Redes de Computadores - TP3
PeeringDB REST API

André Fernandes La Rocca Teixeira - 2014004107

1- Introdução

O objetivo deste trabalho é criar uma API REST para servir dados do PeeringDB, e um programa cliente para consultar esses dados e exportar análises. Para isso, foi criado um par cliente-servidor, em que o servidor consulta arquivos que contém os dados, e os expõe através de *endpoints*. O programa cliente pode requisitar os dados através da chamada para a API e é responsável por criar duas análises.

As análises são: *IXPs por Rede*, que mostra a quantidade de *Internet Exchange Points* para cada uma das redes, acompanhada do identificador e nome da rede; *Redes por IXP*, responsável por mostrar a quantas redes um IXP está conectado.

As rotas disponíveis através do programa servidor são: /api/ix, responsável por retornar todos os dados de IXPs contidos no arquivo de dados; /api/ixnets/<ix_id>, rota que busca uma lista de identificadores de rede referentes ao identificador de IXP passado por parâmetro; e, finalmente, */api/netname/<net_id>*, que busca o nome da rede que possui o identificador passado por parâmetro.

2 - Instruções

Para executar o projeto, basta inicializar cada um dos programas com os seus respectivos parâmetros, conforme mostrado abaixo:

python server.py <porto> <path_to_net.json> <path_to_ix.json> <path_to_netixlan.json>

python client.py <endereco_servidor>:<porto> <opcao_analise>

O servidor executará continuamente até que seja manualmente parado, através do atalho *ctrl+c*. O programa cliente irá consultar os dados do servidor, preparar e exportar a análise e, então, terminar a sua execução.

3 - Detalhes de Implementação

3.1 - Programa Cliente

O programa cliente é responsável por mostrar as análises. Para isso, por padrão, o programa cliente solicita os dados de todos os IXPs e, para cada um deles, solicita todas as redes associadas. Em seguida, verifica a opção de análise passada pelo usuário, e executa as tarefas referentes, conforme detalhamento:

3.1.1 - IXPs por Rede

O programa cliente cria um novo dicionário de dados, indexado pelo código da rede. Então, para cada um dos IXPs, verifica todas as suas redes, e incrementa o número de ocorrências no novo dicionário. Ao final, percorre o novo dicionário, mostrando o identificador da rede, o nome da rede e o número de IXPs identificados para a rede.

3.1.2 - Redes por IXP

Para cada um dos IXPs buscados no início da execução, o programa cliente mostra o identificador, o nome, e o tamanho da lista de redes associadas. Neste caso, é possível mostrar toda a análise com somente uma iteração sobre os dados.

3.2 - Programa Servidor

O programa servidor inicia a sua execução recebendo os parâmetros passados pela linha de comando. Em seguida, usando os parâmetros de nome de arquivo, abre e carrega os JSONs de todos eles. Então, se conecta ao porto passado. Enfim, serve as rotas de dados, conforme descrito a seguir:

3.2.1 - */api/ix*

A rota é responsável por retornar todos os dados do arquivo *ix.json*, que contém os dados de todas as IXPs cadastradas. Para isso, pega os dados do arquivo, insere dentro do objeto de resposta, e envia de volta para o requisitante.

3.2.2 - */api/ixnets/<ix_id>*

É responsável por retornar todas as redes associadas ao identificador do IXP passado por parâmetro na rota. Para isso, busca em todos os dados do arquivo *netixlan.json*, os registros que contém identificador de IXP igual ao passado por parâmetro. Em seguida, adiciona o identificador de rede do registro à lista que será enviada como resposta. Depois de todos os registros percorridos, a lista é enviada de volta para o solicitante.

3.2.3 - */api/netname/<net_id>*

Essa rota é responsável por retornar o nome da rede cujo identificador foi passado por parâmetro. Para isso, consulta o arquivo *net.json*, procura o registro que possui o identificador passado, e envia o seu nome no objeto resposta.

4 - Conclusão

Foi possível simular situações reais através da experiência de criar uma API REST, e um cliente que se comunica com ela. Os dados foram enviados corretamente, possibilitando que as análises fossem feitas e exportadas com sucesso.