

Utilizei as seguintes ferramentas para o desenvolvimento do projeto:
Visual Studio 2017 Community para Backend e Services

IIS (10) para Hospedar o serviço localmente.

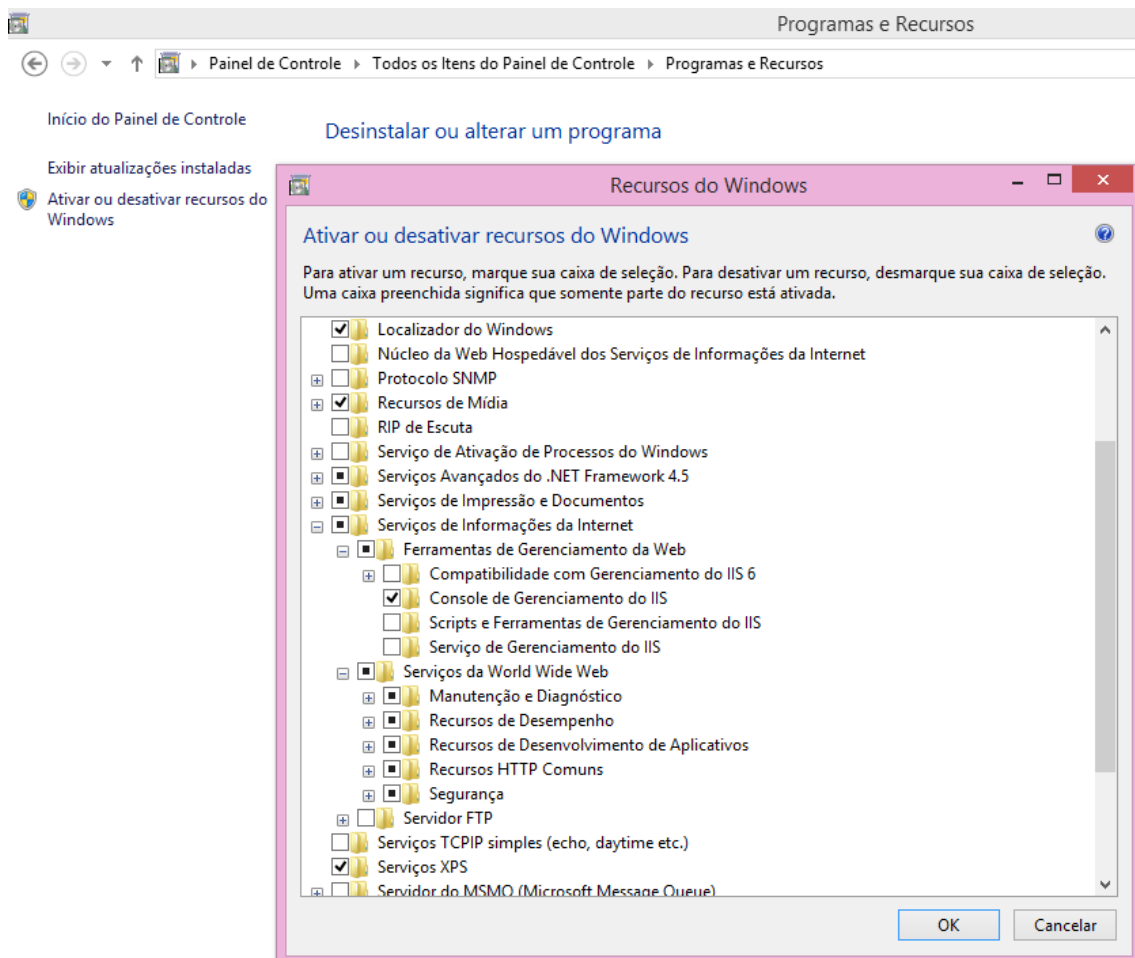
Postgres e pgAdmin 4 para banco. Link: <https://www.pgadmin.org/download/>

SoapUI 5.5.0 para testes do Webservice. Link:
<https://www.soapui.org/downloads/soapui.html>

Configuração do IIS:

Painel de Controle => Programas e Recursos => Ativar ou desativar recursos do Windows

Ativar as opções em “Serviços de Informações da Internet” no item “Ferramentas de Gerenciamento da Web” a opção de “Console de Gerenciamento do IIS” e a opção “Serviços da World Wide Web”



Ao abrir o IIS adicionar um novo site com o nome “Default” caminho físico
“C:\inetpub\wwwroot”

Dentro dele adicionar diretório com nome de web_service para pasta de publish do
web_service “WebApplication2”.

O Postgres, necessário um usuário com máxima permissão, utilizei o por padrão “postgres”

Caso necessário, alterar os arquivos: “appsettings.json” do projeto “WebApplication1”

A linha 3 - “DefaultConnection”:

```
"Host=localhost;Port=5432;Pooling=true;Database=Geral;User  
Id=postgres;Password=senha;"
```

Com os dados do banco postgres que utilizará.

No projeto “WebApplication2” o arquivo WebService1.asmx na linha 20 `public string
URLBanco = "Server=127.0.0.1;User Id=postgres;Password=senha;Database=Geral;"`;

O ip caso necessário e demais dados do Acesso ao postgres.

Caso o IIS e o Banco esteja configurado.

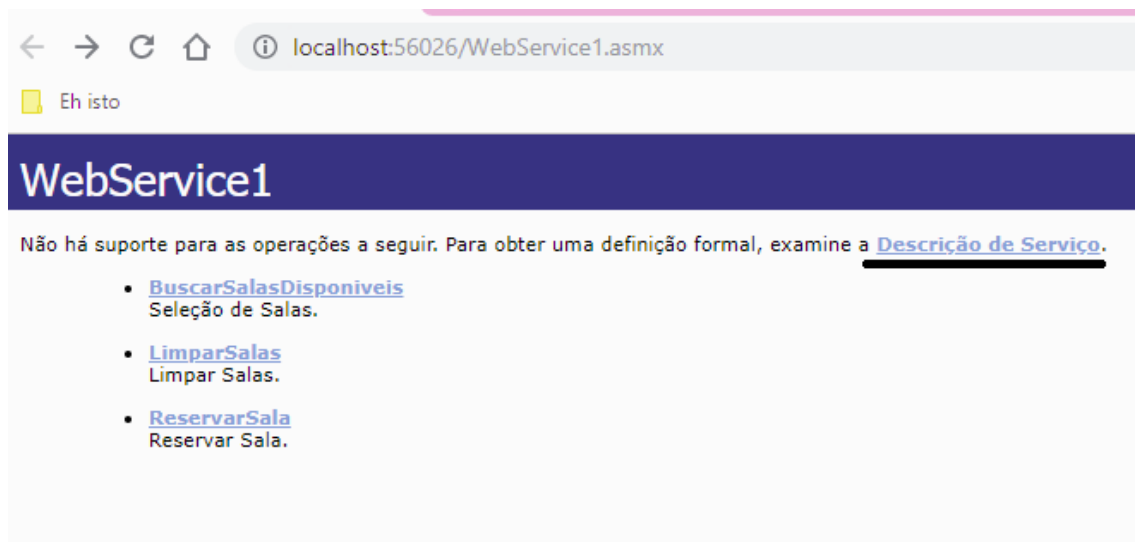
Abrir o projeto “WebApplication1” e executar (f5) , para gerar o banco e startar a aplicação web.

Inserir dados de 1 ou mais registros para teste.

Abrir o projeto “WebApplication2” e executar (f5) , podendo utilizar pelo navegador as funções do WebService para Reservar Salas.

Caso queira utilizar pelo SoapUI necessário configurar para realizar os testes utilizando se do padrão XML para executar as Funções.

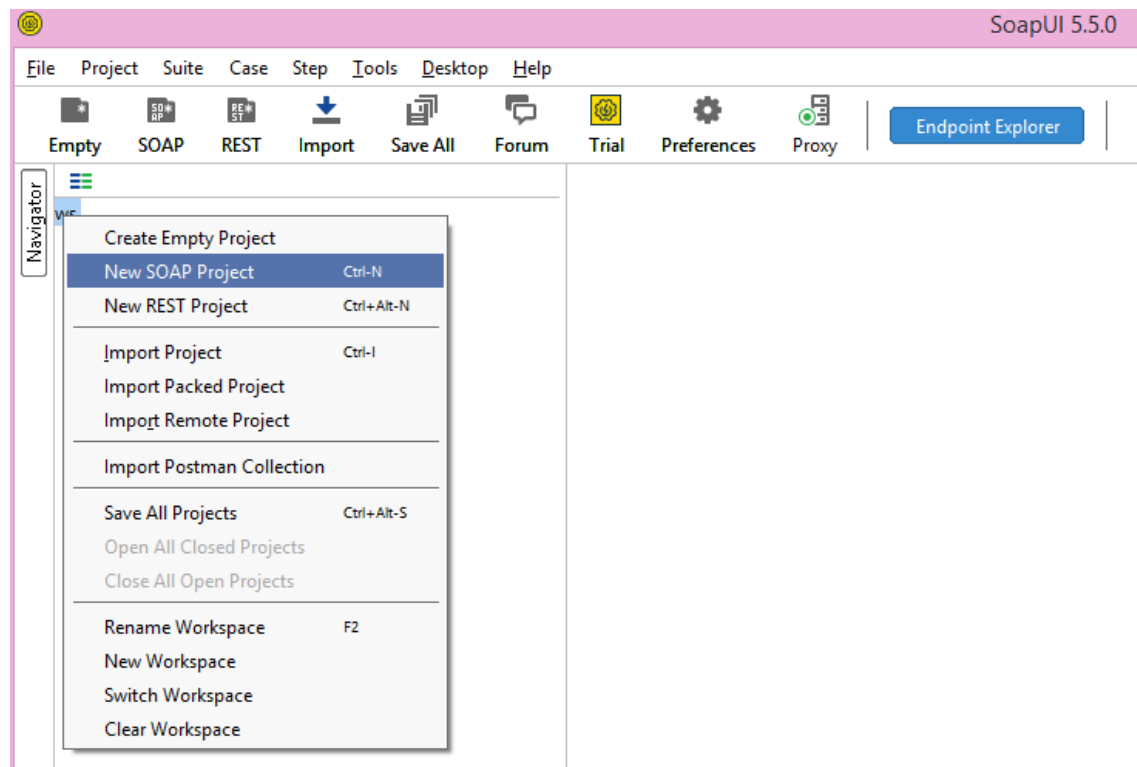
Na pagina do WebService



Ir em Descrição de Serviço e copiar a url para o WSDL



No SOAPUI Adicionar um novo SOAP Project



Inserindo a url com o WSDL no campo

New SOAP Project

Creates a WSDL/SOAP based Project in this workspace

Project Name:

Initial WSDL:

Create Requests: ☒ Create sample requests for all operations?

Create TestSuite: ☐ Creates a TestSuite for the imported WSDL

Relative Paths: ☐ Stores all file paths in project relatively to project file (requires save)

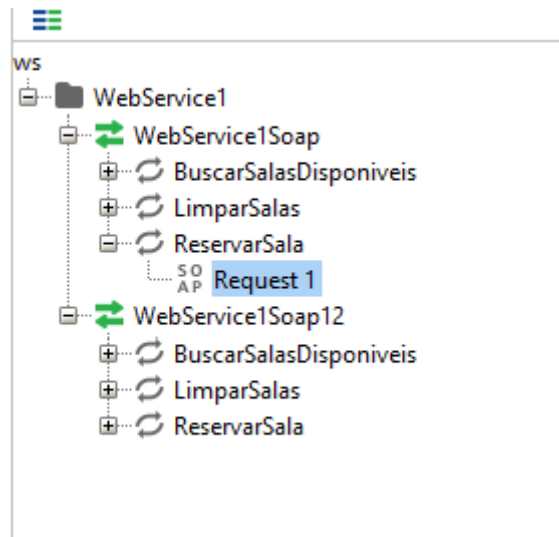
Ao criar as opções e funções estarão disponíveis para teste.

Project Explorer:

- WebService1
 - WebService1Soap
 - BuscarSalasDisponiveis
 - Request 1
 - LimparSalas
 - Request 1
 - ReservarSala
 - Request 1
 - WebService1Soap12
 - BuscarSalasDisponiveis
 - LimparSalas
 - ReservarSala

Raw XML View:

```
<?xml version='1.0' encoding='utf-8'>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:BuscarSalasDisponiveis/>
  </soapenv:Body>
</soapenv:Envelope>
```



Request 1

http://localhost:56026/WebService1.asmx

Submit request to specified endpoint URL (Alt-Enter)

```
<?xml version='1.0' encoding='utf-8'>
<soapenv:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <web:ReservarSala>
      <!--Optional:-->
      <web:titulo>teste</web:titulo>
      <!--Optional:-->
      <web:sala>6</web:sala>
      <!--Optional:-->
      <web:data>10/07/2019</web:data>
    </web:ReservarSala>
  </soapenv:Body>
</soapenv:Envelope>
```

```
<?xml version='1.0' encoding='utf-8'>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ReservarSalaResponse xmlns="http://localhost/WebService">
      <ReservarSalaResult>A sala foi Reservada com Sucesso.</ReservarSalaResult>
    </ReservarSalaResponse>
  </soap:Body>
</soap:Envelope>
```

... Head... Attach... ... W... JMS ... JMS Pro... Headers (9) Attachments (0) SSL Info WSS (0) JMS (0)