

<p>Documentação complementar:</p> <ul style="list-style-type: none"> • Postman • RESTXQ • Repositório

PARTE I

1. Utilizando a ferramenta Postman, realize o download e importação dos pedidos SOAP disponibilizados no seguinte URL: <https://explore.postman.com/templates/7275/public-soap-apis>. Teste cada um dos pedidos, analisando o método HTTP utilizado, o Header e o Body dos pedidos.

2. Considere a API disponibilizada no seguinte endereço: <https://www.amiiboapi.com>.

2.1. Utilizando a ferramenta Postman, realize um pedido HTTP à API apresentada. Proceda à invocação do *endpoint* que lhe permita obter todos os amiibos de um determinado personagem. Analise o pedido. Relativamente à pergunta 1 da ficha, quais são as diferenças entre os dois pedidos?

3. Utilizando BaseX, crie uma base de dados: *cities*. Crie um *endpoint* que receba conteúdo XML e armazene numa base de dados BaseX. Teste o *endpoint* com o Postman.

Indicações:

Considere os exemplos: <https://github.com/ESTG-PEI/2020-2021/blob/main/BaseX/myRequestsAPP6.xqm>. Na pasta de instalação do BaseX, em particular na pasta *webapp*, crie um novo ficheiro .xqm para a criação da *web API*

Crie uma função XQuery:

```
declare %updating
    %rest:path("/addcity")
    %rest:consumes("application/xml", "text/xml")
    %rest:POST("{ $body }")
function page:addToDatabase($body as document-node()) {
    db:add("cities", $body, "city.xml")
};
```

Consulte a documentação e analise cada uma das anotações:

https://docs.basex.org/index.php?title=RESTXQ&mobileaction=toggle_view_desktop

Teste com o seguinte documento. Para isso deverá criar um pedido HTTP (Postman) com o seguinte *body*:

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

4. Considerando a coleção do Postman da pergunta 1 (“Public SOAP APIs”), considere a pasta: *countries* e em específico o pedido: “Capital City for a Country”.

4.1. Crie uma coleção no Postman.

4.2. Copie o pedido: “Capital City for a Country” para a nova coleção.

4.3. Crie um novo *environment*.

4.4. Na aba *Tests*, inclua o seguinte código:

```
pm.environment.set("XMLresponse", responseBody);
```

Esta função permite definir uma variável de ambiente com a resposta do pedido.

4.5. Na coleção, crie um novo pedido para invocar o *endpoint* criado na alínea 3. No *body* do pedido (Postman), referencie a variável de ambiente:

```
{ {XMLresponse} }
```

Execute a coleção e valide o seu *output* na base de dados BaseX.

5. Crie um *endpoint* que permita retornar todas as cidades armazenadas na base de dados criada na alínea 3. Deverá apresentar (como *output*) um documento simplificado (sem referências ao pedido SOAP).

PARTE 2

1. Utilizando o módulo RESTXQ do BaseX, crie um conjunto de endpoints (num ficheiro à sua escolha), de acordo com os seguintes requisitos:

1.1. Crie um endpoint que através do método GET, receba dois parâmetros: *numero1* e *numero2*, devendo retornar uma resposta com a soma desses dois números. Deve desenvolver duas versões:

Enviando os valores dos números diretamente no URL (exemplo : `/soma/1/2`);

Enviando os valores dos números utilizando parâmetros no URL (exemplo: `/soma?number1=1&number2=2`). Dica: Investigue a anotação: `query-param`.

1.2. Crie um *endpoint* que através do método POST, receba através do body um documento XML (deve restringir o tipo de conteúdo aceite) com uma lista de livros (*bookstore.xml*), retornando o número de livros existentes no documento. Dica: Investigue a anotação: `consumes`.

1.3. Crie um endpoint que através do método PUT receba um documento XML (deve restringir o tipo de conteúdo aceite) com uma lista de livros (*bookstore.xml*). Crie um vocabulário XSD para uma *bookstore*. Deverá validar a estrutura do documento recebido (https://docs.basex.org/wiki/Validation_Module).

A função deverá retornar um novo documento XML de acordo com a estrutura representada pelo documento *newBookStore.xml* disponibilizado no moodle. O *endpoint* deve explicitamente indicar que devolverá um documento XML. Dica: Investigue a anotação: `produces`.

2. Crie uma base de dados: *books*. De seguida, crie os seguintes *endpoints*:

2.1. Adicionar um ficheiro XML. Crie pedidos para inserir os documentos disponibilizados: *books1.xml* e *books2.xml*. Dica: Investigue a anotação: `%updating`.

2.2. Obter todos os livros num único documento.