Страницы

Страница – основной объект системы. Она является контейнером представлений.

Представление – пользовательский контрол. Может располагаться на нескольких страницах.

Страница – единственная точка входа отображения результатов работы приложения. Воспринимает на вход модель страницы, которую формирует контроллер страницы. Контроллер страницы выбирается на основе правил, задающих урлы страницы. Правила задаются в роутах.

Страница

Page controllers

Page controllers

Page controllers

View

URL

Модель страницы

Page controllers

Page controllers

Page controllers

View models

Контроллер страницы

Page controllers

Page controllers

Page controllers

View controllers

# 2. Модели представлений

## 2.1. Описание

Модели представлений – модель, не сохраняющая состояние между запросами. Является частью модели страницы. Доступна для создания и редактирования в контроллере представления. Используется в представлении. Модели представлений сохраняются в хранилище моделей.

Хранилище представлений – иерархическая структура хранения моделей представлений. Степень вложенности не ограниченна.

Модель страницы – контейнер моделей представления. Например, если на странице имеется два контрола ctrl1 и ctrl2, то модель страницы будет следующей:

public class Page1Model

{

public ViewModel1 ctrl1\_Model {get;set;}

public ViewModel2 ctrl2\_Model {get;set;}

}

Где Page1Model – модель страницы, ViewModel1 и ViewModel2 – модели представлений.

Модель страницы формируется автоматически на основе информации о том, какие представления добавлены на страницу. Модели страниц не сохраняются в общее хранилище моделей.

Так как одно представление может быть вложено в другое, модель представления также может содержать модели других представлений. Такие модели не могут сохраняться в хранилище моделей, они также как и модели страницы генерируются автоматически

public class ViewModel1

{

public string ModelItem {get;set;}

}

public class Page1\_ctrl1\_Model : ViewModel1

{

public ViewModel2 ctrl1\_Model {get;set;}

}

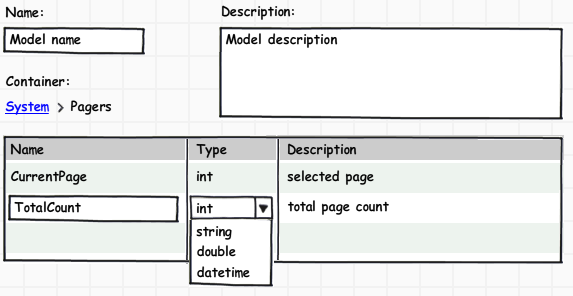
Где ViewModel1 – модель представления, сохраняющееся в хранилище, ViewModel2 – модель вложенного представления, Page1\_ctrl1\_Model – модель сгенерированная автоматически.

## 2.2. Функции

### 2.2.1. Добавление, редактирование, удаление

Модель имеет следующие атрибуты

1. Имя
2. Описание
3. Контейнер
4. Список свойств
   1. Имя свойства
   2. Тип
   3. Описание



Все свойства редактируемы.

В случае

* удаления какого-либо параметра модели, уже использующейся в каком-либо представлении
* изменения типа какого-либо параметра модели, уже использующейся в каком-либо представлении
* переименования какого-либо параметра модели, уже использующейся в каком-либо представлении

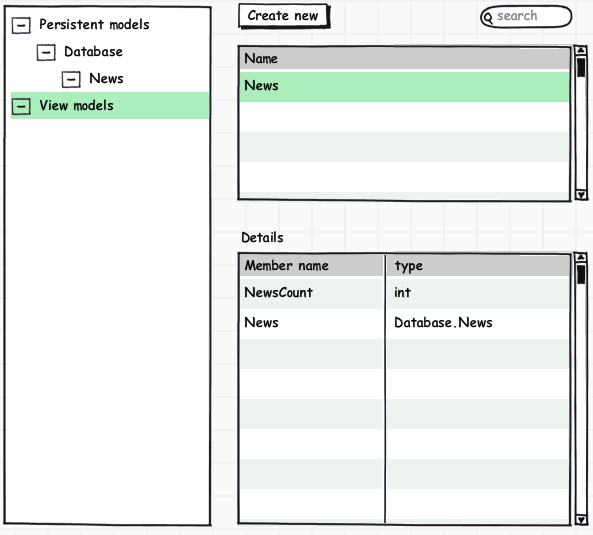
должно выдаваться предупреждение со списком использующих модель контроллеров. Если пользователь подтверждает выполнение, производится операция.

Если модель удаляется и она уже используется в каком-либо представлении, необходимо выдать предупреждение со списком использующих модель контроллеров.

### 2.2.2. Список моделей, поиск

Список моделей отображается для выбранного контейнера. Контейнеры – иерархическая структура хранилища моделей.

Контейнер можно переименовывать и добавлять. Если в контейнере нет моделей, его можно удалить.



Список моделей можно фильтровать по названию модели. При выборе модели из списка в разделе Details появляются параметры модели. Создание модели доступно на этой же странице без перехода на страницу модели.

# 3. Персистентные модели

## 3.1. Описание

Персистентные модели – модели, сохраняющие состояние между запросами. Группируются по доменам.

Домены – набор персистентных моделей, хранящихся в одном источнике данных. С доменом связана строка подключения. Имя домена – имя linq контекста.

## 3.2. Функции домена

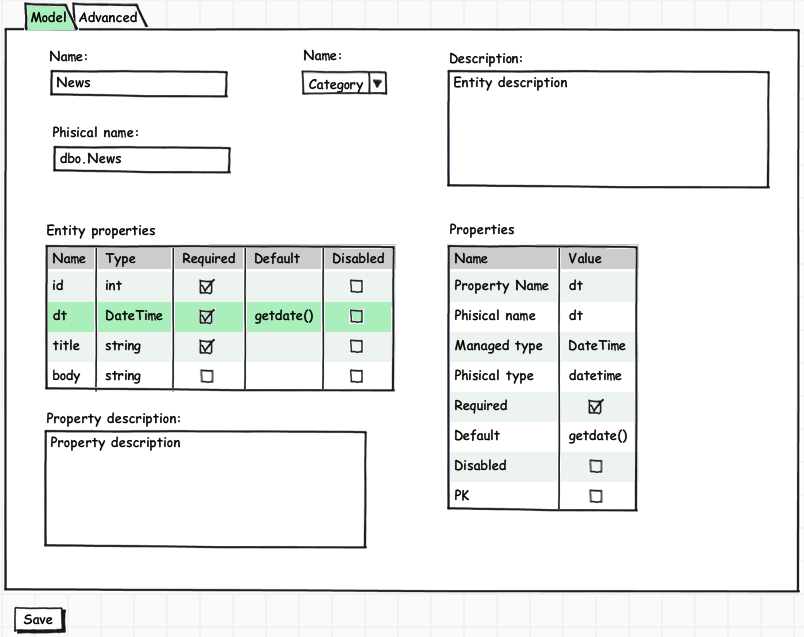
### 3.2.1. Синхронизация с источником

### 3.2.2. Добавление таблиц в источник

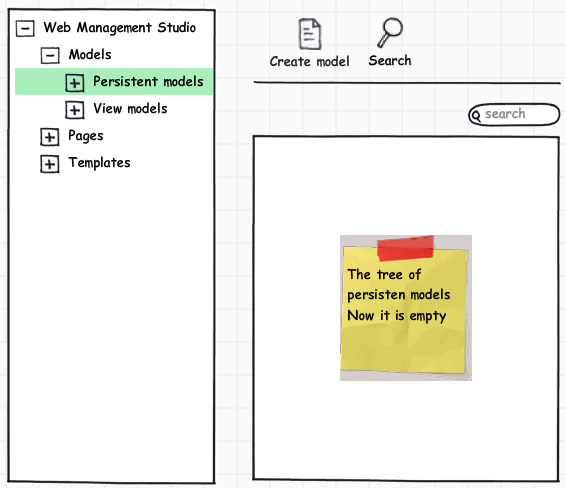
### 3.2.3. Добавление сущностей из источника

## 3.3. Функции модели

### 3.3.1. Создание, редактирование, удаление модели



### 3.3.2. Список моделей, поиск



# 4. Контроллер страницы

## 4.1. Описание

Контроллер страницы – блок кода, формирующий модель страницы, а также все модели всех представлений данной страницы.

Контроллер страницы может иметь входные параметры. Они должны автоматически привязываться к соответствующим полям запроса (по имени).

# 5. Класс контроллера страницы

## 5.1. Описание

Контроллер страницы предназначен для формирования модели страницы с помощью контроллеров представлений. Он представляет собой метод в классе контроллера страницы, который вызывает другие методы этого класса (методы реализующие контроллеры представлений).

Класс контроллера страницы содержит контекст – набор полей, доступных всем контроллерам представлений. Контекст инициализируется при создании экземпляра класса контроллера страницы.

Одной странице соответствует один класс контроллера страницы и наоборот. Код инициализации контекста задается в дизайнере страницы.

Если в качестве контекста выбран домен, он инициализируется автоматически.

Класс контроллера страницы полностью генерируется автоматически на основе используемых контроллеров представлений и кода инициализации контекста.

# 6. Контроллеры представлений

## 6.1. Описание

Контроллеры представлений – функции, возвращающие модели представлений. Контроллер представлений включается в класс контроллера страницы как метод.

Контроллеры представлений (view controllers) предназначены для создания моделей для представлений (view models). Контроллеры представлений являются частью контроллеров страниц (page controllers) и участвуют наравне с последними в формировании модели страницы (page model).

Контроллер страницы создает модель страницы и заполняет свойства ctrl1\_Model и ctrl2\_Model с помощью контроллеров представлений.

public ActionResult Index()

{

Page1Model model = new Page1Model();

model.ctrl1\_Model = ctrl1\_GetModel();

model.ctrl2\_Model = ctrl2\_GetModel();

return View(model);

}

Где Index – контроллер страницы, ctrl1\_GetModel и ctrl2\_GetModel – контроллеры представлений.

Контроллеры представлений создаются автоматический при добавлении контрола на страницу. При добавлении контрола в контрол, контроллер также попадает в контроллер страницы.

public ActionResult Index()

{

Page1Model pageModel = new Page1Model();

{

ViewModel1 Model = ctrl1\_GetModel();

pageModel.ctrl1\_Model = Model;

Model. ctrl1\_Model = ctrl1\_ctrl1\_GetModel();

}

return View(pageModel);

}

Контроллеру доступен контекст. Контекст – набор данных, хранящийся в классе контроллера страницы и доступный для всех контроллеров представлений.

Контроллер представлений может иметь входные параметры. Для задания значения параметров могут использоваться параметры контроллера страницы или свойства модели родительского представления.

public ActionResult Index(int ID)

{

Page1Model pageModel = new Page1Model();

{

ViewModel1 Model = ctrl1\_GetModel();

pageModel.ctrl1\_Model = Model;

Model. ctrl1\_Model = ctrl1\_ctrl1\_GetModel(ID, Model.ModelItem);

}

return View(pageModel);

}

В случае совпадения имен параметров контроллера страницы и контроллера представления, привязка должна осуществляться автоматически. Если невозможно найти параметр с таким именем в контроллере страницы, список параметров контроллера страницы автоматически расширяется для включения этого параметра.

## 6.2. Функции

### 6.2.1. Создание, редактирование, удаление

Контроллер представления имеет следующие атрибуты:

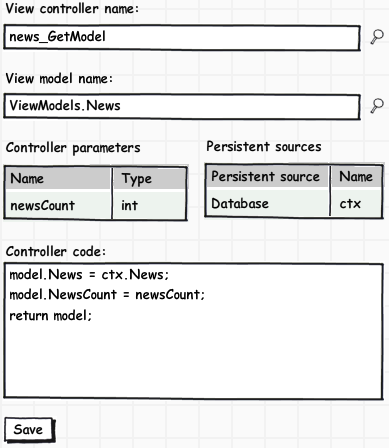
* Имя
* Модель представления
* Параметры
  + Имя
  + Тип
* Контекст
  + Имя
  + Тип
* Код

Имя контроллера и код можно менять без ограничений. Если контроллер представлений уже используется в каком-либо контроллере страницы необходимо выдавать предупреждение при сохранении если пользователь

* Изменяет модель представления
* Изменяет список параметров (добавление, удаление, переименование, изменение типа)
* Изменяет список контекстных параметров (добавление, удаление, переименование, изменение типа)

В предупреждении необходимо указать список затрагиваемых контроллеров страниц. При этом пользователю предоставляются следующие варианты:

* Отменить операцию сохранения
* Сохранить контроллер под новым именем
* Сохранить контроллер под старым именем (возможны косяки с существующим кодом)

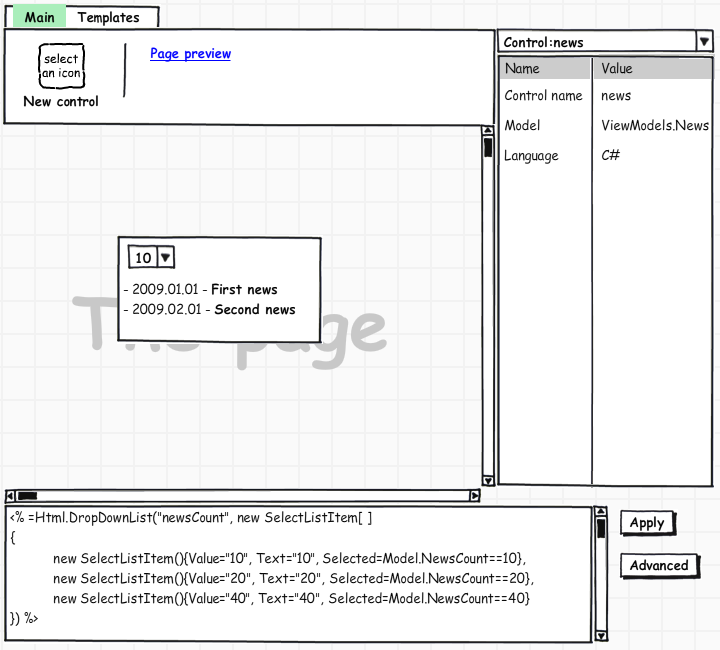


При удалении также нужно выдать предупреждение, если контроллер уже связан с контроллерами страниц.

# 7. Дизайнер

Дизайнер – визуальный редактор страниц и представлений. В нем можно выполнять редактирование

* Окно редактора
* HTML-кода
* Окон свойств выбранного объекта
* Тулбар



В режиме редактирования страницы, в дизайнере появляется окно редактирования процедуры инициализации контекста.

При выборе в дизайнере контрола, появляется окно редактирования привязки параметров контроллера представления.

