

УДК 50.41

## РАЗРАБОТКА МУЛЬТИМЕДИЙНОГО СЕРВЕРА ДЛЯ ВИДЕОТРАНСЛЯЦИИ

*Шитько А.М.*

*Белорусский государственный технологический университет, г. Минск*

В данной публикации описано созданное веб-приложение, представляющее собой мультимедийный сервер, предназначенный для видеотрансляции. Основным моментом является использование созданного .NET веб-сервиса, дающего возможность более быстрого потокового считывания видеофайла прямо из базы данных ORACLE и воспроизведения его в браузере с помощью новой технологии HTML 5 плеера.

**Введение.** Жизнь современного человека без постоянного доступа во всемирную сеть Internet довольно трудно представить. Повсеместное распространение точек доступа и постоянное увеличение популярности привлекли интерес разработчиков, что в свою очередь привело к изобретению новых интернет-технологий. Поточковая трансляция видеофайлов завоевала популярность с самых первых дней своего существования. Мало какие информационные интернет-ресурсы обходятся без видео, не говоря уже о развлекательных порталах. Трансляцию видео используют практически везде и в самых разных целях. Однако, при большом количестве файлов, встает вопрос об упорядоченном хранении и сортировке видеофайлов. Лучшим решением этой проблемы стало использование базы данных, а именно – СУБД ORACLE.

**Основная часть.** Цель разработки – организация упорядоченного хранения большого объема видеофайлов.

Данное приложение выполняет следующие задачи:

- загрузка собственных видеофайлов на сервер и их автоматическое сохранение в базе данных;
- получение списка всех имеющих в базе данных видеофайлов;
- организация потокового вещания выбранного видеофайла;
- возможность просмотра видеопотока с помощью HTML 5 видеоплеера.

Для хранения видеофайлов в базе данных была создана таблица с тремя полями: *id*, хранящее номер видеофайла для его сортировки и адресации; *name*, хранящее имя видеофайла для отображения в общем списке; *data*, хранящее собственно сам видеофайл. В последнем поле использовался тип данных BLOB, который позволяет хранить большие данные (до 4 Гб) даже при удалении оригинального файла.

Для того чтобы хранить файлы в базе данных, необходимо их туда «загрузить». Для этого используется стандартный элемент управления “*FileUpload*” и с помощью специального запроса добавляется файл:

```
cmd.CommandText = "insert into video_table values (" + new_id.ToString() + ", '" + video_name.Text + "', 'AAA')";  
cmd.ExecuteNonQuery();
```

Во избежание переполнения буферов памяти «кусками» по 1 Мб файл копируется в объект BLOB при помощи входящего потока:

```
using (reader)  
{  
    reader.Read();  
    OracleLob blob = reader.GetOracleLob(2);  
    int bytes = 0;  
    byte[] buffer = new byte[1000000];
```

```
do
{
bytes = hpf.InputStream.Read(buffer, 0, 1000000);
blob.Write(buffer, 0, bytes);
}
```

Немаловажным шагом является формирование индивидуальных ссылок на страницу с HTML 5 плеером для каждого файла, поэтому в ссылку передается параметр «*id*», значение которого соответствует идентификатору файла в базе данных. После этого HTML 5 плеер обращается к веб-сервису с запросом на файл с указанным параметром «*id*».

После формирования ссылки для доступа к файлу базы данных используется «посредник» между HTML 5 плеером и хранилищем. Это веб-сервис, который соединяется с базой данных и выбирает необходимый файл:

```
using(OracleConnection conn = new OracleConnection(GenerateConnectionString()))
{
conn.Open();
OracleCommand cmd = conn.CreateCommand();
cmd.Transaction = cmd.Connection.BeginTransaction();
cmd.CommandText = "select * from video_table where id = " + video_id.ToString();
OracleDataReader reader = cmd.ExecuteReader();
using (reader)
{
reader.Read();
OracleLob blob = reader.GetOracleLob(2);
byte[] buffer = new byte[1000000];
do
{
bytes = blob.Read(buffer, 0, 50000);
Context.Response.OutputStream.Write(buffer, 0, bytes);
}
while (bytes > 0);
blob.Position = 0;
cmd.Transaction.Commit();
}
conn.Close();
}
```

Для увеличения быстродействия воспроизведения в HTML 5 плеер встроен механизм воспроизведения минимальной достаточной информации, то есть плеер начинает воспроизводить видео только тогда, когда у него будет достаточно данных для воспроизведения, при этом, не прекращая загрузку файла. В связи с этим лучшим решением было считывать файл в исходящий поток не целиком, а «кусками» по 50 кб, что позволило сократить задержку до начала воспроизведения до минимума.

**Заключение.** Использование веб-сервисов для потокового считывания из базы данных весьма оправдано, так как позволяет увеличить скорость работы приложения за счет того, что сервисы менее требовательны к ресурсам компьютера. К достоинствам можно отнести также кроссплатформенность и отсутствие необходимости установки веб-сервиса на компьютер.

#### Список цитированных источников

1. Кайт, Т. Oracle для профессионалов. – СПб.: ДиаСофтЮП, 2003. – 848 с.
2. Урман, С. Oracle 9i. Программирование на языке PL/SQL. – М.: Лори, 2004. – 528 с.
3. Шапошников, И.И. Web-сервисы Microsoft .NET. – СПб.: БХВ-Петербург, 2002. – 334 с.