**Observações**

1. Dentro do Projeto, pasta resources, tem um arquivo chamado ‘data.sql’. Nele tem o SQL de cadastro dos status Jedi e dos Jedis (inclusive do primeiro, que deve ser cadastrado manualmente)
2. No mesmo local do item anterior, tem um arquivo chamado ‘Padwan Api.postman\_collection.json’ com as requisições que podem ser usadas para os testes.

**Teste de conceito para desenvolvedores juniors**

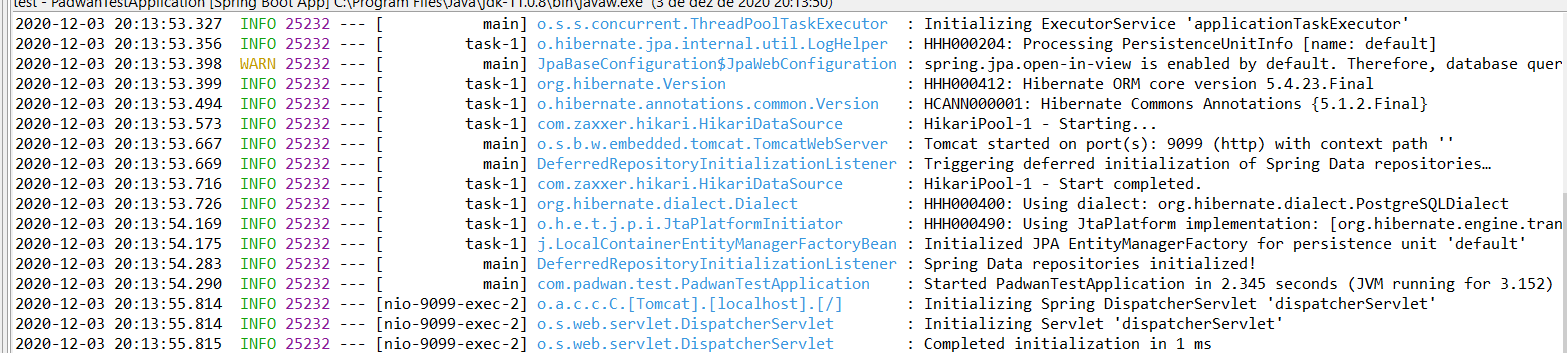
Olá, esse teste foi criado por: João Alcides e Glauco, somos líderes técnicos de duas squads de desenvolvimento da Pacto Soluções, mais especificamente: Squad Innovation e Squad AD, escrevemos esse teste para avaliar algumas competências suas. É simples, mas vai nos ajudar a entender seu nível de conhecimento de back-end, e também a sua desenvoltura com a resolução de problemas.

Primeiramente, clone o repositório <https://github.com/jarguimaraes/padwan-test.git>. Clonado o projeto, você terá uma aplicação em java, onde usamos o framework Spring Boot, e as dependências são gerenciadas com o maven. Se você não sabe sobre o assunto que estamos abordando, por favor, faça essa leitura: [https://spring.io/guides/gs/spring-boot](https://spring.io/guides/gs/spring-boot/).

Execute a aplicação. Ela vai subir na porta 9099. Acesse essa doc: <https://documenter.getpostman.com/view/116353/TVCZYW7D>. Veja como funciona e me responda algumas perguntas:

1. **Yoda**: Como primeira parte de seu treinamento, capaz você deverá ser para subir a aplicação;

**Feito**!



1. Qual será o retorno se eu fizer um **POST** em <http://localhost:9099/generic-post> passando o seguinte body?

{

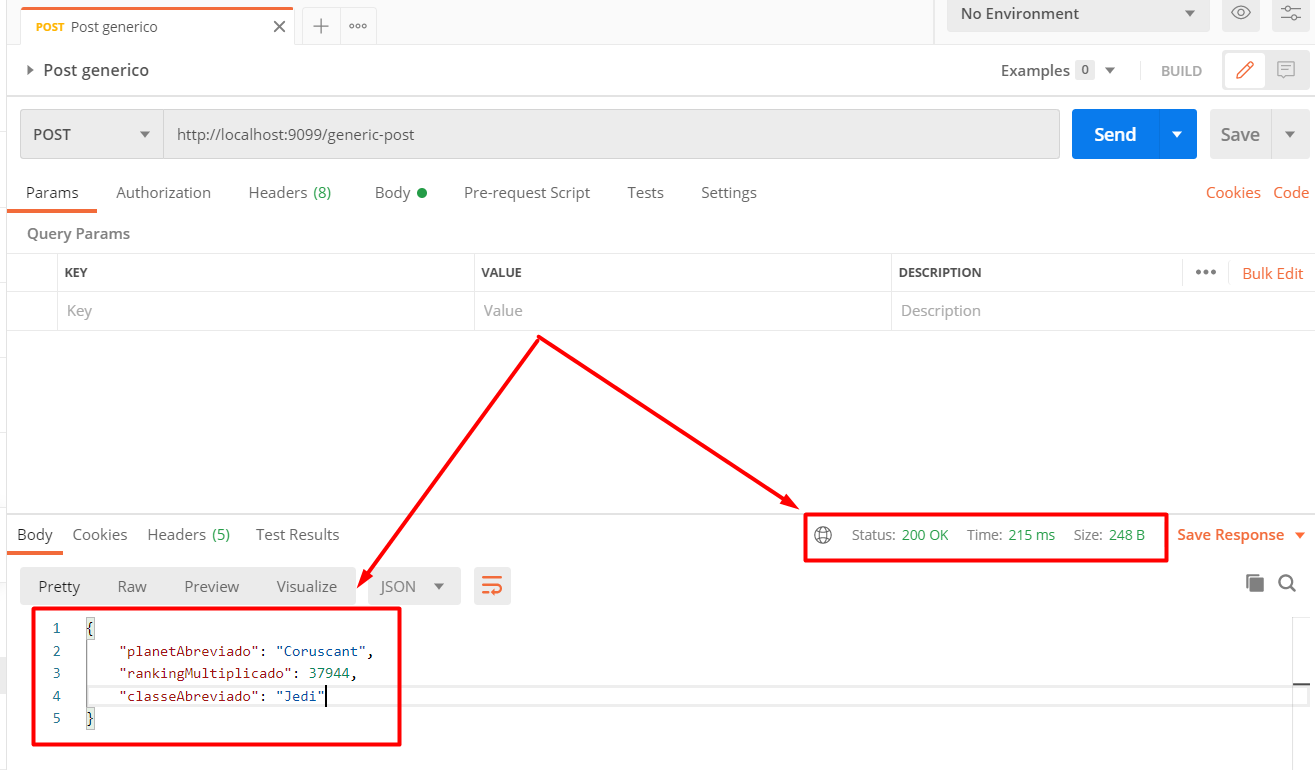
"planet": "Coruscant 985",

"ranking": 42,

"classe": "Jedi Knight"

}

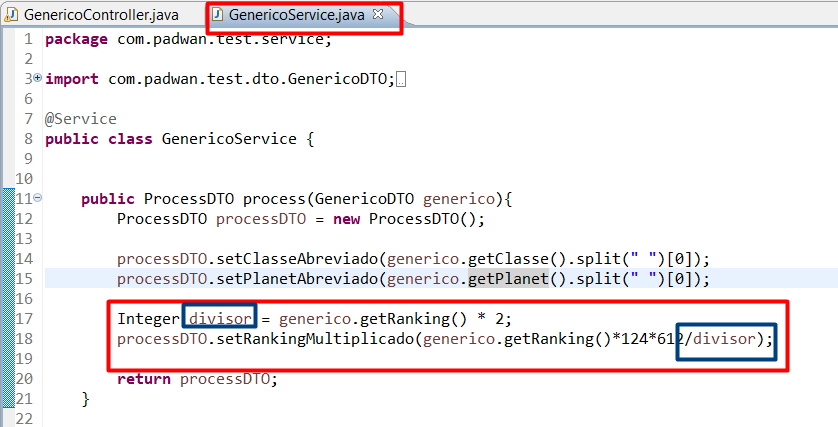
**Retorno:**



1. Se mudarmos o campo “ranking” para 0, a aplicação dá um erro. Por quê?

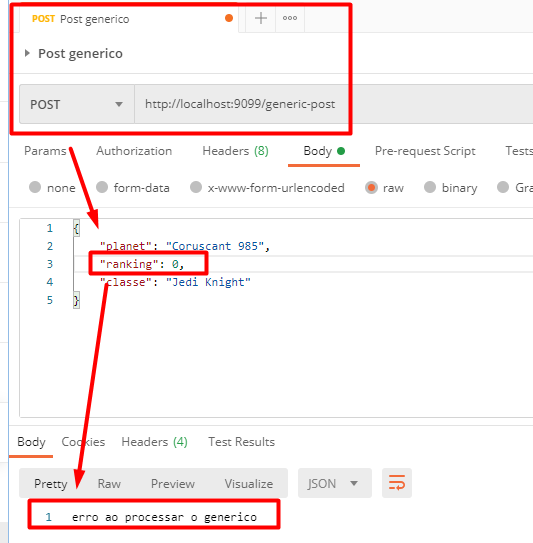
**Resposta:**

Por que, na Classe responsável por realizar o processamento (GenericoService), dentro do cálculo do Ranking Multiplicado, existe a operação de divisão pela variável “divisor”, que é calculada dividindo o ranking passado por 2:

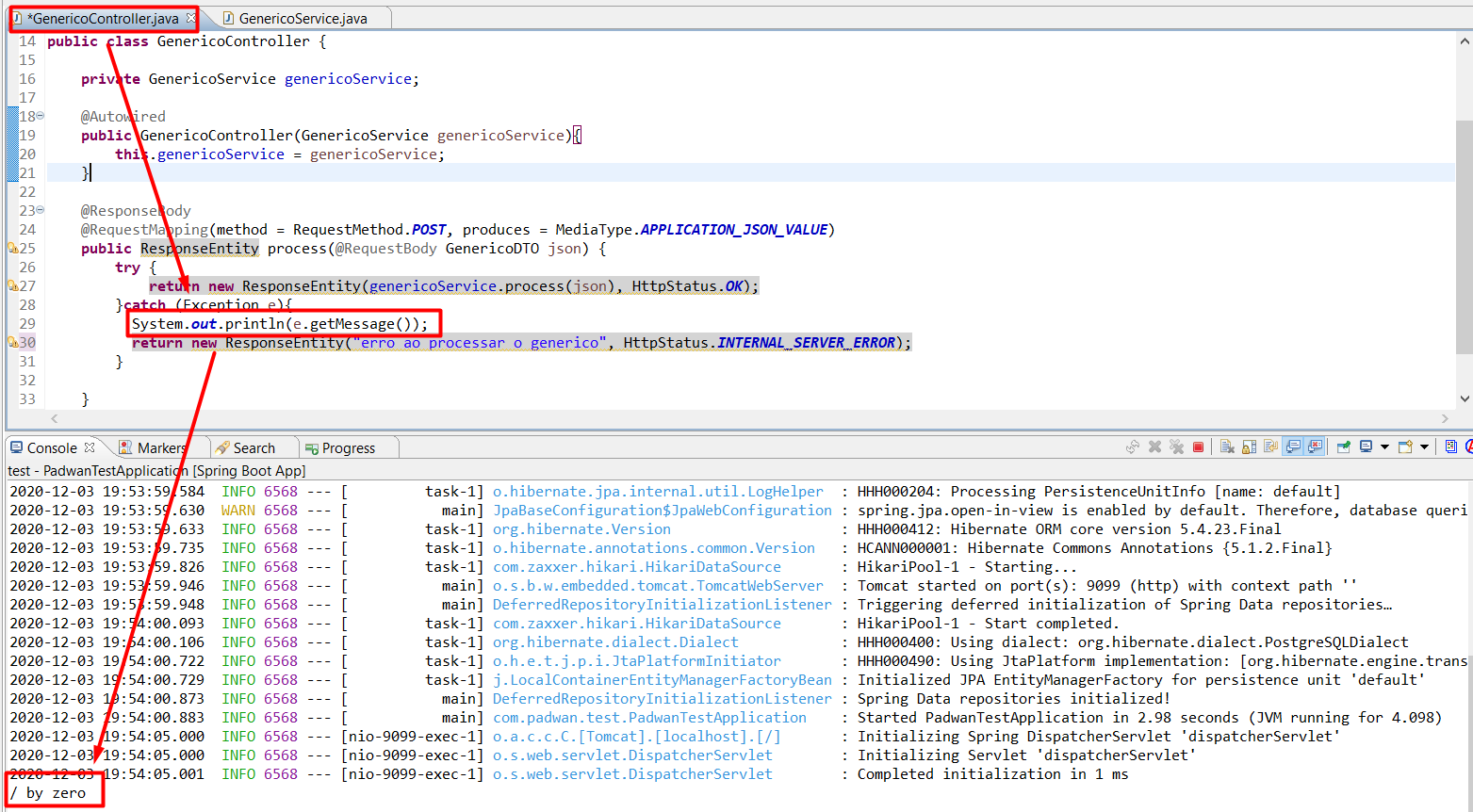


Se o ranking passado for 0, uma exceção é lançada, isso porque não é possível matematicamente se ter uma divisão por 0.

Esse é o retorno no Postman:

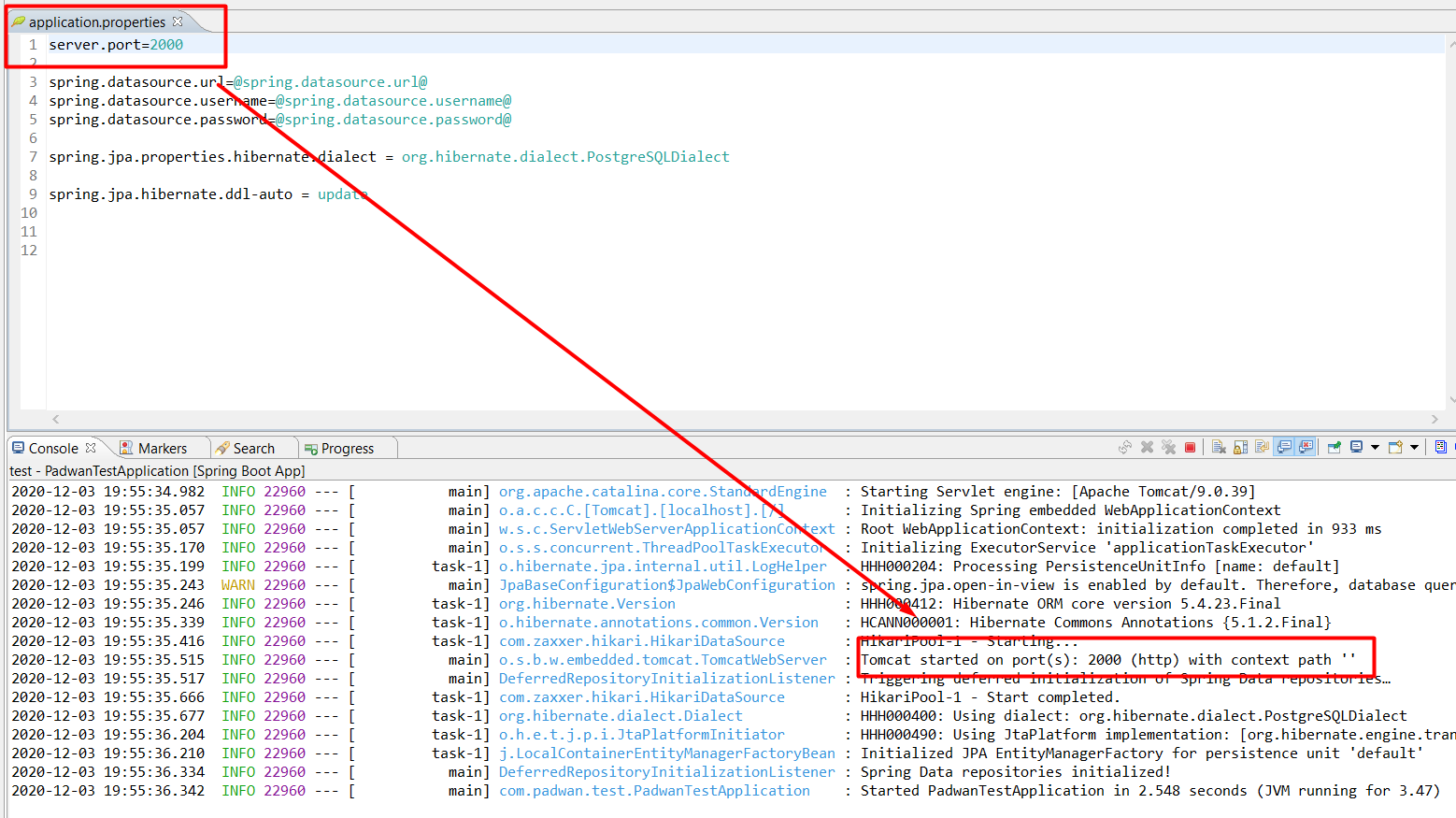


Expandindo o log de erro é possível verificar exatamente o erro:

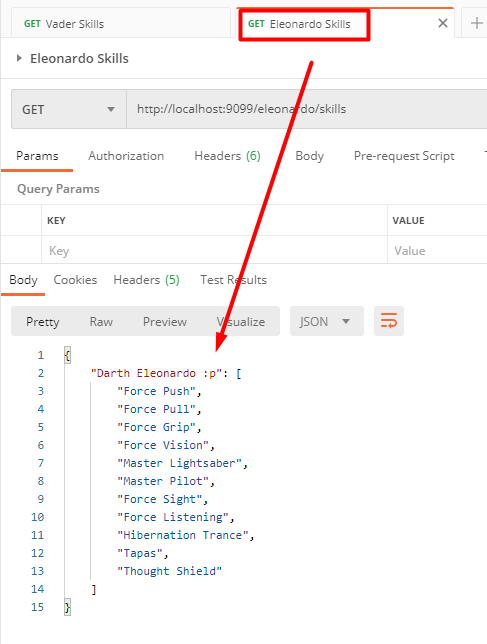


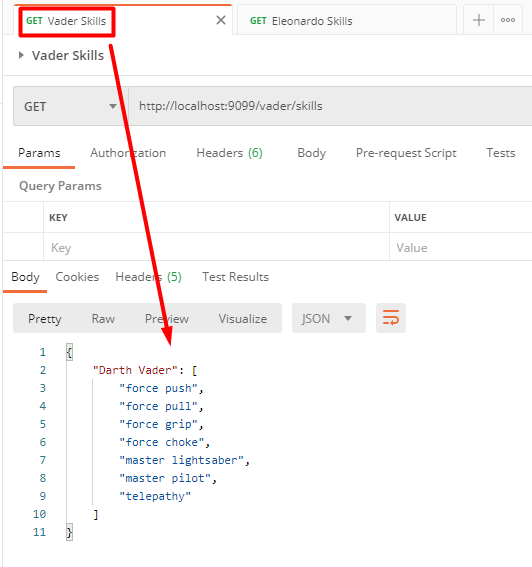
1. Como podemos mudar a porta onde a aplicação sobe?

Alterando a propriedade “server.port” no arquivo “application.properties”, conforme demonstrado no print abaixo:



1. Agora você vai ter que programar. Massa, hein?! Quando acessamos o endpoint <http://localhost:9099/vader/skills>, a aplicação retorna uma lista de habilidades do Darth Vader. Faça com que o sistema tenha um endpoint igual a esse, só que com seu nome e suas habilidades. Por exemplo, o meu seria [http://localhost:9099/joaoalcides/skills](http://localhost:9099/vader/skills). Detalhe muito importante: **ao fazer isso, você não pode impactar o do Darth Vader.** Envie para mim seu usuário do github, vou te adicionar nesse projeto para que você possa subir uma branch com suas alterações.





1. **Mestre Windu** explica que todo Jedi já foi um Padawan, então necessário é definir nome, status (padawan/jedi/mestre jedi) e um mentor para cada Jedi;

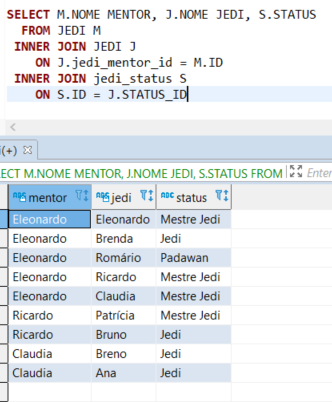
**Info:** É preciso o cadastro de um primeiro Mestre Jedi que não teve um mestre (afinal, alguém teve que ser autodidata pra aprender sobre a força lá no começo né?)



1. **Qui-Gon** ao fazer contato, estando no mundo espiritual da Força, nos diz que é possível mensurar a quantidade de midichlorians em cada Jedi, por isso, você deverá informar isso lá;



1. **Anakin Skywalker** antes de se tornar o Darth Vader, deixou para você a última missão, após isso, estará apto a se tornar um Jedi. Essa missão é a de criar, com Native SQL as seguintes consultas, com seus respectivos endpoints:
   1. Listar todos os mestres Jedis e seus aprendizes;



{

    "result": [

        {

            "jedi": "Eleonardo",

            "mentor": "Eleonardo",

            "status": "Mestre Jedi"

        },

        {

            "jedi": "Brenda",

            "mentor": "Eleonardo",

            "status": "Jedi"

        },

        {

            "jedi": "Romário",

            "mentor": "Eleonardo",

            "status": "Padawan"

        },

        {

            "jedi": "Ricardo",

            "mentor": "Eleonardo",

            "status": "Mestre Jedi"

        },

        {

            "jedi": "Claudia",

            "mentor": "Eleonardo",

            "status": "Mestre Jedi"

        },

        {

            "jedi": "Patrícia",

            "mentor": "Ricardo",

            "status": "Mestre Jedi"

        },

        {

            "jedi": "Bruno",

            "mentor": "Ricardo",

            "status": "Jedi"

        },

        {

            "jedi": "Breno",

            "mentor": "Claudia",

            "status": "Jedi"

        },

        {

            "jedi": "Ana",

            "mentor": "Claudia",

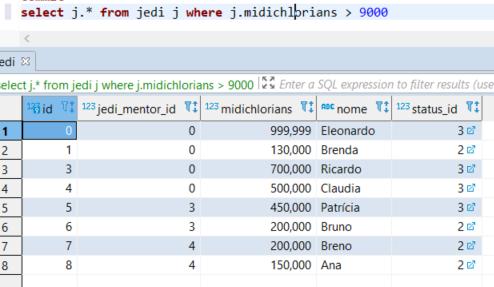
            "status": "Jedi"

        }

    ]

}

* 1. Listar todos Jedis cujo midichlorians sejam acima de 9000;



[

    {

        "id": 0,

        "nome": "Eleonardo",

        "status": {

            "id": 3,

            "status": "Mestre Jedi",

            "hibernateLazyInitializer": {}

        },

        "jediMentorId": 0,

        "midichlorians": 999999

    },

    {

        "id": 1,

        "nome": "Brenda",

        "status": {

            "id": 2,

            "status": "Jedi",

            "hibernateLazyInitializer": {}

        },

        "jediMentorId": 0,

        "midichlorians": 130000

    },

    {

        "id": 3,

        "nome": "Ricardo",

        "status": {

            "id": 3,

            "status": "Mestre Jedi",

            "hibernateLazyInitializer": {}

        },

        "jediMentorId": 0,

        "midichlorians": 700000

    },

    {

        "id": 4,

        "nome": "Claudia",

        "status": {

            "id": 3,

            "status": "Mestre Jedi",

            "hibernateLazyInitializer": {}

        },

        "jediMentorId": 0,

        "midichlorians": 500000

    },

    {

        "id": 5,

        "nome": "Patrícia",

        "status": {

            "id": 3,

            "status": "Mestre Jedi",

            "hibernateLazyInitializer": {}

        },

        "jediMentorId": 3,

        "midichlorians": 450000

    },

    {

        "id": 6,

        "nome": "Bruno",

        "status": {

            "id": 2,

            "status": "Jedi",

            "hibernateLazyInitializer": {}

        },

        "jediMentorId": 3,

        "midichlorians": 200000

    },

    {

        "id": 7,

        "nome": "Breno",

        "status": {

            "id": 2,

            "status": "Jedi",

            "hibernateLazyInitializer": {}

        },

        "jediMentorId": 4,

        "midichlorians": 200000

    },

    {

        "id": 8,

        "nome": "Ana",

        "status": {

            "id": 2,

            "status": "Jedi",

            "hibernateLazyInitializer": {}

        },

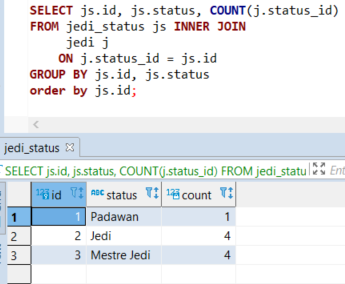
        "jediMentorId": 4,

        "midichlorians": 150000

    }

]

* 1. Listar por categoria, quantos são os Jedis;



{

    "result": [

        {

            "count": "1",

            "id": "1",

            "status": "Padawan"

        },

        {

            "count": "4",

            "id": "2",

            "status": "Jedi"

        },

        {

            "count": "4",

            "id": "3",

            "status": "Mestre Jedi"

        }

    ]

}

Um desenvolvedor júnior deve estar pronto para reconhecer as suas dificuldades e acima de tudo, saber perguntar. Dúvidas a respeito do item 4, seu usuário do github (se não tiver crie um, não tem problema estar vazio) e as respostas das questões, envie por e-mail para [alcides@pactosolucoes.com.br](mailto:alcides@pactosolucoes.com.br) e [glauco@pactosolucoes.com.br](mailto:glauco@pactosolucoes.com.br) com cópia para o e-mail julianalara@pactosolucoes.com br

Boa sorte, e até mais!

Que a força esteja com você.