

ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ ПОСТРОЕНИЯ ОПЕРАТИВНОЙ ОТЧЕТНОСТИ

Квасникова И. К.
Научный руководитель А.В. Лепустин
Томский политехнический университет
foxkik@tpu.ru

Введение

Для успешного функционирования организациям необходимы системы планирования и отчетности. Наличие таких систем позволяет обеспечить удобный доступ к данным и предоставлять информацию в подготовленном виде. Использование приложений, выполняющих такую работу, снижают временные затраты персонала на доступ и обработку информации.

Целью данной работы являлось написание веб-приложения для предоставления пользователям информации путем исполнения заранее подготовленных запросов к базам данных различных информационных систем. Необходимость в подобном приложении возникла в силу того, что по различным причинам запросы не были и не могут быть интегрированы в вышеупомянутые системы. Областью применения программного продукта является сфера мониторинга и оценки качества образования.

Проектирование

На начальном этапе планирования было определено, что приложение должно обладать следующим функционалом. Для администратора – добавление запросов, папок, групп баз данных и самих баз, их редактирование и удаление. Для пользователя:

- предоставление списка существующих запросов к базам данных;
- отображение списка баз данных для запросов, которые могут быть исполнены более чем для одной базы;
- предоставление полей для ввода или выбора пользователем значений в случае запроса с параметрами;
- исполнение запроса и отображение результата его выполнения;
- фильтрация полученных данных;
- загрузка на компьютер пользователя полученных результатов с сохранением фильтрации.

После утверждения функционала была спроектирована база данных приложения для хранения следующей информации:

- запросы, написанные на языке SQL;
- иерархия папок, в которых хранятся запросы;
- информация о базах данных, к которым написаны запросы;
- информация о группах, в которые объединены базы с общей структурой.

Базы объединены в группы, так как один и тот

же запрос может быть выполнен для нескольких баз данных разных лет.

Для выполнения некоторых запросов необходимо получение дополнительных данных от пользователя. Такие данные было решено называть параметрами.

Для хранения параметров был введен свой формат записи их в запросе. Таким образом, появилась возможность находить такие параметры в тексте запроса, определять вид представления их пользователю (текстовое поле или выпадающий список).

Формат хранения параметров в тексте запроса имеет следующий вид:

```
###имя_переменной тип##enter/query##  
поясняющий текст##значение по умолчанию/sql-  
запрос####
```

где ### – признак начала параметра,
имя_переменной тип – объявление имени переменной и её типа, которые будут применяться в запросе,

– внутренний разделитель,
enter/query – ключевые слова, определяющие тип параметра: enter – от пользователя требуется ввод значения с клавиатуры, query – пользователю необходимо выбрать значение из выпадающего списка,

поясняющий текст – имя параметра, которое будет отображаться пользователю,
значение по умолчанию – значение переменной, которым заполняется поле (только для типа enter),
sql-запрос – текст запроса для получения перечня значений параметров, выводимых пользователю в виде выпадающего списка (только для типа query),

– признак окончания параметра.

Пример хранения параметра типа enter:

```
###examdate varchar(10)##enter##дата  
экзамена##2016.06.02####
```

Пример хранения параметра типа query:

```
###basename varchar(100)##query##база  
данных##select name from sys.databases where name  
like 'TR55DB2%'####
```

Использованные технологии

Для создания базы данных была использована СУБД Microsoft SQL Server 2008R2, а для создания приложения использовался ASP.NET MVC Framework – фреймворк для создания веб-приложений, который реализует шаблон Model-view-controller.

Схема архитектуры Model-View-Controller (MVC) разделяет приложение на три основных компонента: модель, представление и контроллер.

Объекты моделей являются частями приложения, реализующими логику для домена данных приложения. Представления служат для отображения пользовательского интерфейса приложения. Контроллеры осуществляют взаимодействие с пользователем, работу с моделью, а также выбор представления, отображающего пользовательский интерфейс.

Также при создании приложения была использована библиотека пользовательских визуальных элементов управления DevExpress, так как она предоставляет все необходимые для данного проекта инструменты и обладает подробной документацией.

Для отображения таблицы с результатом выполнения запроса использовался инструмент GridView, который предоставляет возможность сортировать, группировать, фильтровать загруженные в него данные. Для отображения запросов и папок в форме древовидной структуры был использован компонент TreeList. Для удобного размещения этих компонентов на странице был использован компонент Splitter.

Функциональность некоторых из вышеописанных компонентов была расширена путем изменения обработчиков событий на стороне клиента с помощью языка JavaScript.

Созданные компоненты

Кроме вышеописанных таблиц были созданы классы для реализации необходимого функционала приложения (рисунок 1).

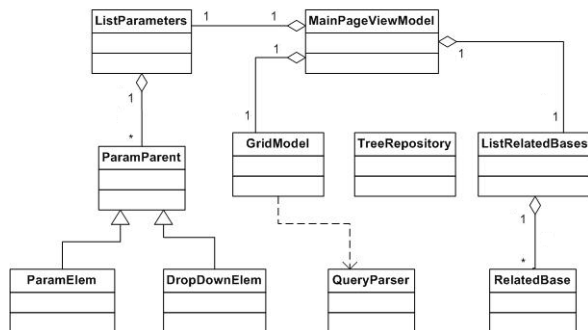


Рис. 1. Диаграмма классов приложения

Класс GridModel необходим для хранения таблицы с результатом выполнения запроса и настроек для отображения в GridView. ParamParent – модель данных для представления параметра, необходимого для выполнения запроса (ParamElem – для вводимого с клавиатуры, DropDownParam – для выбираемого из выпадающего списка). ListParameters – модель для хранения списка параметров, id запроса и id базы. QueryParser – класс для обработки текста запроса с целью нахождения параметров и составления итоговой строки запроса. TreeRepository – класс для формирования древовидного списка папок и запросов. MainPageViewModel – модель данных,

содержащая экземпляры моделей, которые необходимо отобразить на странице.

Интерфейс разработанного приложения

На рисунке 2 изображена страница веб-приложения. Левую часть страницы занимает представляемый в виде дерева список запросов для удобства навигации.

Основная часть центральной панели страницы отведена под таблицу с результатом выполнения запроса. В верхней части центральной панели располагается выпадающий список, в котором выводятся названия баз данных, к которым можно выполнить выбранный запрос. При необходимости в верхней части также появляются поля для ввода пользователем значений параметров запроса.

Если запрос и база данных выбраны, параметры введены, и запрос успешно выполнен, в компоненте GridView отображаются полученные данные. После этого становятся доступными возможности GridView для группирования, сортировки и фильтрации отображаемых данных. Также пользователь может загрузить полученные данные на локальный компьютер в формате MS Excel 2003/2010.

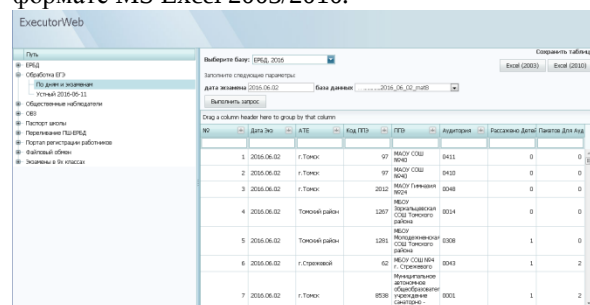


Рис. 2. Страница веб-приложения

Заключение

В результате проделанной работы было создано веб-приложение для отображения и экспорта результатов выполнения запросов к базам данных. Разработан ряд запросов к существующим базам данных различных информационных систем, и произведено первичное заполнение базы данных веб-приложения. В настоящее время приложение находится в стадии внедрения.

Список использованных источников

1. Разработка веб-приложений с использованием ASP.NET MVC Framework. [Электронный ресурс]. URL: <http://www.intuit.ru/studies/courses/527/383/info> (Дата обращения 07.10.2016)
2. DevExpress MVC Extensions [Электронный ресурс]. URL: <https://demos.devexpress.com/MVCxGridViewDemo/s/> (Дата обращения 07.10.2016)
3. Современный учебник JavaScript [Электронный ресурс]. URL: <https://learn.javascript.ru/> (Дата обращения 07.10.2016)