



Описание веб-сервиса АРМ ПК «Энергия»

2015

Настоящий документ представляет собой техническое описание веб-сервиса АРМ ПК «Энергия». Веб-сервис АРМ является внешним интерфейсом, с помощью которого субъекты/объекты электроэнергетики могут взаимодействовать с ПК «Энергия», осуществляя просмотр и ввод фактической информации.

Оглавление

1. Термины и определения	4
2. Назначение	5
3. Расположение веб-сервиса АРМ	6
4. Клиентский прокси-класс (.NET)	7
5. Операции	8
5.1. ValidateUser	8
5.1.1. Функциональность	8
5.1.2. Параметры	8
5.1.3. Пример входящего soap-сообщения	8
5.1.4. Пример возвращаемого soap-сообщения	8
5.2. GetData	9
5.2.1. Функциональность	9
5.2.2. Параметры	9
5.2.3. Пример входящего soap-сообщения	9
5.2.4. Пример возвращаемого soap-сообщения	9
5.3. SaveData	10
5.3.1. Функциональность	10
5.3.2. Параметры	10
5.3.3. Пример входящего soap-сообщения	10
6. Порядок работы	12
7. Типы данных	13
7.1. ValidateUserRequest	13
7.2. ValidateUserResponse	13
7.3. ArmRequest	13
7.4. ArmSaveRequest	13
7.5. ArmResponse	14
7.6. ArmFactor	14
7.7. ArmValue	15
7.8. ArmEntity	15
8. WSDL описание веб-сервиса АРМ	16
9. Особенности использования веб сервиса	17

1. Термины и определения

ПК	Программный Комплекс
АРМ	Автоматизированное рабочее место
ПК «Энергия»	Программное обеспечение «Автоматизированная система сбора, достоверизации и формирования плановой и оперативной, отчетной информации»
Файл WSDL	Документ XML, составленный согласно XML-грамматике, называемой языком описания веб-служб (WSDL). Этот файл определяет поведение XML-веб-службы и указывает клиентам правила взаимодействия с ней.

2. Назначение

Веб-сервис АРМ предоставляет внешним организациям (субъектам/объектам электроэнергетики) возможность просмотра и ввода фактических данных в ПК «Энергия».

3. Расположение веб-сервиса АРМ

По умолчанию веб-сервис АРМ доступен в ПК «Энергия» по следующему url-адресу:

[https:// ws.so-ups.ru:8500/Energy2010/webservices/armservice.asmx](https://ws.so-ups.ru:8500/Energy2010/webservices/armservice.asmx)

Расположение веб-сервиса АРМ и параметры доступа могут быть изменены в связи с настройками безопасности.

4. Клиентский прокси-класс (.NET)

Клиентский прокси-класс (со всеми необходимыми элементами) доступен в *Описание веб-сервиса АРМ_Proxy.cs*

5. Операции

Список всех операций веб-сервиса АРМ представлен в таблице ниже.

Операция	Описание
GetData	Возвращает данные для АРМ.
SaveData	Сохраняет данные от АРМ.
ValidateUser	Проверяет пользователя АРМ (на валидность).

5.1. ValidateUser

5.1.1. Функциональность

Осуществляет проверку учетных данных (логин и пароль) пользователя АРМ.

5.1.2. Параметры

Название	Тип	Описание
Входные параметры		
request	ValidateUserRequest	Запрос на проверку пользователя АРМ.
Выходные параметры		
	ValidateUserResponse	Результат проверки пользователя АРМ.

5.1.3. Пример входящего soap-сообщения

```
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" >
  <soap:Body>
    <ValidateUser xmlns="http://tempuri.org/" >
      <ValidateUserRequest>
        <Login>Логин пользователя АРМ (незашифрованный)</Login>
        <Password>Пароль пользователя АРМ (незашифрованный)</Password>
      </ValidateUserRequest>
    </ValidateUser>
  </soap:Body>
</soap:Envelope>
```

5.1.4. Пример возвращаемого soap-сообщения

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" >
  <soap:Body>
```



```
<ValidateUserResponse xmlns="http://tempuri.org/">
  <ValidateUserResponse>
    <IsValid>Результат проверки учетных данных (true или false)</IsValid>
  </ValidateUserResponse>
</ValidateUserResponse>
</soap:Body>
</soap:Envelope>
```

5.2. GetData

5.2.1. Функциональность

Осуществляет получение данных для АРМ за указанные операционные сутки и тип данных (оперативные или месячные).

5.2.2. Параметры

Название	Тип	Описание
Входные параметры		
request	ArmRequest	Запрос на получение данных для АРМ.
Выходные параметры		
	ArmResponse	Ответ с данными АРМ для указанных в запросе параметров.

5.2.3. Пример входящего soap-сообщения

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetData xmlns="http://tempuri.org/">
      <ArmRequest>
        <Login>Логин пользователя АРМ (незашифрованный).</Login>
        <OperationDate>dateTime</OperationDate>
        <Password>Пароль пользователя АРМ (незашифрованный).</Password>
        <Variant>Тип данных - оперативные или уточненные
(месячные) (OperationalData(=1)/MonthlyData(=4))</Variant>
      </ArmRequest>
    </GetData>
  </soap:Body>
</soap:Envelope>
```

5.2.4. Пример возвращаемого soap-сообщения

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
```

```

<soap:Body>
  <GetDataResponse xmlns="http://tempuri.org/">
    <ArmResponse>
      <Factors>
        <ArmFactor>
          <EntityId>Идентификатор энергообъекта</EntityId>
          <FactorId>Идентификатор показателя</FactorId>
          <Name>Название показателя</Name>
          <PreventNegative>Предотвращать ввод отрицательных значений
          </PreventNegative>
        </ArmFactor>
        ...
      </Factors>
      <Data>
        <ArmEntity>
          <Values xsi:nil="true" />
          ...
        </ArmEntity>
        ...
      </Data>
      <CloseGateDateLocal>Дата закрытия ворот по местному
      времени</CloseGateDateLocal>
      <CloseGateDateMsk>Дата закрытия ворот по московскому
      времени</CloseGateDateMsk>
      <IsCorrection>Признак что полученные данные являются
      корректировкой</IsCorrection>
    </ArmResponse>
  </GetDataResponse>
</soap:Body>
</soap:Envelope>

```

5.3. SaveData

5.3.1. Функциональность

Осуществляет сохранение данных, измененных пользователем АРМ. В процессе сохранения выполняются также проверки – контроль данных.

5.3.2. Параметры

Название	Тип	Описание
Входные параметры		
request	ArmSaveRequest	Запрос на сохранение данных АРМ.

5.3.3. Пример входящего soap-сообщения

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SaveData xmlns="http://tempuri.org/">

```

```
<ArmSaveRequest>
  <Data>
    <ArmEntity>
      <Values xsi:nil="true" />
    </ArmEntity>
    ...
  </Data>
</ArmSaveRequest>
</SaveData>
</soap:Body>
</soap:Envelope>
```

6. Порядок работы

При работе с веб-сервисом АРМ желательно придерживаться следующего логического порядка вызова методов:

1. **ValidateUser** – проверяем корректность используемых учетных данных (логин и пароль). Попытка вызова методов **GetData** или **SaveData** с некорректными учетными данными приведет к возникновению исключения.
2. **GetData** – получаем значения показателей по «нашим» энергообъектам/показателям за указанные операционные сутки и тип данных (оперативные/месячные). Учетные данные определяют - какой набор энергообъектов/показателей нам доступен. Разрешение на доступ пользователю АРМ к энергообъектам/показателям задаются в РДУ/ОДУ/ИА.
3. **SaveData** - на основе полученных на шаге 2 данных, можно сформировать запрос на сохранение данных.

Вызов метода **GetData** помимо данных возвращает также дату и время закрытия ворот. Необходимо учитывать, что сохранение данных до закрытия ворот приведет к немедленному сохранению данных и будет видно на уровне РДУ. Сохранение данных после закрытия ворот приведет к созданию «корректировки», которая будет отдельно рассматриваться РДУ. Корректировка может быть как принята, так и отклонена. Принятие корректировки приведет к реальному сохранению данных, отправленных АРМ после закрытия ворот. Отклонение корректировки приведет к отмене данных, введенных АРМ после закрытия ворот, и восстановлению их предыдущего последнего значения.

7. Типы данных

7.1. ValidateUserRequest

Представляет собой запрос на проверку пользователя АРМ.

Свойство	Описание
Login	Логин пользователя АРМ (незашифрованный).
Password	Пароль пользователя АРМ (незашифрованный).

7.2. ValidateUserResponse

Представляет собой ответ с результатами проверки пользователя АРМ.

Свойство	Описание
IsValid	Признак существования пользователя АРМ с учетными данными указанными в ValidateUserRequest.

7.3. ArmRequest

Представляет собой запрос на получение данных для АРМ.

Свойство	Описание
Login	Логин пользователя АРМ (незашифрованный).
Password	Пароль пользователя АРМ (незашифрованный).
OperationDate	Операционные сутки, за которые требуется получить данные. Для оперативных данных - это дата с точностью до дня (dd.ММ.уууу). Для уточненных данных – это дата с точностью до месяца, т.е. первое число месяца (01.ММ.уууу).
Variant	Вариант представления данных. Т.е. тип данных, которые необходимо получить. Значение: <ul style="list-style-type: none"> • OperationalData (=1) – оперативные данные; • MonthlyData (=4) – уточненные (месячные) данные.

7.4. ArmSaveRequest

Представляет собой запрос на сохранение данных АРМ.

Свойство	Описание
Login	Логин пользователя АРМ (незашифрованный).

Password	Пароль пользователя АРМ (незашифрованный).
OperationDate	Операционные сутки, за которые требуется получить данные. Для оперативных данных - это дата с точностью до дня (dd.ММ.уууу). Для уточненных данных – это дата с точностью до месяца, т.е. первое число месяца (01.ММ.уууу).
Variant	Вариант представления данных. Т.е. тип данных, которые необходимо получить. Значение: <ul style="list-style-type: none"> • OperationalData (=1) – оперативные данные; • MonthlyData (=4) – уточненные (месячные) данные.
Data	Сохраняемые данные (по энергообъектам и показателям).

7.5. ArmResponse

Представляет собой ответ с данными для АРМ.

Свойство	Описание
Factors	Список показателей, по которым представлены данные.
Data	Список энергообъектов со значениями показателей.
CloseGateDateLocal	Локальные дата и время закрытия ворот для АРМ для указанных в запросе параметров (тип данных и операционные сутки). Дата и время представлены в часовом поясе, который указан в настройках данного АРМ в ПК «Энергия».
CloseGateDateMsk	Дата и время закрытия ворот для АРМ для указанных в запросе параметров (тип данных и операционные сутки). Дата и время представлены в московском часовом поясе.
IsCorrection	Признак сохранения данных как корректировки.

7.6. ArmFactor

Представляет собой сведения об отдельном показателе.

Свойство	Описание
EntityId	Уникальный идентификатор энергообъекта. Каждый энергообъект может иметь свой набор показателей доступных АРМ для заполнения.
FactorId	Уникальный идентификатор показателя.
Name	Название показателя.
PreventNegative	Признак необходимости предотвращения ввода отрицательных значений показателя.

7.7. ArmValue

Представляет собой значение отдельного показателя или значение корректировки по показателю.

Свойство	Описание
Id	Уникальный идентификатор значения АРМ.
Value	Значение показателя/корректировки.
FactorId	Уникальный идентификатор показателя.
EntityId	Уникальный идентификатор энергообъекта.
ValueAbsPt	Значение для расчета нарастающего итога для АРМ: <ul style="list-style-type: none"> • Для уточненных данных это значение за предыдущий месяц. • Для оперативных данных это значение за предыдущий день либо 0 (в зависимости от типа показателя и дня месяца).
CorrectionId	Идентификатор корректировки (вспомогательное поле).
IsPending	Признак, указывающий, что значение является корректировкой и до сих пор не принято в РДУ.

7.8. ArmEntity

Представляет собой данные по отдельному энергообъекту.

Свойство	Описание
Id	Уникальный идентификатор энергообъекта.
Code	Уникальный код энергообъекта.
Name	Название энергообъекта.
Values	Список значений показателей.

8. WSDL описание веб-сервиса АРМ

WSDL находится в отдельном файле *Описание веб-сервиса АРМ_ArmService.wsdl*

9. Особенности использования веб сервиса

При обращении к веб сервису АРМ используется Forms authentication. Ниже описаны основные

- Создание прокси класса. Необходимо явно указать имя пользователя и пароль АРМа.

```
ArmServiceSoapClient srv = new ArmServiceSoapClient("ArmServiceSoap",
"https://62.33.248.120:8500/WebServices/ArmService.asmx");
    srv.ClientCredentials.UserName.UserName = "eng*****";
    srv.ClientCredentials.UserName.Password = "lgg*****";
```

- В запросе конкретного метода логин и пароль также необходимо указывать

```
ValidateUserRequest request = new ValidateUserRequest()
{
    Login = "eng*****",
    Password = "lgg*****";
};
```

- В конфигурационный файл необходимо добавить секцию *system.serviceModel*

```
<system.serviceModel>
  <bindings>
    <basicHttpBinding>
      <binding name="ArmServiceSoap" closeTimeout="00:01:00" openTimeout="00:01:00" receiveTimeout="00:10:00"
sendTimeout="00:01:00" allowCookies="false" bypassProxyOnLocal="false" hostNameComparisonMode="StrongWildcard"
maxBufferSize="2147483647" maxBufferPoolSize="2147483647" maxReceivedMessageSize="2147483647" messageEncoding="Text"
textEncoding="utf-8" transferMode="Buffered" useDefaultWebProxy="true">
      <readerQuotas maxDepth="2147483647" maxStringContentLength="2147483647" maxArrayLength="2147483647"
maxBytesPerRead="2147483647" maxNameTableCharCount="50000000" />
      <security mode="Transport">
        <transport clientCredentialType="Basic" />
      </security>
    </binding>
  </basicHttpBinding>
</bindings>
<client>
  <endpoint address="http://localhost:50000/WebServices/ArmService.asmx"
binding="basicHttpBinding"
bindingConfiguration="ArmServiceSoap"
contract="ArmService.ArmServiceSoap"
name="ArmServiceSoap"
behaviorConfiguration="ArmServiceEndpointBehavior" />
</client>
<behavior>
  <endpointBehaviors>
    <behavior name="ArmServiceEndpointBehavior">
      <httpClientIP />
      <authHeader />
    </behavior>
  </endpointBehaviors>
</behavior>
<extensions>
  <behaviorExtensions>
    <add name="httpClientIP" type="EnerSys.Energy.enARM.Hel pers.HttpClientIPBehaviorExtensionElement,
EnerSys.Energy.enARM" />
    <add name="authHeader" type="EnerSys.Energy.enARM.Hel pers.HttpAuthHeaderBehaviorExtensionElement,
EnerSys.Energy.enARM" />
  </behaviorExtensions>
</extensions>
</system.serviceModel>
```

- Необходимо добавить реализацию расширения прокси класса для добавления дополнительной информации по безопасности (ниже примеры на C#). Особое внимание необходимо обратить на *HttpAuthHeaderMessageInspector*, в котором в примере явно

указывается логин и пароль. Его необходимо брать из основного приложения или конфигурационного файла.

```
/// <summary>
/// Класс, позволяющий подключать расширение функциональности клиентской прокси веб-сервиса в конфиге.
/// </summary>
public class HttpAuthHeaderBehaviorExtensionElement : BehaviorExtensionElement
{
    /// <summary>
    /// Gets the type of behavior.
    /// </summary>
    public override Type BehaviorType
    {
        get
        {
            return typeof(HttpAuthHeaderEndpointBehavior);
        }
    }

    /// <summary>
    /// Creates a behavior extension based on the current configuration settings.
    /// </summary>
    /// <returns>The behavior extension. </returns>
    protected override object CreateBehavior()
    {
        return new HttpAuthHeaderEndpointBehavior();
    }
}

/// <summary>
/// Класс, расширяющий функциональность клиентской прокси веб-сервиса, для добавления заголовка "Authorization:
Basic".
/// </summary>
public class HttpAuthHeaderEndpointBehavior : IEndpointBehavior
{
    #region Конструктор

    /// <summary>
    /// Конструктор по умолчанию.
    /// </summary>
    public HttpAuthHeaderEndpointBehavior()
    {
    }

    #endregion Конструктор

    #region Методы

    /// <summary>
    /// Implement to pass data at runtime to bindings to support custom behavior.
    /// </summary>
    /// <param name="endpoint">The endpoint to modify. </param>
    /// <param name="bindingParameters">The objects that binding elements require to support the
behavior. </param>
    public void AddBindingParameters(ServiceEndpoint endpoint,
System.ServiceModel.Channels.BindingParameterCollection bindingParameters)
    {
    }

    /// <summary>
    /// Implements a modification or extension of the client across an endpoint.
    /// </summary>
    /// <param name="endpoint">The endpoint that is to be customized. </param>
    /// <param name="clientRuntime">The client runtime to be customized. </param>
    public void ApplyClientBehavior(ServiceEndpoint endpoint, System.ServiceModel.Dispatcher.ClientRuntime
clientRuntime)
    {
        HttpAuthHeaderMessageInspector inspector = new HttpAuthHeaderMessageInspector();
        clientRuntime.MessageInspectors.Add(inspector);
    }

    /// <summary>
    /// Implements a modification or extension of the service across an endpoint.
    /// </summary>
    /// <param name="endpoint">The endpoint that exposes the contract. </param>
    /// <param name="endpointDispatcher">The endpoint dispatcher to be modified or extended. </param>

```

```

        public void ApplyDispatchBehavior(ServiceEndpoint endpoint,
        System.ServiceModel.Dispatcher.EndpointDispatcher endpointDispatcher)
        {
            // <summary>
            // Implement to confirm that the endpoint meets some intended criteria.
            // </summary>
            // <param name="endpoint">The endpoint to validate.</param>
            public void Validate(ServiceEndpoint endpoint)
            {
            }

            #endregion Методы
        }

// <summary>
// Хелперный класс для добавления заголовка "Authorization: Basic" к запросу веб-сервиса.
// </summary>
public class HttpAuthHeaderMessageInspector : IClientMessageInspector
{
    #region Конструктор

    // <summary>
    // Конструктор по умолчанию.
    // </summary>
    public HttpAuthHeaderMessageInspector()
    {
    }

    #endregion Конструктор

    #region IClientMessageInspector Members

    // <summary>
    // Enables inspection or modification of a message after a reply message is
    // received but prior to passing it back to the client application.
    // </summary>
    // <param name="reply">The message to be transformed into types and handed back to the client
    application.</param>
    // <param name="correlationState">Correlation state data.</param>
    public void AfterReceiveReply(ref Message reply, object correlationState)
    {
    }

    // <summary>
    // Enables inspection or modification of a message before a request message is sent to a service.
    // </summary>
    // <param name="request">The message to be sent to the service.</param>
    // <param name="channel">The client object channel.</param>
    // <returns>
    // The object that is returned as the correlationState argument of the
    System.ServiceModel.Dispatcher.IClientMessageInspector.AfterReceiveReply(System.ServiceModel.Channels.Message@, System.Obj
    ect)
    // method. This is null if no correlation state is used. The best practice is
    // to make this a System.Guid to ensure that no two correlationState objects are the same.
    // </returns>
    public object BeforeSendRequest(ref Message request, IClientChannel channel)
    {
        string userName = "eng*****";
        string password = "Igg*****";

        string authorizationHeader = "Basic " +
        Convert.ToBase64String(Encoding.ASCII.GetBytes(userName + ":" + password));

        HttpRequestMessageProperty httpRequestMessage;
        object httpRequestMessageObject;

        if (request.Properties.TryGetValue(HttpRequestMessageProperty.Name, out
        httpRequestMessageObject))
        {
            httpRequestMessage = (HttpRequestMessageProperty)httpRequestMessageObject;

            if
            (string.IsNullOrEmpty(httpRequestMessage.Headers[HttpRequestHeader.Authorization]))
            {
                httpRequestMessage.Headers[HttpRequestHeader.Authorization] =
                authorizationHeader;
            }
            else
            {
                httpRequestMessage = new HttpRequestMessageProperty();
            }
        }
    }
}

```

```

        httpRequestMessage.Headers[HttpRequestHeader.Authorization] = authorizationHeader;
        request.Properties.Add(HttpRequestMessageProperty.Name, httpRequestMessage);
    }

    return null;
}

#endregion IClientMessageInspector Members

/// <summary>
/// Класс, позволяющий подключать расширение функциональности клиентской прокси веб-сервиса в конфиге.
/// </summary>
public class HttpClientIpBehaviorExtensionElement : BehaviorExtensionElement
{
    /// <summary>
    /// Gets the type of behavior.
    /// </summary>
    public override Type BehaviorType
    {
        get
        {
            return typeof(HttpClientIpEndpointBehavior);
        }
    }

    /// <summary>
    /// Creates a behavior extension based on the current configuration settings.
    /// </summary>
    /// <returns>The behavior extension. </returns>
    protected override object CreateBehavior()
    {
        return new HttpClientIpEndpointBehavior();
    }
}

/// <summary>
/// Класс, расширяющий функциональность клиентской прокси веб-сервиса, для добавления X-Client-IP.
/// </summary>
public class HttpClientIpEndpointBehavior : IEndpointBehavior
{
    #region Конструктор

    /// <summary>
    /// Конструктор по умолчанию.
    /// </summary>
    public HttpClientIpEndpointBehavior()
    {
    }

    #endregion Конструктор

    #region Методы

    /// <summary>
    /// Implement to pass data at runtime to bindings to support custom behavior.
    /// </summary>
    /// <param name="endpoint">The endpoint to modify. </param>
    /// <param name="bindingParameters">The objects that binding elements require to support the
behavior. </param>
    public void AddBindingParameters(ServiceEndpoint endpoint,
System.ServiceModel.Channels.BindingParameterCollection bindingParameters)
    {
    }

    /// <summary>
    /// Implements a modification or extension of the client across an endpoint.
    /// </summary>
    /// <param name="endpoint">The endpoint that is to be customized. </param>
    /// <param name="clientRuntime">The client runtime to be customized. </param>
    public void ApplyClientBehavior(ServiceEndpoint endpoint, System.ServiceModel.Dispatcher.ClientRuntime
clientRuntime)
    {
        HttpClientIpMessageInspector inspector = new HttpClientIpMessageInspector();
        clientRuntime.MessageInspectors.Add(inspector);
    }

    /// <summary>
    /// Implements a modification or extension of the service across an endpoint.
    /// </summary>
    /// <param name="endpoint">The endpoint that exposes the contract. </param>
    /// <param name="endpointDispatcher">The endpoint dispatcher to be modified or extended. </param>
    public void ApplyDispatchBehavior(ServiceEndpoint endpoint,
System.ServiceModel.Dispatcher.EndpointDispatcher endpointDispatcher)
    {
    }
}

```

```

    /// <summary>
    /// Implement to confirm that the endpoint meets some intended criteria.
    /// </summary>
    /// <param name="endpoint">The endpoint to validate. </param>
    public void Validate(ServiceEndpoint endpoint)
    {
    }

    #endregion Методы
}

/// <summary>
/// Хелперный класс для добавления заголовка X-Client-IP к запросу веб-сервиса.
/// </summary>
public class HttpClientMessageInspector : IClientMessageInspector
{
    #region Константы

    /// <summary>
    /// Название заголовка X-Client-IP.
    /// </summary>
    private const string X_CLIENT_IP_HEADER = "X-Client-IP";

    #endregion Константы

    #region Конструктор

    /// <summary>
    /// Конструктор по умолчанию.
    /// </summary>
    public HttpClientMessageInspector()
    {
    }

    #endregion Конструктор

    #region IClientMessageInspector Members

    /// <summary>
    /// Enables inspection or modification of a message after a reply message is
    /// received but prior to passing it back to the client application.
    /// </summary>
    /// <param name="reply">The message to be transformed into types and handed back to the client
    application. </param>
    /// <param name="correlationState">Correlation state data. </param>
    public void AfterReceiveReply(ref System.ServiceModel.Channels.Message reply, object correlationState)
    {
    }

    /// <summary>
    /// Enables inspection or modification of a message before a request message is sent to a service.
    /// </summary>
    /// <param name="request">The message to be sent to the service. </param>
    /// <param name="channel">The client object channel. </param>
    /// <returns>
    /// The object that is returned as the correlationState argument of the
    System.ServiceModel.Dispatcher.IClientMessageInspector.AfterReceiveReply(System.ServiceModel.Channels.Message@, System.Obj
    ect)
    /// method. This is null if no correlation state is used. The best practice is
    /// to make this a System.Guid to ensure that no two correlationState objects are the same.
    /// </returns>
    public object BeforeSendRequest(ref System.ServiceModel.Channels.Message request,
    System.ServiceModel.IClientChannel channel)
    {
        if (HttpContext.Current != null && HttpContext.Current.Request != null &&
        !string.IsNullOrEmpty(HttpContext.Current.Request.UserHostAddress))
        {
            HttpRequestMessageProperty httpRequestMessage;
            object httpRequestMessageObject;

            if (request.Properties.TryGetValue(HttpRequestMessageProperty.Name, out
            httpRequestMessageObject))
            {
                httpRequestMessage = (HttpRequestMessageProperty)httpRequestMessageObject;

                if (string.IsNullOrEmpty(httpRequestMessage.Headers[X_CLIENT_IP_HEADER]))
                {
                    httpRequestMessage.Headers[X_CLIENT_IP_HEADER] =
                    HttpContext.Current.Request.UserHostAddress;
                }
            }
            else
            {

```

```
        httpRequestMessage = new HttpRequestMessageProperty();
        httpRequestMessage.Headers[X_CLIENT_IP_HEADER] =
HttpContext.Current.Request.UserHostAddress;
        httpRequestMessage.Properties.Add(HttpRequestMessageProperty.Name,
        httpRequestMessage);
    }
    }
    return null;
}
#endregion IClientMessageInspector Members
}
```