

Разработка приложений на платформе .NET

Лекция 22.

ADO.NET

Отсоединенная модель доступа к данным

На прошлой лекции

Присоединенная модель доступа к данным

- Создание соединения `DbConnection`
 - Использование строки подключения
- Создание и настройка команды `DbCommand`.
 - Использование параметров
- Выполнение команды `ExecuteXXX`
- Чтение данных с помощью `DbDataReader`
- Освобождение `DbDataReader` и закрытие соединения `DbConnection`

Сегодня

Отсоединенная модель доступа к данным

- Понятие отсоединенного уровня
- Основные типы
 - DataSet
 - DataTable
 - DataColumn
 - DataRow
- Адаптеры
- Типизированные DataSet
- Отображение данных с помощью DataGrid (WPF)

Отсоединенный уровень

- Моделирует в памяти реляционные данные из БД
- Работа с данными, как будто есть постоянное подключение к БД
- Этапы работы:
 - Соединение с БД, получение данных
 - Работа с данными: редактирование, удаление, добавление без необходимости подключения к БД
 - Возможность отмены изменений
 - Эффективное применение изменений данных в БД (открытие соединения, внесение только изменений в данных и закрытие соединения)
- Вся логика работы с БД вынесена в отдельный тип – адаптер (DbDataAdapter)



- Основные классы могут использоваться и без БД
 - таблицы, столбцы, строки, отношения между таблицами

ОСНОВНЫЕ ТИПЫ

- **DataSet** – представляет реляционные данные в памяти
- **DataTable** – представляет таблицу с данными
- **DataColumn** – представляет схему таблицы
- **DataRow** – представляет строку в таблице. Содержит данные.
- **DataRelation** – отношение таблиц
- **DbDataAdapter** – адаптер данных. Поставляет данные из/в БД

DataSet

- Представляет реляционные данные в памяти
- Свойство **Tables** содержит коллекцию таблиц DataTable
- Свойство **Relations** содержит коллекцию отношений родительский/дочерний DataRelations между своими таблицами.
- Можно использовать навигацию между таблицами по отношениям.
- Свойство ExtendedProperties – представляет доп. информацию в виде пар имя/значение

DataSet. Свойства

- **DataSetName** – имя экземпляра DataSet
- **CaseSensitive** – чувствительность к регистру при сравнении (false)
- **EnforceConstraints** – применяются ли правила ограничений (true)
- **HasErrors** – имеются ли ошибки в любой строке любой таблицы в этом DataSet
- **RemotingFormat** – как DataSet должен сериализоваться (XML)

DataSet. Методы

- **RejectChanges()** – отменяет все изменения в DataSet (сделанные после загрузки или после вызова AcceptChanges())
- **AcceptChanges()** – применяет все изменения в DataSet
- **HasChanges()** – имеются ли изменения в DataSet
- **GetChanges()** – возвращает копию DataSet, содержащую только изменения
- **Clear()** – очищает DataSet от данных (структура сохраняется)
- **Clone()** / **Copy()** – копируют структуру / структуру и данные в новый DataSet
- **ReadXml()** / **WriteXml()** – чтение / запись DataSet в поток (только данные или структуру и данные)

DataTable

- Представляет таблицу с данными
- Обладает многими свойствами и методами аналогичными DataSet
- Свойство **Columns** – содержит коллекцию колонок DataColumn
- Свойство **Rows** – содержит коллекцию строк DataRow

DataTable

- **TableName** – имя таблицы
- **ChildRelations / ParentRelations** – дочерние / родительские отношения для таблицы
- **Constraints** – коллекция ограничений в таблице
- **PrimaryKey** – массив столбцов – первичный ключ
- **DataSet** – ссылка на DataSet, которому принадлежит данная таблица (если есть)
- **CaseSensitive, Copy, RemotingFormat** и др. – аналогично DataSet

DataColumn

- Представляет один столбец в DataTable
- Множество всех DataColumn в таблице представляют схему таблицы
- Определяет тип данных в колонке
- Может содержать набор ограничений:
 - Первичный ключ
 - Значение по умолчанию
 - Уникальность
 - Допустимость DBNull
 - Разрешение только на чтение и др.

DataColumn

- ColumnName – имя колонки
- Caption – отображаемый заголовок колонки
- DataType – тип данных в данном столбце
- AutoIncrement, AutoIncrementSeed, AutoIncrementStep – задание автоинкремента для колонки
- AllowDBNull – задает, могут ли содержаться пустые значения
- DefaultValue – задает значение по умолчанию
- Unique – указывает, что данные в столбец должен быть уникальным
- ReadOnly – указывает, что данные только для чтения
- Table – получает DataTable, содержащий данную колонку
- Ordinal – числовое положение колонки в коллекции колонок Columns

DataRow

- Представляет конкретные **данные** в таблице
- Нельзя создать без таблицы. Нет конструктора
Создание с помощью
`dataRow = dataTable.NewRow();`
- Доступ к данным, содержащимся в строке с помощью индекса по номеру или имени колонки
- Имеет состояния

Демонстрация

Простой DataSet

DataRow. Свойства

- **RejectChanges()** – отменяет все изменения в строке (сделанные после загрузки или после вызова **AcceptChanges()**)
- **AcceptChanges()** – применяет все изменения в строке
- **Table** – таблица, содержащая строку
- **BeginEdit()**, **EndEdit()**, **CancelEdit()** – начало, окончание и отмена редактирования
- **Delete()** – помечает строку для удаления
- **HasErrors**, **GetColumnsInError**, **GetColumnError**, **ClearErrors**, **RowError** – позволяют определить ошибки в данных
- **RowState** – состояние текущей строки

RowState

- Added - добавлена
 - Deleted - удалена
 - Detached – не присоединена ни к одной таблице
 - Modified - изменена
 - Unchanged – неизменна
-
- По этим состояниям определяются изменения в таблицах и DataSet

Демонстрация

Состояния строки

Адаптеры данных

- Классы `DbDataAdapter`
 - `SqlDataAdapter`, `OracleDataAdapter` и т.д.
- Содержит команды:
 - `SelectCommand`
 - `InsertCommand`
 - `UpdateCommand`
 - `DeleteCommand`
- Метод `Fill()` – загружает данные в `DataSet` или таблицу
- Метод `Update` – обновляет данные в БД
- `SqlCommandBuilder` – помогает строить остальные команды по `SelectCommand`

Демонстрации

Редактирование таблицы БД

Типизированный DataSet

- Конкретное приложение работает с конкретной БД
 - Известны типы столбцов
 - Простой DataSet не типизирован
- Типизированный DataSet
 - Автоматическая генерация по БД
 - Типизация всех таблиц, полей и т.д.
 - Удобные методы поиска
 - Контроль типов на этапе компиляции

Типизированный DataSet

- Состоит из:
 - XML-схемы (описывает формат)
 - С#-кода (используется в коде)
 - Дополнительных адаптеров

Демонстрации

Типизированный DataSet