

Consumindo Web Service SOAP

Para obtermos um recurso via SOAP, é necessário que exista um *endpoint* [WSDL](http://webservices.oorsprong.org/websamples.countryinfo/CountryInfoService.wso?WSDL). Assim, para esse tutorial usaremos um serviço que retorna algumas informações de países e obteremos o DDI do país informado. Esse *endpoint* SOAP está disponível gratuitamente em <http://webservices.oorsprong.org/websamples.countryinfo/CountryInfoService.wso?WSDL>

Pré requisitos

- Projeto do tipo *mobile-web* ou *web* criado. Caso haja dúvidas de como criar esse tipo de projeto acesse o link [criar projeto](#).

Passos

Importar serviço SOAP

Clique com o botão direito sobre o nome do projeto e selecione a opção **Importar** no menu. Ao abrir a janela de importação, selecione a opção **SOAP** e finalize (Figura 1).

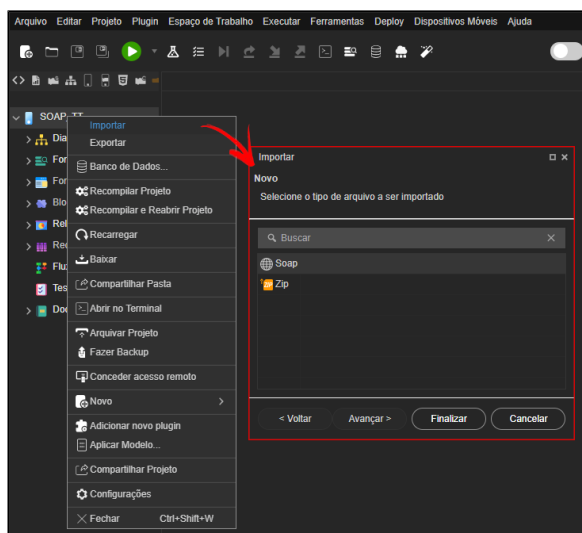


Figura 1 - Importando serviço web do tipo SOAP

Após clicar em finalizar, será aberta a janela de importação do SOAP, insira a URL do *webservice* desejado e clique em **importar** (Figura 1.2).

Nesta página

- [Pré requisitos](#)
- [Passos](#)
 - [Importar serviço SOAP](#)
 - [Utilizando o serviço importado](#)

Outras informações:

[Web Services](#)

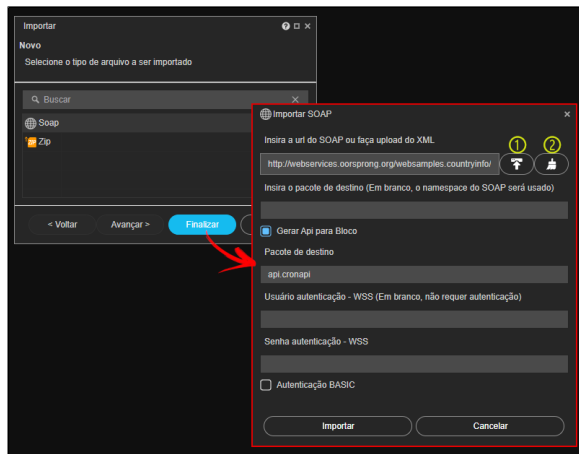


Figura 1.2 - Dados da importação

- **Insira a url do SOAP ou faça upload do XML:** informe o endereço do serviço SOAP na caixa de texto ou utilize o botão **Enviar arquivos** para fazer upload do XML ([exemplo de endereço SOAP](#)).
Obs.: Endereços SOAP sempre terminam com a notação `?WSDL`.
- 1. **Enviar arquivos:** abre a janela para a seleção do arquivo XML do SOAP.
- 2. **Limpar:** remove o conteúdo do campo.
- **Insira o pacote de destino:** nome do pacote que será criado para armazenar os dados do serviço no projeto, caso deixe em branco, será utilizado o nome do pacote do serviço SOAP.
- **Gerar API para Blocos:** gera automaticamente uma categoria de blocos de programação, convertendo as funções em blocos. Esses blocos serão exibidos no [Editor de blocos](#) servidor.
- **Pacote de destino:** pasta onde armazenará o pacote criado do *webservice*.
- **Usuário autenticação - WSS:** usuário de autenticação [Web Security Service](#).
- **Senha autenticação - WSS:** senha de autenticação [Web Security Service](#).
- **Autenticação BASIC:** essa opção deve ser selecionada sempre que for necessária a autenticação basic do serviço SOAP. Dessa forma, o usuário e a senha informados serão incluídos no *header* da requisição, no formato de codificação Base64, utilizando [autenticação basic](#).

Caso o seu projeto esteja com o [Modo Avançado](#) habilitado, após importar o *webservice* com sucesso, será exibido uma janela informando os pacotes [Maven](#) necessários que serão adicionados no arquivo POM, então clique em **salvar** para baixá-los.

Após carregar os pacotes (o que pode levar um tempo) é necessário **recompilar e reabrir o projeto**, a IDE gera um arquivo Java com as chamadas para as funções. Esse arquivo é gerado automaticamente, independente da arquitetura e/ou linguagem de programação em que o *webservice* foi desenvolvido (Figura 2).

Importante

É necessário habilitar o [Modo Avançado](#) para que seja possível visualizar o arquivo.

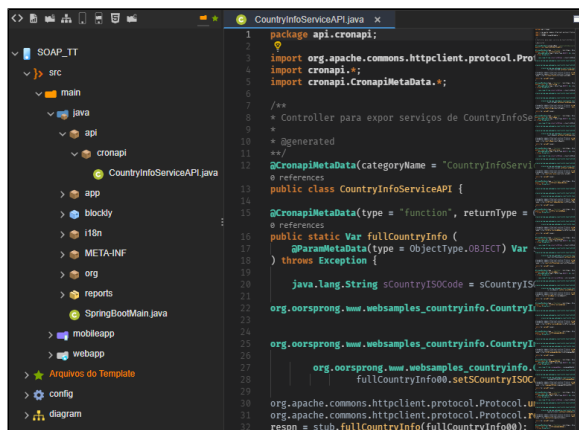


Figura 2 - Arquivo gerado com as chamada das funções Java

Utilizando o serviço importado

Ao abrir/criar um bloco de programação do lado servidor, uma nova categoria é inserida com as funções de blocos importadas do SOAP (Destaque 1 da Figura 3).

O nome da categoria, métodos e parâmetros são os mesmos passados pelo serviço, por isso podem vir pouco legíveis. Nesse caso, utilize a ferramenta de [Internacionalização](#) (servidor) para alterar esses nomes e deixá-los mais legíveis.

Como o objetivo deste tutorial é apresentar apenas a conversão das funções do serviço SOAP em Blocos de programação, não iremos detalhar as configurações da função de bloco (Figura 3) e view (Figura 4) usados nesse exemplo. Caso tenha dúvidas, veja em [Bloco de programação](#) e [Editor de views](#).

Na Figura 3, internacionalizamos o nome do método *countryIntPhoneCode* (**DDI do País**) e do seu parâmetro *sCountryISOCodeVar* (**Sigla do País**) (destaque 2 da figura 3), mas não fizemos no nome da categoria: **CountryInfoService** (destaque 1 da Figura 3).

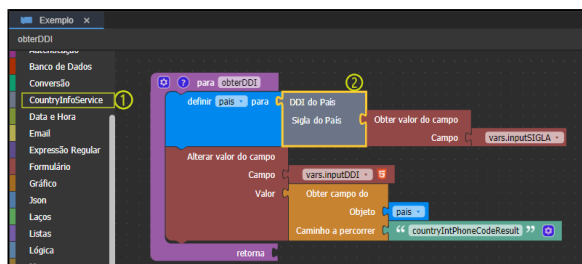


Figura 3 - Função que faz consulta do SOAP e alimenta os campos de uma página

Após a requisição do cliente consumidor, o provedor do serviço retornará sua resposta em formato XML, como podemos ver abaixo. Por esse motivo será necessário usar o bloco "**Obter campo do**" para percorrer o caminho até o *countryIntPhoneCodeResult* do objeto e obter o seu valor.

Retorno do serviço

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <m:FullCountryInfoResponse xmlns:m="http://www.oorsprong.org/websamples.countryinfo">
      <m:FullCountryInfoResult>
        <m:sISOCode>BR</m:sISOCode>
        <m:sName>Brazil</m:sName>
        <m:sCapitalCity>Brasilia</m:sCapitalCity>
        <m:sPhoneCode>55</m:sPhoneCode>
        <m:sContinentCode>AM</m:sContinentCode>
        <m:sCurrencyISOCode>BRL</m:sCurrencyISOCode>
        <m:sCountryFlag>http://www.oorsprong.org/WebSamples.CountryInfo/Images/Brazil.jpg</m:sCountryFlag>
        <m:Languages>
          <m:tLanguage>
            <m:sISOCode>por</m:sISOCode>
            <m:sName>Portuguese</m:sName>
          </m:tLanguage>
        </m:Languages>
      </m:FullCountryInfoResult>
    </m:FullCountryInfoResponse>
  </soap:Body>
</soap:Envelope>
```

Na Figura 4 vemos o exemplo de uma tela que retorna o DDI de um país baseado no seu código ISO.



The image shows a mobile application interface with a blue header bar containing the title "Exemplo" and two hamburger menu icons. Below the header is a search bar with the text "BRA" and a button labeled "Pesquisar" with a magnifying glass icon. Below the search bar, the result "DDI: 55" is displayed.

DDI:	55
------	----

Figura 4 - Resultado do exemplo