

# Programação Orientada a Serviços

**Prof. Diego Cirilo**

**Aula 12: APIs SOAP**

# SOAP

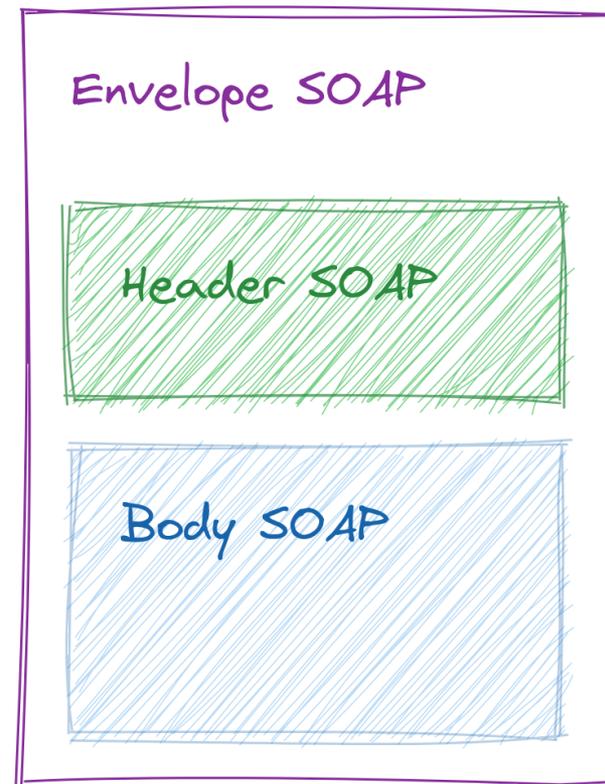
- *Simple Object Access Protocol*
- Usualmente XML sobre HTTP
- Define um protocolo, ao contrário do REST
- Mais antigo, mais complexo e mais bem definido que o REST
- Características:
  - Extensibilidade
  - Neutralidade
  - Independência

# SOAP

- Envelope que define a estrutura de mensagens
- Conjunto de regras de *encoding* para expressar os tipos de dados
- Convenção para expressar as chamadas e respostas

# Elementos da mensagem SOAP

Elemento	Descrição	Obrigatório?
Envelope	Identifica o documento XML como uma mensagem SOAP	Sim
Cabeçalho	Contém informação de cabeçalho ( <i>header</i> )	Não
Corpo	Informações de chamada e retorno ( <i>body</i> )	Sim
Falha	Informações de erros durante o processamento da mensagem	Não



# WSDL

- *Web Services Description Language*
- Baseado em XML
- Usado para descrever um serviço SOAP, como um *schema*.



# Exemplo de API SOAP

- <http://webservices.oorsprong.org/websamples.countryinfo/CountryInfoService.wso>
- Exemplo de operação: CountryCurrency
- Requisição (POST)

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <CountryCurrency xmlns="http://www.oorsprong.org/websamples.countryinfo">
      <sCountryIS0Code>string</sCountryIS0Code>
    </CountryCurrency>
  </soap:Body>
</soap:Envelope>
```

# Exemplo de API SOAP

- Resposta

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <CountryCurrencyResponse xmlns="http://www.oorsprong.org/websamples.countryinfo">
      <CountryCurrencyResult>
        <sISOCODE>string</sISOCODE>
        <sName>string</sName>
      </CountryCurrencyResult>
    </CountryCurrencyResponse>
  </soap:Body>
</soap:Envelope>
```

# Consumindo SOAP com Python

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

# XML estruturado
payload = """<?xml version="1.0" encoding="utf-8"?>
    <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
        <soap:Body>
            <CountryIntPhoneCode xmlns="http://www.oorsprong.org/websamples.countryinfo">
                <sCountryISOCODE>BR</sCountryISOCODE>
            </CountryIntPhoneCode>
        </soap:Body>
    </soap:Envelope>"""

# headers
headers = {
    'Content-Type': 'text/xml; charset=utf-8'
}
# request POST
response = requests.request("POST", url, headers=headers, data=payload)

# imprime a resposta
print(response.text)
print(response)
```

# Tarefa

- Utilize a função `CapitalCity` da API de exemplo para descobrir a capital da Nova Zelândia (NZ)
- Realize testes com outras 3 funções (escolha) e faça o *parse* usando a biblioteca `xml.dom.minidom`

# Zeep

- Biblioteca que facilita a interação com APIs SOAP
- Automaticamente recupera as funções disponíveis através do WSDL e transforma em métodos
- O usuário não precisa operar com XML diretamente
- [Documentação](#)

# Exemplo com Zeeep

```
import zeeep

# define a URL do WSDL
wsdl_url = "http://webservices.oorsprong.org/websamples.countryinfo/CountryInfoService.wso?WSDL"

# inicializa o cliente zeeep
client = zeeep.Client(wsdl=wsdl_url)

# define o código do país para BR
country_code = "BR"

# faz a chamada do serviço
result = client.service.CountryIntPhoneCode(
    sCountryISOCODE=country_code
)
# imprime o resultado
print(f"O código de telefone do {country_code} é {result}")

# define o código do país para US
country_code = "US"

# faz a chamada do serviço
result = client.service.CountryIntPhoneCode(
    sCountryISOCODE=country_code
)

# imprime o resultado
print(f"O código de telefone do {country_code} é {result}")
```

# Tarefa

- Utilizando a mesma lógica do exemplo, imprima o nome da capital da Noruega (NO)
- Acesse a documentação da API de conversão de números ([link](#)), descubra a URL do WSDL e faça um programa que imprime um número digitado (ex. 223) por extenso em inglês (ex. two hundred and twenty three)

# Dúvidas?

