

## **РАЗРАБОТКА ВЕБ-СЕРВИСА УДАЛЁННОГО ВЗАИМОДЕЙСТВИЯ С БАЗАМИ ДАННЫХ**

---

Исследуются наиболее популярные подходы удалённого доступа к СУБД, анализируются их недостатки. Предлагается подход, использующий технологию веб-сервисов, для осуществления взаимодействия с удаленной СУБД, позволяющий в значительной мере избежать недостатков популярных подходов.

### **1. Введение**

На сегодняшний день удалённое использование сервером баз данных в Web – привычное явление. Однако большинство систем, предоставляющих данную возможность, являются серверными СУБД, доступ к базам которых возможно осуществлять удалённо, используя TCP/IP [1]. Такое взаимодействие клиента БД и сервера СУБД имеет ряд недостатков: конструктивная сложность; привязка к конкретной платформе; небезопасность подключения.

### **2. Анализ предметной области**

Рассмотрим более детально перечисленные аспекты предметной области.

Конструктивная сложность состоит в необходимости оборудования машины-сервера (если речь идёт о подключении к удалённому узлу), установке и настройке серверной СУБД, организации сетевого взаимодействия (вне зависимости от того, производится ли подключение к удалённому узлу, или же на 127.0.0.1, необходимо выделить и прослушивать порт(ы) для получения входящих запросов к СУБД), а также в доступе (создание аккаунтов, разграничение прав). Если последний пункт просто необходим для обеспечения безопасности взаимодействия, то по поводу первых можно внести некоторые коррективы.

С точки зрения простоты реализации веб-сервис является одним из наиболее оптимальных решений. Создание программного продукта такого класса не требует высокой квалификации разработчика в области знания протоколов уровня представления – XML-RPC, SOAP, а также языка описания веб-сервисов WSDL.

Привязка к платформе обусловлена необходимостью согласовывать платформу клиентского приложения с платформой СУБД посредством использования соответствующего драйвера или поставщика СУБД (набор классов, предназначенных для взаимодействия с хранилищем данных определенного типа). По этой причине наиболее распространены сочетания PHP + MySQL, C# + MS SQL, C++ + MS Access и др. Благодаря использованию технологии веб-сервисов, подобной зависимости можно избежать, так как к веб-сервису можно обращаться с клиентов любых программно-аппаратных платформ. С другой стороны, из веб-сервиса возможно обращаться к СУБД различных типов, используя технологию ADO .NET.

ADO.NET представляет собой набор библиотек, входящих в Microsoft .NET Framework и предназначенных для взаимодействия с различными хранилищами данных из .NET-приложений. Библиотеки ADO.NET включают все необходимые классы для подключения к источникам данных практически произвольного формата, выполнения запросов к этим источникам и получения результата. Кроме того, несомненным достоинством ADO.NET является возможность работы с отсоединенными источниками данных, представляющими собой структуры, которые организуют данные в оперативной памяти компьютера и работать с которыми возможно с применением ставших уже привычными средств доступа к данным. Таким образом, ADO.NET можно использовать в качестве надежного, иерархически организованного отсоединенного от источника кэша данных для автономной работы, что незаменимо при построении масштабируемых приложений. Применение этой технологии позволяет в значительной мере решить проблему согласования клиента и СУБД различных платформ.

Небезопасность авторизации и передачи данных возможно устранить при использовании шифрования данных авторизации, а также самого потока передачи запроса на обращение к хранилищу данных. Случайное комбинирование хотя бы нескольких различных алгоритмов шифрования в этом вопросе позволит существенно повысить уровень безопасности взаимодействия. Также на пользу идёт факт логического и/или пространственного разделения области работы приложения веб-сервиса и СУБД. Благодаря этому, клиент может обращаться с запросом к интерфейсу веб-сервиса, расположенному на одном узле, а веб-сервис, в свою очередь, переадресовывать запрос на удалённый узел (один или группу, в зависимости от наличия распределённости СУБД). Это неизбежно снизит быстродействие работы с базой данных для клиента, однако поспособствует сохранности передаваемой информации.

### 3. Постановка задачи

Из описания существующей ситуации можно сделать вывод о необходимости создания веб-сервиса для удалённого взаимодействия клиентских приложений с хранилищами данных. Этот веб-сервис может быть публичным и обладать системой разделения прав пользователей на доступ и операции с данными.

В качестве платформы разработки самого веб-сервиса было принято решение использовать Microsoft .NET [2]. Это обусловлено, в первую очередь, простотой реализации, а также широкими возможностями, предоставляемыми WSDL веб-сервисами: возможность использования как HTTP, так и SOAP протокола для внешнего взаимодействия; наличие wsdl-описания, что позволяет без труда сделать сервис доступным из репозитория, а также получить перечень веб-методов, доступных для вызова клиентским приложением.

С другой стороны, веб-сервис будет взаимодействовать с удалённой или же локальной БД под управлением СУБД MS SQL Server, как наиболее нативной для платформ MS, в целом. Как уже было упомянуто ранее, это не будет иметь никакого значения в плане особенностей организации взаимодействия клиентского приложения с веб-сервисом.

Выдвинуты следующие требования к реализуемому сервису:

- Возможность взаимодействия с локальной или удалённой СУБД MS SQL Server.
- Наличие функций для работы с данными (создание, добавление, обновление, удаление таблиц, кортежей).
- Наличие системы пользовательских аккаунтов.
- Разграничение прав пользователей на доступ к данным.
- Шифрование данных авторизации, а также передаваемого потока посредством комбинированного использования нескольких алгоритмов.
- Доступность сервиса и его WSDL-описания в UDDI репозиториях.

### 4. Программная реализация

В соответствии с описанной задачей и поставленными требованиями был реализован веб-сервис удаленного доступа к БД. Данный сервис построен на базе платформы Microsoft ASP .NET и применяет протокол SOAP XML для взаимодействия с клиентом. В качестве СУБД использована Microsoft SQL Server [3], взаимодействие с которой производится через объекты отсоединенных источников данных.

Схема взаимодействия веб-сервиса с клиентским приложением и с СУБД имеет следующий вид (рис. 1):

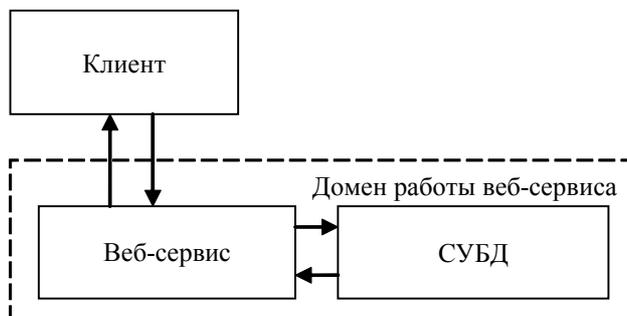


Рис. 1. Схема работы веб-сервиса

Схема взаимодействия также показывает строгое разделение домена веб-сервиса и клиентского приложения. Таким образом, непосредственный доступ к СУБД является инкапсулированным относительно клиентского приложения.

Классовая иерархия программного продукта представлена на рис. 2.

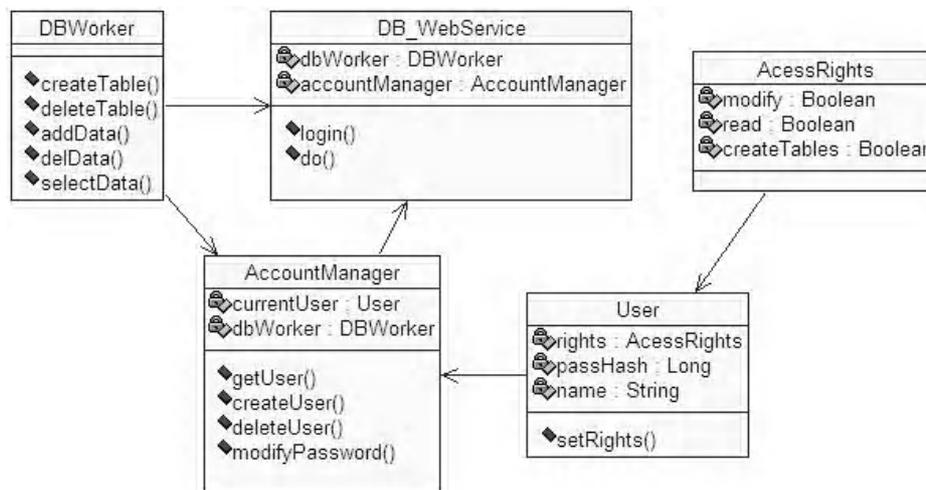


Рис. 2. UML-диаграмма классов веб-сервиса

В соответствии с разработанной структурой, с клиентской стороны доступны два метода веб-сервиса (их прототипы и аннотации представлены в WSDL-спецификации): `int login(string name, string pass)`, возвращающий клиенту идентификатор сессии для работы с базой данных (по умолчанию, сессия действительна в течение 20 минут от момента логина); `string do(string action, object[] data)` – служит для выполнения действия `action` с базой данных (перечень доступных действий, а также их описание доступны в WSDL-аннотации) и набором данных `data`. В результате работы метод возвращает данные в виде xml-строки - ответа СУБД. Регистрация пользователя в системе осуществляется также либо с применением этого метода, либо локально (на стороне веб-сервиса).

Основной класс веб-сервиса (`DB_WebService`, наследник класса `WebService`) также использует для своей работы агрегированные объекты-экземпляры вспомогательных классов: менеджера работы с аккаунтами (`AccountManager`), а также объект для доступа к БД через отсоединенные источники данных (`DBWorker`).

## 5. Выводы

По причине существующей проблемы взаимодействия клиентских приложений с удалёнными СУБД была проанализирована предметная область и поставлена задача разработать веб-сервис для удалённой работы с базами данных. Веб-сервис реализован на платформе ASP .NET, работает по протоколу SOAP XML и удовлетворяет основным требованиям, обозначенным при рассмотрении задачи: простота взаимодействия, платформонезависимость и безопасность.

Одним из направлений дальнейших исследований является разработка веб-сервиса, позволяющего организовывать двухстороннее взаимодействие: как с клиентами на разных платформах – с одной стороны, так и с различными СУБД – с другой.

**Список литературы:** 1. Фуфаев Э.В. и др. Разработка и эксплуатация удалённых баз данных. М.: Академия, 2008. 256с. 2. Шапошников И. Web-сервисы Microsoft .NET. СПб.: БХВ-Петербург, 2002. 334с. 3. Хендерсон К. Microsoft SQL Server: структура и реализация. Профессиональное руководство. М.: Вильямс, 2005. 1056с. 4. Шилдт Г. Полный справочник по C#. Пер. с англ. М.: Вильямс, 2004. 752с.

Поступила в редколлегию 26.05.2010

**Корниловский Артём Викторович**, студент специальности интеллектуальные системы принятия решений, факультет КН, ХНУРЭ. Адрес: Украина, 61019, Харьков, пр. Ильича, 103а, кв. 40, тел. 376-27-27.

**Губин Вадим Александрович**, ст. преподаватель кафедры искусственного интеллекта ХНУРЭ. Научные интересы: интернет-технологии, интеллектуальный анализ данных. Адрес: Украина, 61054, Харьков, ул. Гв.-Широнинцев, 23, кв. 287, тел. 710-64-12.