Web Service REST/JSON
André Vinício Resende Silva

+ Ferramentas utilizadas

- NetBeans 8.2
- pgAdmin3
- Android
- XAMPP

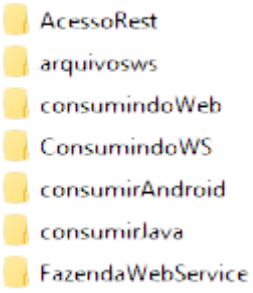
+ Especificação de cada ferramenta:

- NetBeans: Foi onde realizei as operações Restful para desenvolver o webservice.
- PgAdmin3 é minha base de dados utilizada
- Android irá fazer seu consumo no webservice
- XAMPP para fazer o acesso aos dados (para testes iniciais p/ conexão)

Todo o processo é executado por teste Restful por padrão do Netbeans, como o projeto inicial deu errado fiz este com algumas classes que já tinha em projetos de outras matérias no meu computador. Fiz o consumo do webservice em Java, web e Android e todos é apenas um retorno do usuário cadastrado com seus respectivos dados para cadastro. Para execução será necessário colocar os caminhos no código do webservice para sincronizar com o banco de dados.

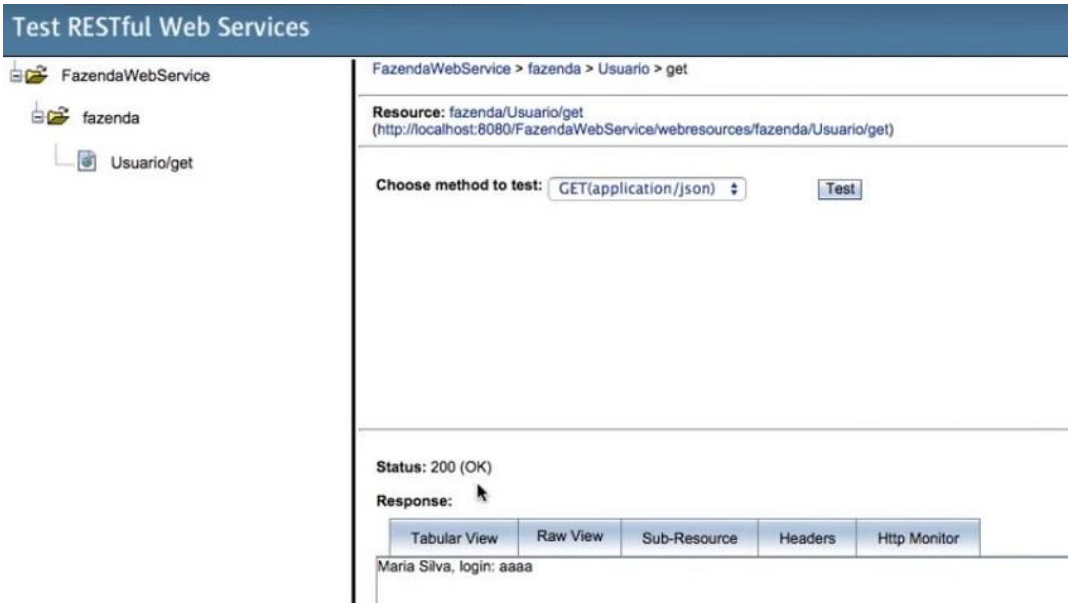
Dentro da pasta zipada segue alguns dados que utilizei para rodar as classes, já estão todas configuradas mais caso haja algum erro você pode importá-las novamente.

Estrutura do diretório:

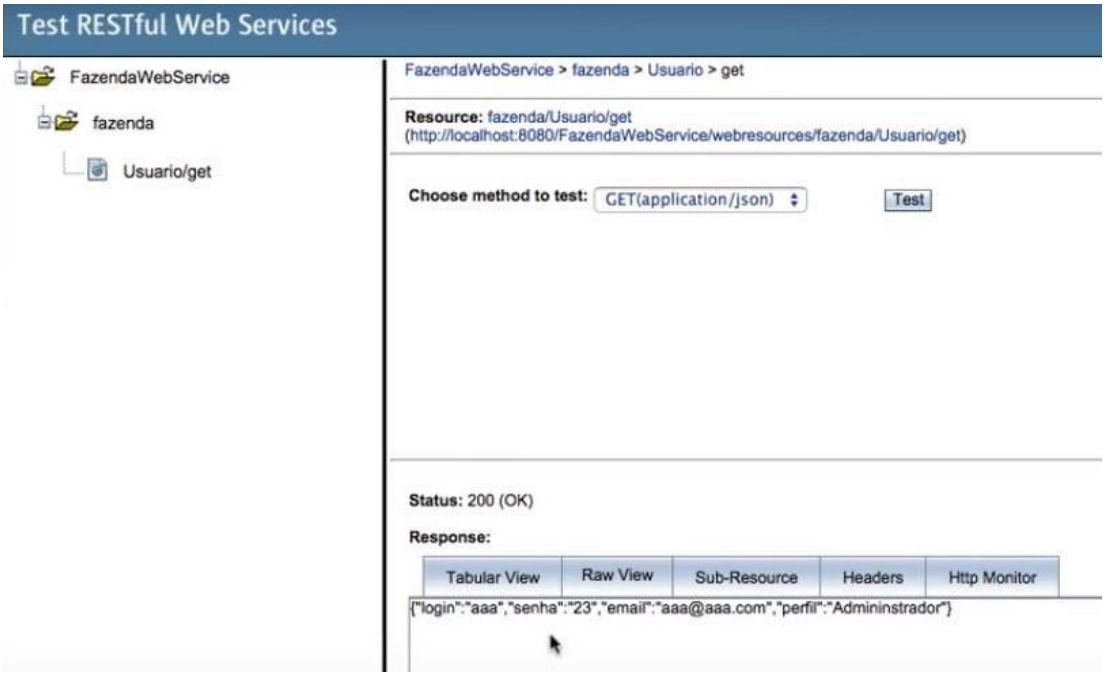


Os testes realizados já estão com os usuários cadastrados no banco, segue alguns exemplos:

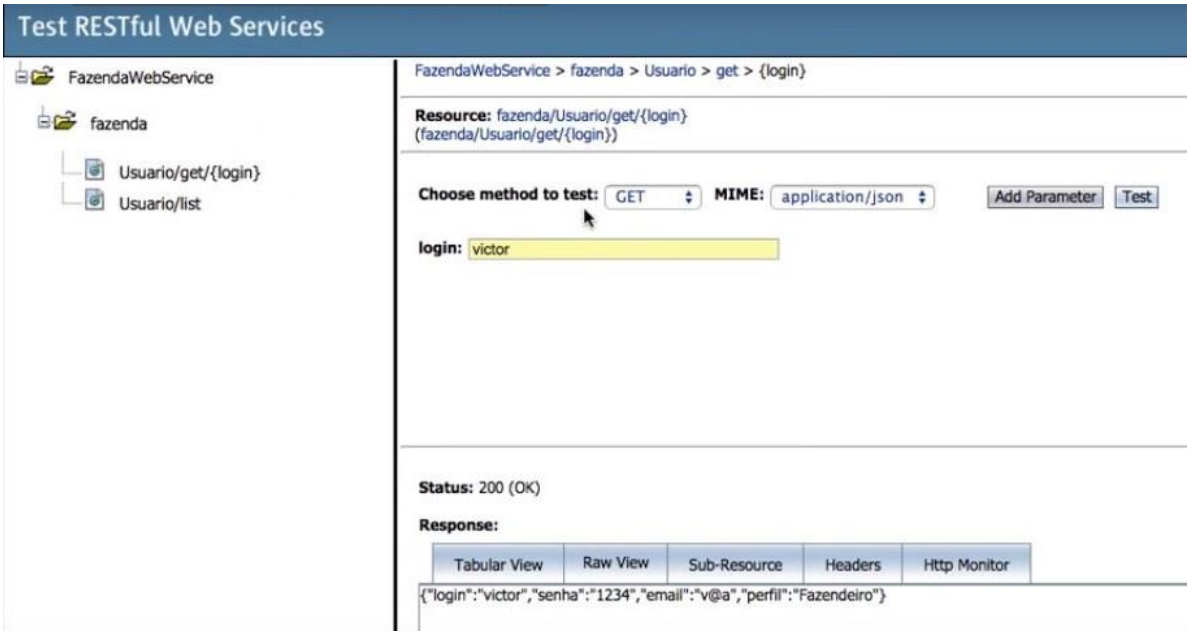
Este primeiro exemplo é apenas o retorno do banco, no caso sem nenhum tipo de estrutura configurada.



Já no segundo exemplo é gerado um JSON que está na minha parte Web Service sendo o segundo método o primeiro determinei como /text para exemplificar a diferença entre ambos e no exemplo abaixo coloquei /Json então por consequência já colocou no formato certo.



Já no consumo do Web Service com Java determinei no meu código a entrada pro login, poderia ser por login e senha mais para exemplificar coloquei somente por login. Victor está cadastrado no meu banco se eu colocar o usuário já consome pelo Web Service e no formato JSON.




```
@GET
@Produces("application/json")
@Path("Usuario/get/{login}")
public String getUsuario(@PathParam("login") String login)
{
    Usuario u = new Usuario();
    u.setLogin(login);

    UsuarioDAO dao = new UsuarioDAO();
    u = dao.buscar(u);

    //Converter para Gson
    Gson g = new Gson();
    return g.toJson(u);
}
```

Já no Android o App tem apenas o retorno do login do usuário, que no caso é Victor, poderia colocar todos os dados da tabela, porém gerou alguns erros e eu pra exemplificar coloquei apenas o usuário retornando no app consumindo o Web Service. No Android usei o JSONObject como forma de manipulação do JSON.

Conclusão:

O projeto em si deu bastante problemas, e o único retorno que eu consegui realmente sem erro nenhum foi o usuário. Para fazer esta aplicação rodar foi complicado, não é nada de muito grandioso a aplicação, na verdade é bem simples, mas eu consegui ter uma base mais ou menos para aprimorar.. Usei o REST/JSON como especificado por você e funcionou com o usuário. Consegui aprender o conceito, porém não ficou algo grandioso, mais espero que tenha feito como o especificado. Sobre a interface não fiz pois meu android realmente não roda alguns layouts, tentei resolver o problemas mas mediante a versão eu não consegui por isso fiz o retorno somente. No entanto tentei rodar novamente o projeto e já deu alguns erros de versões porém o projeto zipado foi o backup feito após terminar a aplicação.