

Integração de Sistemas

2017 / 2018

Trabalho 2

Programação de Web Services (RESTlet Framework), Bases de Dados e Aplicações Web

Duração: 4 aulas acompanhadas por docente (+1 semana para entrega)

Entrega: 13 de Maio de 2018

1. INTRODUÇÃO

Os *web services* (WS) têm um papel de crescente importância no contexto da integração da sistemas. Existem duas características fundamentais num *web service*. A ausência de estado e a facto de possuírem uma interface auto-descritiva normalmente documentada em WSDL (*Web Services Description Language*) ou WADL (*Web Application Description Language*). Os *web services* enquanto ferramenta de integração não especificam normalmente a forma como os dados que lhes são submetidos e a respetiva resposta são introduzidos e apresentados. No presente trabalho irá explorar os WS como ferramenta de integração e uma aplicação externa (móvel ou *web based*) como tecnologia que permite a introdução de dados e invocação de serviços através de um navegador web (“browser”) ou dispositivo móvel.

2. Apresentação do Problema

“Dispositivos de geração de energia na *Cloud*”

Com a crescente utilização da Internet para a partilha de conteúdos, surgiu também o aparecimento de novas abordagens para a utilização da mesma. Neste caso o objetivo é utilizar esta ferramenta “global” de forma a possibilitar que cada consumidor gira e consulte os seus geradores através de aplicações externas. Neste caso o utilizador deverá ter a hipótese de ligar e desligar, consultar estados e produção de energia de todos os seus geradores através de uma aplicação externa. Com esta funcionalidade, também se torna possível que os fabricantes recebam informação dos seus produtos, durante o tempo de vida destes.

Infraestrutura

No trabalho é pedido que se construa uma infraestrutura (Figura 1), disponível na Internet, em que o consumidor seja capaz de inscrever um novo eletrodoméstico e operá-lo através de uma aplicação externa. Também é pedido que o fabricante consiga consultar historiais dos vários dispositivos de forma a poder analisar esta informação.

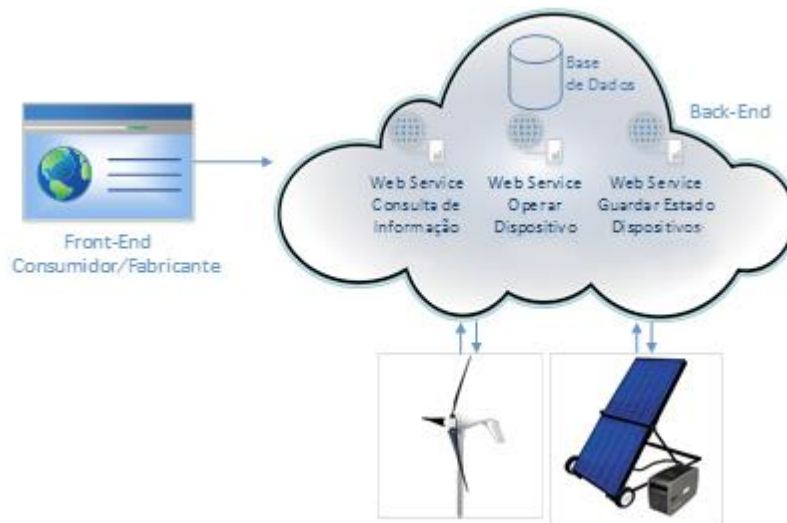


Figura 1 - Infraestrutura a implementar

No *Front-End* do consumidor deveram ser disponibilizadas opções, **tais como:**

- Adicionar novo consumidor;
- Associar um novo dispositivo;
- Consultar dispositivos associados;
- Historiais do dispositivo;
- **Etc.**

No *Front-End* do fabricante deveram ser apresentadas informações associadas aos estados dos dispositivos, com algumas opções de pesquisa, **tais como:**

- Historial de um determinado dispositivo;
- Historial de um determinado utilizador;
- Historial de dispositivos de um determinado tipo (Eólico, Fotovoltaico, Hidroelétrico, etc);
- **Etc.**

Base de Dados

A base de dados (Figura 2), já fornecida pelo corpo docente, guarda os vários utilizadores registados e os dispositivos associados, tal como o seu historial.

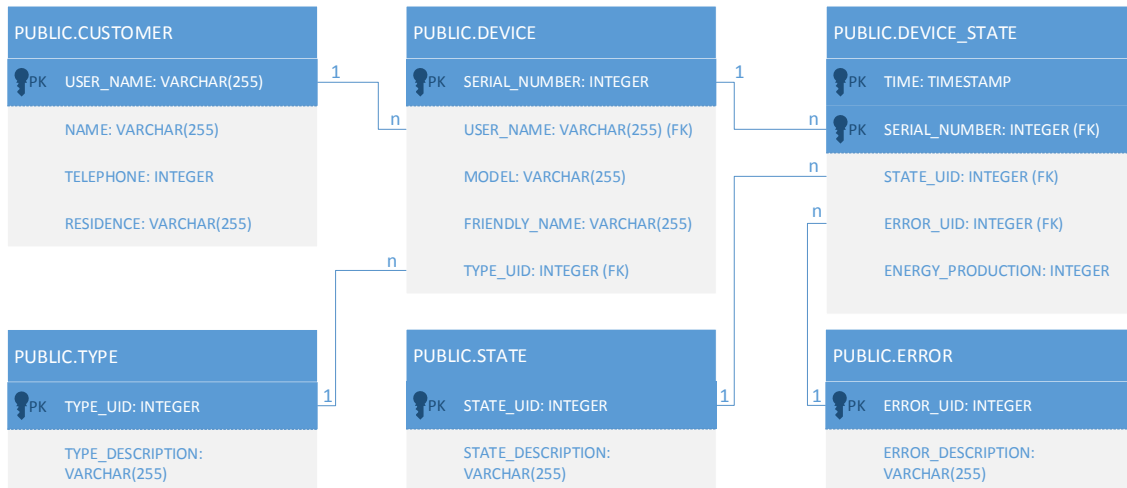


Figura 2 - Diagrama de Entidade e Relacionamento da Base de Dados

A tabela *PUBLIC.TYPE* guarda o tipo de dispositivo inscrito, em que o *TYPE_UID* numérico corresponde a uma descrição, por exemplo o *TYPE_UID* 1 corresponde a “Fotovoltaico”.

A base de dados inicialmente é criada com os seguintes parâmetros:

- Tabela *PUBLIC.TYPE*:

TYPE_UID	TYPE_DESCRIPTION
1	Fotovoltaico
2	Eólico

- Tabela *PUBLIC.STATE*:

STATE_UID	STATE_DESCRIPTION
1	ON
2	OFF

- Table *PUBLIC.ERROR*:

<i>ERROR_UID</i>	<i>ERROR_DESCRIPTION</i>
0	Sem Erros
1	Produção de Energia
2	Extração de Dados

Todas as restantes tabelas inicializam-se vazias.

3. Implementação

Na implementação pedida será utilizado o seguinte material:

- Linguagem JAVA (Netbeans/IntelliJ IDEA/Eclipse);
- RESTlet Framework;
- H2 DB
- Angular CLI (Webstorm/Sublime/Atom)
- Código fornecido pelos docentes da cadeira;

4. Planeamento das Aulas

Aula 1 – Back-End API

1. Siga as instruções fornecidas pelos docentes e instale um dos IDEs de Java indicados.
2. Estude o código fornecido pelos docentes.

Leia atentamente todo o código fornecido, familiarizando-se com as funcionalidades implementadas.

3. Implemente os serviços que deverão ser consumidos pelo front-end. Deverá criar os *ServerResources* necessários, bem como adicionar as rotas necessárias na classe *BackendServerApplication* do package *BackendServer*.

Deverá implementar **pelo menos** os seguintes serviços:

- Inserir novo consumidor;
- Associar novo dispositivo;
- Listar dispositivos associados a um utilizador;
- Criar Base de Dados;
- Ver historial de um utilizador;
- Ver historial de um dispositivo.

Tenha em atenção os serviços que serão necessários quando criar o *front-end*.

Aula 2 – Chamada do primeiro serviço no Front-End - CreateDB

1. Siga as instruções fornecidas pelos docentes e instale um dos IDEs/Text Editors indicados para *web development*.
2. Instale o Node.js, npm e Angular CLI na sua máquina, seguindo os passos indicados em:

<https://angular.io/guide/quickstart>

3. Abra uma janela de *Command Prompt* na directoria do projecto *devicemanagementfrontend* fornecido pelos docentes e execute o comando *npm install*. Abra o projeto no software escolhido e desenvolva o código necessário para chamar o seu primeiro serviço na página de teste (Figura 3).

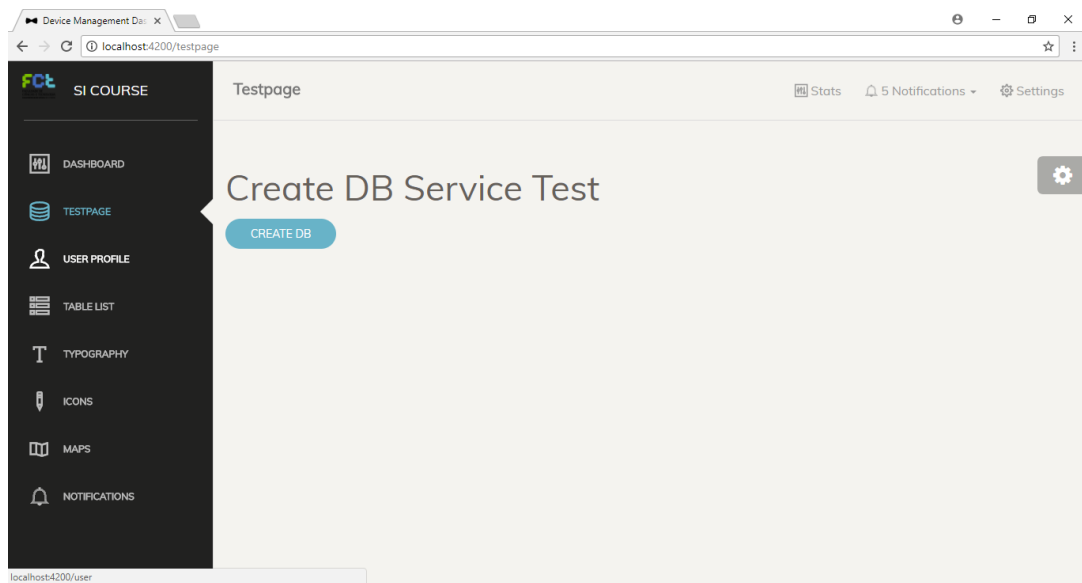


Figura 3 – Vista geral da página de teste

Para este efeito terá que trabalhar nos seguintes ficheiros:

- **testpage.service.ts:**
 1. Instancie o URL para o endpoint correcto no backend;
 2. Injete o cliente http no construtor;
 3. Implemente o método *createDB()*.
- **testpage.component.ts:**
 1. Injete o serviço desenvolvido no ponto anterior no construtor;
 2. Implemente o método *onClick()* que deverá ser chamado quando o utilizador pressiona o botão *Create DB*.
- **app.routing.ts:**
 1. Adicione a rota para a página de teste.
- **app.module.ts:**
 1. Importe os módulos *HttpModule* e *TestpageModule*.
 2. Declare o *TestpageService* no array de *providers*.
- **sidebar.component.ts:**

1. Adicione a *route info* relativa à página de teste de forma a esta ser corretamente adicionada ao menu de navegação lateral.

Finalmente, abra novamente a janela da consola na diretoria do projeto e execute o comando *ng serve* para lançar a aplicação. Abra o browser, navegue para o endereço <http://localhost:4200> e teste a aplicação. O resultado no Front-End e no Back-End deverá ser idêntico ao mostrado nas Figuras 4 e 5, respectivamente.

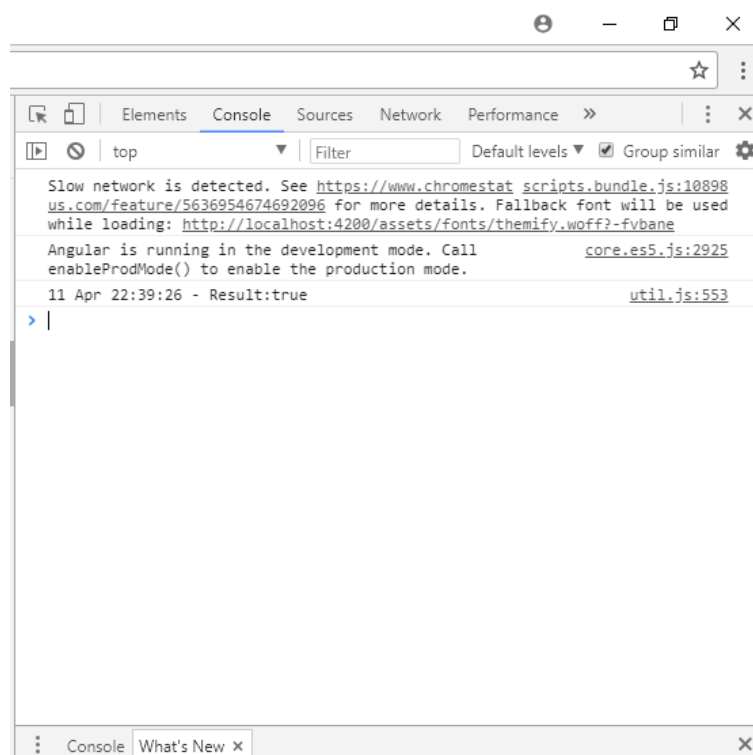


Figura 4 – Resultado na consola do Chrome Developer Tools (pressionando F12 no browser)

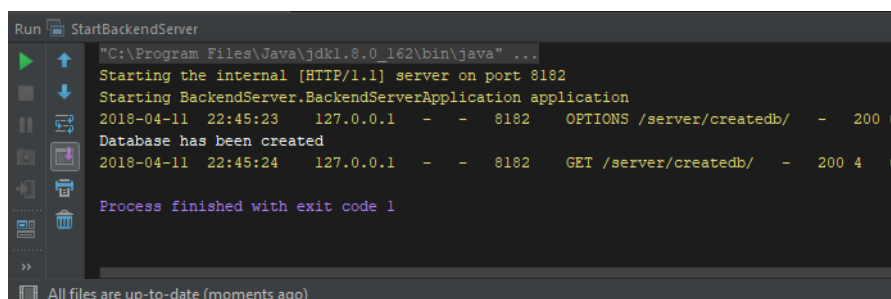


Figura 5 – Resultado da chamada a *createDB()* na consola do servidor Back-End

Aula 3 – Implementação do Front-End

1. Implemente as páginas do *Front-End* do consumidor.

O *front-end* deve possibilitar ao utilizador, **no mínimo**:

- Inscrever-se;
- Associar dispositivo;
- Operar dispositivo;
- Consultar dispositivos associados.

2. Implemente as páginas do *Front-end* do fabricante.

O *front-end* deve possibilitar ao fabricante, **no mínimo**:

- Criar a Base de Dados;
- Consultar historial de um cliente;
- Consultar historial de um dispositivo.

Aula 4 – Conclusão da Implementação do Front-End e Funcionalidades Extra

Funcionalidades extra

1. Crie um Web Service, que emula um pedido para operar um determinado dispositivo. Por exemplo, quando um utilizador pretende ligar um determinado dispositivo, este serviço deve ser chamado, pois é responsável por posteriormente operar o *hardware*, neste caso o dispositivo (possível integração com o primeiro trabalho).
2. Outras funcionalidades extra não definidas tais como autenticação (diferenciação das páginas mostradas a um consumidor e a um fabricante), serviços adicionais (e respetivas alterações na DB), etc.

5. Avaliação

A avaliação do trabalho tem a seguinte ponderação:

- Correta implementação e demonstração de funcionamento do trabalho previsto para as aulas 2, 3 e 4:
 - 16 valores
- Correta implementação e demonstração das funcionalidades extra definidas:
 - 2 valores
- Correta implementação e demonstração de funcionalidades extras não definidas:
 - 2 valores.

Material Auxiliar:

Web Services:

- **REST**

1. <http://pt.slideshare.net/philipmjohnson/introduction-to-rest-and-the-restlet-framework>
2. <http://restlet.com/technical-resources/restlet-framework/guide/2.2/introduction/first-steps/first-application>
3. <http://www.javacodegeeks.com/2013/09/restlet-framework-hello-world-example.html>
4. **Understanding and Using CORS**
<http://restlet.com/company/blog/2015/12/15/understanding-and-using-cors/>

Front-End:

- **Setup Node + NPM + Angular and Quickstart Tutorial:**
 1. <https://angular.io/guide/quickstart>
- **Angular Dashboard Template Live Demo:**
 1. https://pd-angular.creative-tim.com/?_ga=2.24280570.1139612434.1523453267-913462416.1523453267

Bases de Dados:

- **Bases de Dados e H2 Database Engine**
 1. <http://www.h2database.com/html/main.html>
 2. <http://www.h2database.com/html/tutorial.html>
 3. <http://java2s.com/Code/SQL/CatalogSQL.htm>

Docentes

José Barata jab@uninova.pt

Ricardo Peres ricardo.peres@uninova.pt

Tiago Cardoso tomfc@uninova.pt

