

# Высокоскоростные асинхронные методы обмена данными для информационных систем

Авторы: В.И. Анисимов,  
Г.Д. Дмитриевич, С.А. Васильев

# ВВЕДЕНИЕ

**Аспекты информационной системы позволяющие расширить её внутренние и внешние коммуникационные возможности:**

- Сервис-ориентированная архитектура (SOA) - стандартизированные интерфейсы и протоколы обмена информацией, абстрагированные от языков программирования, сред разработки и самих разработчиков.
- Асинхронные методы обмена данными - интерактивной работы сервисов (WebSocket)
- Платформа разработки WCF – безопасность и простота использования

# Реализация SOA на основе WCF и WebSocket

- Стандартная парадигма WCF определяет два интерфейса содержащие методы: принимающие и отправляющий сообщения клиенту.
- WCF платформа позволяет использовать асинхронные методы **ВЫЗОВОВ**
- Новая привязкой `NetHttpBinding`, использует двоично-кодированные SOAP-сообщения через протокол WebSocket (S)

# Реализация коммуникационного модуля серверной составляющей

```
[ServiceBehavior(ConcurrencyMode =  
ConcurrencyMode.Reentrant)]
```

```
public class WebSocketService : IDuplexContract
```

```
{public string SayMessageDuplex(string name
```

```
{OperationContext.Current.
```

```
GetCallbackChannel<IDuplexCallbackContract>().
```

```
SayingMessage(передаваемая информация);
```

```
return передаваемая информация; }
```

```
}
```

```
[ServiceContract(CallbackContract=  
typeof(IDuplexCallbackContract))]
```

```
public interface IDuplexContract
```

```
{[OperationContract]
```

```
string SayMessageDuplex(string name);}
```

```
[ServiceContract]
```

```
public interface IDuplexCallbackContract
```

```
{[OperationContract]
```

```
void SayingMessge(string message);}
```

# Настройка конфигурации серверной составляющей

Привязка `NetHttpBinding` использует протокол HTTP в качестве транспорта для отправки сообщений. При использовании с дуплексным контрактом будут использоваться `WebSocket`.

```
<baseAddresses>
```

```
<add baseAddress="http://localhost:8085/" />
```

```
</baseAddresses>
```

```
<endpoint address="http://localhost:8082" binding="netHttpBinding"  
bindingConfiguration="TextOverWebSockets" contract="Contracts.IDuplexContract" />
```

```
<binding name="RequestResponseWithWebSockets">
```

```
<websocketSettings transportUsage="Always" />
```

```
</binding>
```

```
<binding name="TextOverWebSockets" messageEncoding="Text">
```

```
</binding>
```

# Клиентский коммуникационный модуль

- **Получение WSDL файла конфигурации возможно при обращении по базовому адресу:**

<http://localhost:8085/>

- **При добавление ссылки на службу привязка будет сформирована автоматически:**

При этом схема обращения будет использоваться ws://

# Пример выполнения клиент-серверного приложения в среде WCF

WCF WebSocket/Service/Host/bin/Debug/Host.EXE

```
Service is opened, press Enter to stop  
Address: http://localhost:8082/  
Contact: IDuplexContract  
Binding: NetHttpBinding  
WebSockets mode: WhenDuplex  
WebSockets subProtocol: soap
```

WCF WebSocket/Client/Client/bin/Debug/Client.EXE

```
Press enter when service is ready  
Вызов двустороннего контракта с использованием текстовых сообщений:  
Hello WebSocket client №1 by WebSockets ответ 1  
ответ сервера 1: Hello WebSocket client №1 номер обращения: 1
```

# Особенности создания информационных систем

- Все данные, переданные в соединении WCF WebSocket, находятся в формате SOAP (двоичный или текстовый).
- Чтобы связаться с сервисом WCF WebSocket с кодом JavaScript на веб-странице, требуется самостоятельно разобрать SOAP.
- Передача сложноструктурированных данных (многомерные массивы) возможна посредством протокола SOAP, но имеет различную интерпретацию при работе в кроссплатформенных системах



# ВЫВОДЫ

В соответствии с определением веб-службы методика обеспечивает кроссплатформенное взаимодействие, основанное на базе открытых стандартов и протоколов (SOAP, WSDL), а также полную инфраструктурную поддержку со стороны разработчиков всех современных популярных браузеров для создания клиентских веб-приложений.

## **Преимущества информационной системы достигаемой за счет применения SOA:**

масштабируемость и гибкость за счёт возможности изменения, добавления слабосвязанных компонентов системы

## **Преимущества информационной системы достигаемой за счет применения протокола WebSocket:**

увеличение производительность за счет скорости обмена данными и интерактивность, за счёт асинхронного обмена данными со внешней и внутренней средой

**WCF платформа позволяет быстро и безопасно реализовать описанные технологии**