

Ingesting SOAP Action from SOAP Payload

Toy SOAP Processing

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV = "http://schemas.xmlsoap.org/soap/envelope/"  
  SOAP-ENV:encodingStyle = "http://schemas.xmlsoap.org/soap/encoding/">  
  <SOAP-ENV:Body>  
    <m:MyMessage xmlns:m = "http://electrocommerce.org/abc">  
      <m:MyArgument>Hello</m:MyArgument>  
    </m:MyMessage>  
  </SOAP-ENV:Body>  
</SOAP-ENV:Envelope>
```

SOAP 1.1 example payload



```
<SOAP-ENV:Envelope>  
  <SOAP-ENV:Body>  
    <m:MyMessage>  
      <m:MyArgument>Hello</m:MyArgument>  
    </m:MyMessage>  
  </SOAP-ENV:Body>  
</SOAP-ENV:Envelope>
```



```
{  
  "SOAP-ENV:Envelope": {  
    "SOAP-ENV:Body": {  
      "m:MyMessage": [Object]  
    }  
  }  
}
```

Real World SOAP Payload

```
<wsdl:definitions xmlns:soap = "http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:wsdl = "http://schemas.xmlsoap.org/wsdl/" name = "EmployeeService"
  targetNamespace = "http://www.jpworks.com/employee">

  <wsdl:portType name = "EmployeeServicePortType">
    <wsdl:operation name = "GetEmployeeById">
      <wsdl:input message = "EmployeeByIdRequest"/>
      <wsdl:output message = "EmployeeResponse"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name = "EmployeeServiceSOAP" type = "EmployeeServicePortType">
    <wsdl:operation name = "GetEmployeeById">
      <soap:operation soapAction = "http://www.jpworks.com/employee/GetEmployeeById"/>
      <soap:body use = "literal"/>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name = "EmployeeService">
    <wsdl:port name = "EmployeeServiceSOAP" binding = "EmployeeServiceSOAP"/>
  </wsdl:service>
</wsdl:definitions>
```

heavily truncated to fit in one slide..

Real World SOAP Payload Inspection Result

```
{  
  "wsdl:definitions": {  
    "wsdl:portType": { "wsdl:operation": [Object] },  
    "wsdl:binding": { "wsdl:operation": [Object] },  
    "wsdl:service": { "wsdl:port": "" }  
  }  
}
```

Javascript Painless Conversion

XML to JSON

- String Manipulation
- Regex Manipulation *
- Recursive calls *
- Maps and Lists

XML Element with children XML Element(s) is elementary

XML Element with Attribute(s) and children XML Element(s)
is not obvious in JSON

Painless Script Development

```
POST /apm-7.17.7-transaction-000001/_doc/t9WKMYUB7ikC1HY84JYH/_update
{
  "script": {
    "lang": "painless",
    "source": """
      String unescape(def data){ return / /.matcher(clear(
        /\n|\t/.matcher(data).replaceAll(' '))).replaceAll('');}

      String clear(def data){
        return /[\\w:-]+ = "[^"]+"/.matcher(data).replaceAll(''); }

      String data  = '<soap:x xmlns:soap="http://schemas.xmlsoap.org">';
      ctx._source.transaction.name = unescape(data);
      """
  }
}
```

```
GET /apm-7.17.7-transaction-000001/_doc/t9WKMYUB7ikC1HY84JYH?
_source=transaction
```

See Also

- [Painless Elastic Search Scripting Language](#)
- [Shared Java API](#)
- [Custom ScriptEngine](#)
- [Circuit Breaker Settings](#)
- [Ingest Pipelines to Enhance Observability Data \(blog\)](#)
- [SOAP 1.1](#)
- [SOAP 1.2](#)