

SOAP

SOAP — это протокол, по которому веб-сервисы взаимодействуют друг с другом или с клиентами. Название происходит от сокращения Simple Object Access Protocol («простой протокол доступа к объектам»). SOAP API — это веб-сервис, использующий протокол SOAP для обмена сообщениями между серверами и клиентами. При этом сообщения должны быть написаны на языке XML в соответствии со строгими стандартами, иначе сервер вернет ошибку.

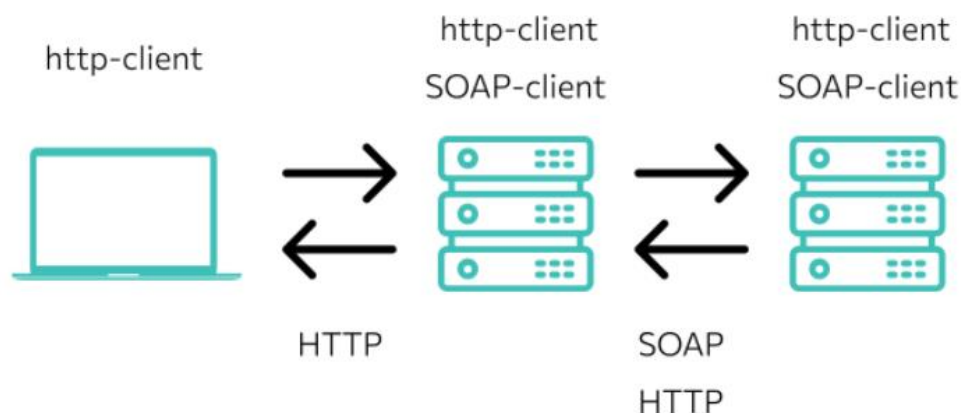


Схема взаимодействия веб-приложений по протоколу SOAP

Протокол SOAP был представлен в 1998 году и быстро стал одним из главных стандартов веб-служб, когда Microsoft продвигала платформу .NET, приложения которой взаимодействовали с помощью SOAP API. Сейчас протокол и API уступают по популярности архитектурному стилю REST. Но веб-приложения, использующие SOAP API, все еще пользуются спросом, особенно в банковском и телекоммуникационном секторах.

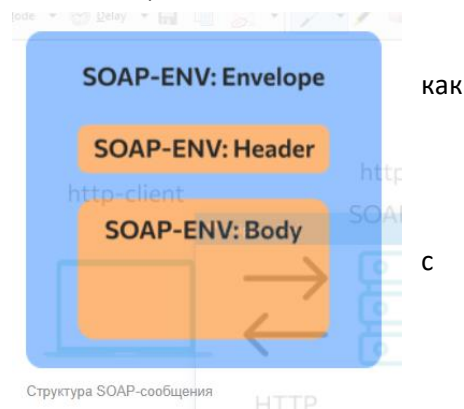
Особенности SOAP API

SOAP может использоваться с протоколами SMTP, FTP, HTTP, HTTPS. Чаще всего — с HTTP как с наиболее универсальным: его поддерживают все браузеры и серверы. Корректное SOAP-сообщение состоит из нескольких структурных элементов: Envelope, Header, Body и Fault.

Envelope («конверт»). Это корневой элемент. Определяет XML-документ сообщение SOAP с помощью пространства имен `xmlns:soap=»http://www.w3.org/2003/05/soap-envelope/»`. Если в определении будет указан другой адрес, сервер вернет ошибку.

Header («заголовок»). Включает в себя атрибуты сообщения, связанные конкретным приложением. В заголовке могут использоваться три атрибута, которые указывают, как принимающая сторона должна обрабатывать сообщение, — `mustUnderstand`, `actor` и `encodingStyle`. Значение `mustUnderstand` — 1 или 0 — говорит принимающему приложению о том, следует ли распознавать заголовок в обязательном или опциональном порядке. Атрибут `actor` задает конкретную конечную точку для сообщения. Атрибут `encodingStyle` устанавливает специфическую кодировку для элемента. По умолчанию SOAP-сообщение не имеет определенной кодировки.

Body («тело»). Сообщение, которое передает веб-приложение. Может содержать запрос к серверу или ответ от него.



Структура SOAP-сообщения

Fault («ошибка»). Опциональный элемент. Передает уведомление об ошибках, если они возникли в ходе обработки сообщения. Может содержать вложенные элементы, которые проясняют причину возникновения ошибки:

- faultcode — код неполадки;
- faultstring — «человекопонятное» описание проблемы;
- faultactor — информация о программном компоненте, который вызвал ошибку;
- detail — дополнительные сведения о месте возникновения неполадки.

Пример. Представлен запрос на сервер request ответ response – список континентов

<http://webservices.oorsprong.org/websamples.countryinfo/CountryInfoService.wso>

SOAP 1.2

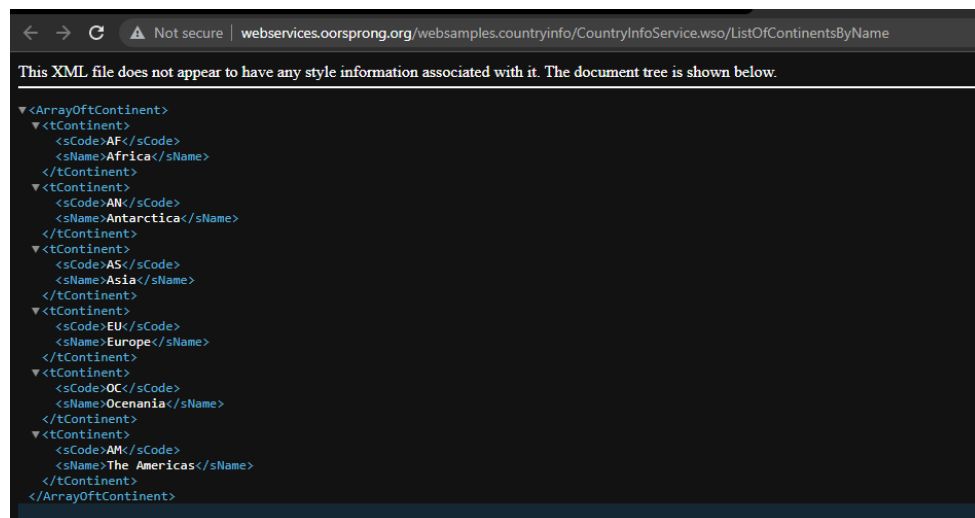
The following is a sample SOAP 1.2 request and response. The placeholders shown need to be replaced with actual values.

```
POST /websamples.countryinfo/CountryInfoService.wso HTTP/1.1
Host: webservices.oorsprong.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ListOfContinentsByName xmlns="http://www.oorsprong.org/websamples.countryinfo">
    </ListOfContinentsByName>
  </soap12:Body>
</soap12:Envelope>

HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <ListOfContinentsByNameResponse xmlns="http://www.oorsprong.org/websamples.countryinfo">
      <ListOfContinentsByNameResult>
        <tContinent>
          <sCode>string</sCode>
          <sName>string</sName>
        </tContinent>
        <tContinent>
          <sCode>string</sCode>
          <sName>string</sName>
        </tContinent>
      </ListOfContinentsByNameResult>
    </ListOfContinentsByNameResponse>
  </soap12:Body>
</soap12:Envelope>
```



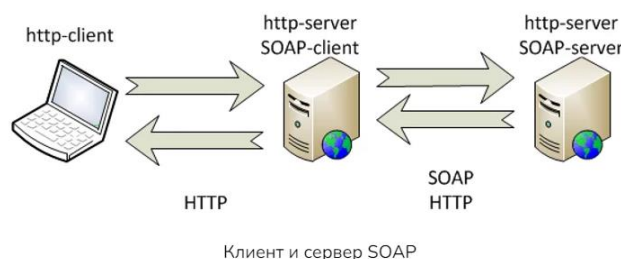
В каких случаях используют SOAP

- Асинхронная обработка и последующий вызов. Стандарт SOAP 1.2 обеспечивает клиенту гарантированный уровень надежности и безопасности.
- Формальное средство коммуникации. Если клиент и сервер имеют соглашение о формате обмена, то SOAP 1.2 предоставляет жесткие спецификации для такого типа взаимодействия. Пример — сайт онлайн-покупок, на котором пользователи добавляют товары в корзину перед оплатой. Предположим, что есть веб-служба, которая выполняет окончательный платеж. Может быть достигнуто соглашение, что веб-сервис будет принимать только название товара, цену за единицу и количество. Если сценарий существует, лучше использовать протокол SOAP.
- Операции с состоянием. Если приложение требует, чтобы состояние сохранялось от одного запроса к другому, то стандарт SOAP 1.2 предоставляет структуру для поддержки таких требований.

Пример работы SOAP вебсервиса

Например, нужно узнать стоимость акции определенной фирмы на бирже. На клиентском компьютере открывается браузер, который является клиентом. По протоколу HTTP он обращается к серверу (HTTP-server).

На этом HTTP-сервере работает приложение, которое отдает информацию, о том, что акция определенной фирмы стоит xxx долларов.



Однако, откуда само приложение, живущее на HTTP-сервере, знает стоимость акции? А все очень просто — приложение в данном случае выступило как SOAP-client и запросило эту информацию на другом сервере (SOAP-server).

Взаимодействие SOAP-client и SOAP-server происходит по протоколу SOAP поверх HTTP. Это значит, что клиент и сервер общаются по протоколу HTTP, но по этому протоколу передаётся не просто стандартное сообщение HTTP, а некий конверт с письмом, причем это письмо написано по правилам протокола SOAP.

То есть сайт, который передал информацию о фирме, сам запросил SOAP-server (то есть биржу акций) по протоколу HTTP и вложил сообщение в конверт SOAP.

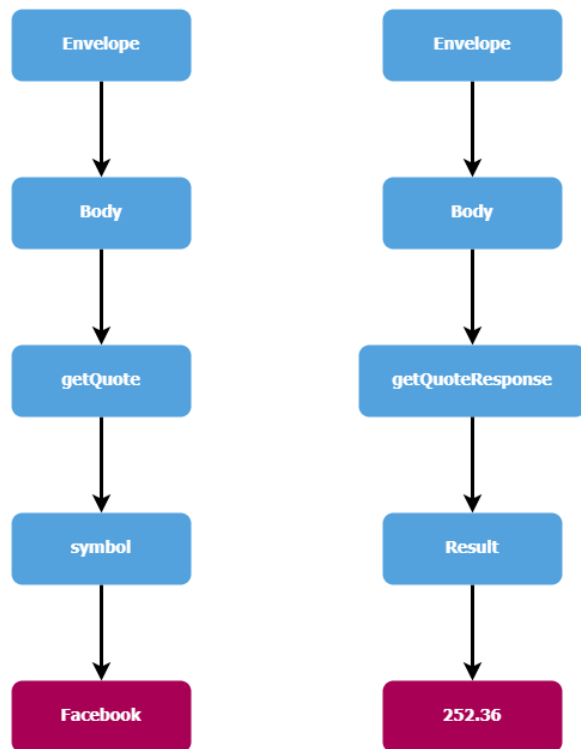
Таким образом, информация о курсе акции пришла не напрямую с биржи, а через посредника — через SOAP-client.

WSDL

В SOAP для описания своего сервиса нужно использовать строгие правила в виде файлов WSDL.

WSDL — это Web Services Description Language - язык описания веб-сервисов и доступа к ним.

Чтобы отправить запрос в биржу акций, нужно ответить на простой вопрос. Название фирмы и сайт биржи акций, которое передается и должны получить ответ «xxx» долларов. Протокол SOAP предполагает, что это текстовое содержимое (название фирмы) вложено внутрь XML-тегов и прописано в стандарте в виде XML-дерева.



Запрос и ответ в виде дерева

Основные теги WSDL-сервера:

- Message — сообщения, используемые web-сервисом.
- PortType — список операций, которые могут быть выполнены с сообщениями.
- Binding — способ, которым сообщение будет доставлено.

На веб-сервисе лежит файл WSDL. И клиент, и сервер руководствуются в своей работе этим файлом: читают его и разбираются, как устроен сервис. И клиент, и сервер умеют читать этот файл и получать из него информацию, так как они знают стандарт SOAP и то, как должен быть устроен файл WSDL.

<http://webservices.oorsprong.org/websamples.countryinfo/CountryInfoService.wso?WSDL>

- operation — это тег, который описывает функции. То есть он указывает на имя функции и то, как должен выглядеть запрос и ответ.
- Вложенные в operation теги input и output содержат информацию о входных и выходных параметрах функции. То есть getQuoteRequest
- Тег binding описывает все технические сведения, о том, что из себя представляет сервис.
- Тег service описывает, где находится сервис.

```

    <operation name="ListOfContinentsByName">
      <documentation>Returns a list of continents ordered by name.</documentation>
      <input message="tns:ListOfContinentsByNameSoapRequest"/>
      <output message="tns:ListOfContinentsByNameSoapResponse"/>
    </operation>
    <operation name="ListOfContinentsByCode">
      <documentation>Returns a list of continents ordered by code.</documentation>
      <input message="tns:ListOfContinentsByCodeSoapRequest"/>
      <output message="tns:ListOfContinentsByCodeSoapResponse"/>
    </operation>
    <operation name="ListOfCurrenciesByCode">

```

```

    <binding name="CountryInfoServiceSoapBinding" type="tns:CountryInfoServiceSoapType">
      <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
      <operation name="ListOfContinentsByName">
        <soap:operation soapAction="" style="document"/>
        <input>
          <soap:body use="literal"/>
        </input>
        <output>
          <soap:body use="literal"/>
        </output>
      </operation>
      <operation name="ListOfContinentsByCode">
        <soap:operation soapAction="" style="document"/>
        <input>
          <soap:body use="literal"/>
        </input>
        <output>
          <soap:body use="literal"/>
        </output>
      </operation>
      <operation name="ListOfCurrenciesByCode">

```

```

    <service name="CountryInfoService">
      <documentation>This DataFlex Web Service opens up country information. 2 letter ISO codes are used for Country code. The
      <port name="CountryInfoServiceSoap" binding="tns:CountryInfoServiceSoapBinding">
        <soap:address location="http://webservices.oorsprong.org/websamples.countryinfo/CountryInfoService.wso"/>
      </port>
      <port name="CountryInfoServiceSoap12" binding="tns:CountryInfoServiceSoapBinding12">
        <soap12:address location="http://webservices.oorsprong.org/websamples.countryinfo/CountryInfoService.wso"/>
      </port>
    </service>
  </definitions>

```

Пример. Получение данных с помощью SOAP на Python

Источник <https://www.geeksforgeeks.org/making-soap-api-calls-using-python/>

```
import requests
# SOAP request URL
url = "http://webservices.oorsprong.org/websamples.countryinfo/CountryInfoService.wso"

# structured XML
payload = """<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ListOfCurrenciesByCode xmlns="http://www.oorsprong.org/websamples.countryinfo">
      </ListOfCurrenciesByCode>
    </soap:Body>
  </soap:Envelope>"""
# headers
headers = {
    'Content-Type': 'text/xml; charset=utf-8'
}
# POST request
response = requests.request("POST", url, headers=headers, data=payload)

# prints the response
print(response.text)
print(response)
```

Результат. Список стран и валют в виде XML

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <m:ListOfCurrenciesByCodeResponse xmlns:m="http://www.oorsprong.org/websamples.countryinfo">
      <m:ListOfCurrenciesByCodeResult>
        <m:tCurrency>
          <m:sISOCODE></m:sISOCODE>
          <m:sName>No Universal Currency</m:sName>
        </m:tCurrency>
        <m:tCurrency>
          <m:sISOCODE>AED</m:sISOCODE>
          <m:sName>Dirhams</m:sName>
        </m:tCurrency>
        <m:tCurrency>
```

Задание

Распечатать на экране список стран и для каждой страны код страны, столица и валюта.

REST vs SOAP

REST — это архитектурный стиль. SOAP — это формат обмена сообщениями.

- Пример реализации RESTful: JSON через HTTP
- Пример реализации SOAP: XML поверх SOAP через HTTP

На верхнем уровне SOAP ограничивает структуры ваших сообщений, тогда как REST — это архитектурный подход, ориентированный на использование HTTP в качестве транспортного протокола.

- Специфика SOAP — это формат обмена данными. С SOAP это всегда SOAP-XML, который представляет собой XML, включающий:
 - Envelope (конверт) — корневой элемент, который определяет сообщение и пространство имен, использованное в документе,
 - Header (заголовок) — содержит атрибуты сообщения, например: информация о безопасности или о сетевой маршрутизации,
 - Body (тело) — содержит сообщение, которым обмениваются приложения,
 - Fault — необязательный элемент, который предоставляет информацию об ошибках, которые произошли при обработке сообщений. И запрос, и ответ должны соответствовать структуре SOAP.
- Специфика REST — использование HTTP в качестве транспортного протокола. Он подразумевает наилучшее использование функций, предоставляемых HTTP — методы запросов, заголовки запросов, ответы, заголовки ответов и т. д.

Формат обмена сообщениями

- В SOAP вы используете формат SOAP XML для запросов и ответов.
- В REST такого фиксированного формата нет. Вы можете обмениваться сообщениями на основе XML, JSON или любого другого удобного формата. JSON является самым популярным среди используемых форматов.

Определения услуг

- SOAP использует WSDL (Web Services Description Language) — язык описания веб-сервисов и доступа к ним, основанный на языке XML.
- REST не имеет стандартного языка определения сервиса. Несмотря на то, что WADL был одним из первых предложенных стандартов, он не очень популярен. Более популярно использование Swagger или Open API.

Простота реализации

RESTful веб-сервисы, как правило, гораздо проще реализовать, чем веб-сервисы на основе SOAP.

REST обычно использует JSON, который легче анализировать и обрабатывать. В дополнение к этому, REST не требует наличия определения службы для предоставления веб-службы.

SOAP необходимо определить свой сервис с использованием WSDL, и при обработке и анализе сообщений SOAP-XML возникают большие накладные расходы.

	SOAP	REST
Стандарт	протокол	архитектурный стиль
Создан	1998, Дейв Винер, передан в W3C	Рой Филдинг, 2000
Транспорт	HTTP	HTTP
Формат обмена сообщениями	XML	можно использовать XML, JSON или любой другой удобный формат
Описание сервиса	использует WSDL (Web Services Description Language) — язык описания веб-сервисов и доступа к ним, основанный на языке XML	не имеет стандартного языка определения сервиса

Используемые источники

1. SOAP API <https://blog.skillfactory.ru/glossary/soap-api/>
2. Применение SOAP при интеграции систем <https://systems.education/soap-integration>