

Tutorial Heroku

OBS: o nome da aplicação é única para cada usuário então crie um nome único para você.

Efetuando o acesso o Heroku

Configurando o git

```
git config --global user.email "suporte@jdevtreinamento.com.br"
git config --global user.name "Alex Fernando Egidio"
```

Login no Heroku

heroku login

-OBS: Informe o e-mail e senha da sua conta criada no Heroku

Vamos criar a aplicação dentro do Heroku

Criando o App

```
heroku create springboot-rest-api-sample
```

Vinculando o banco de dados PostgreSQL

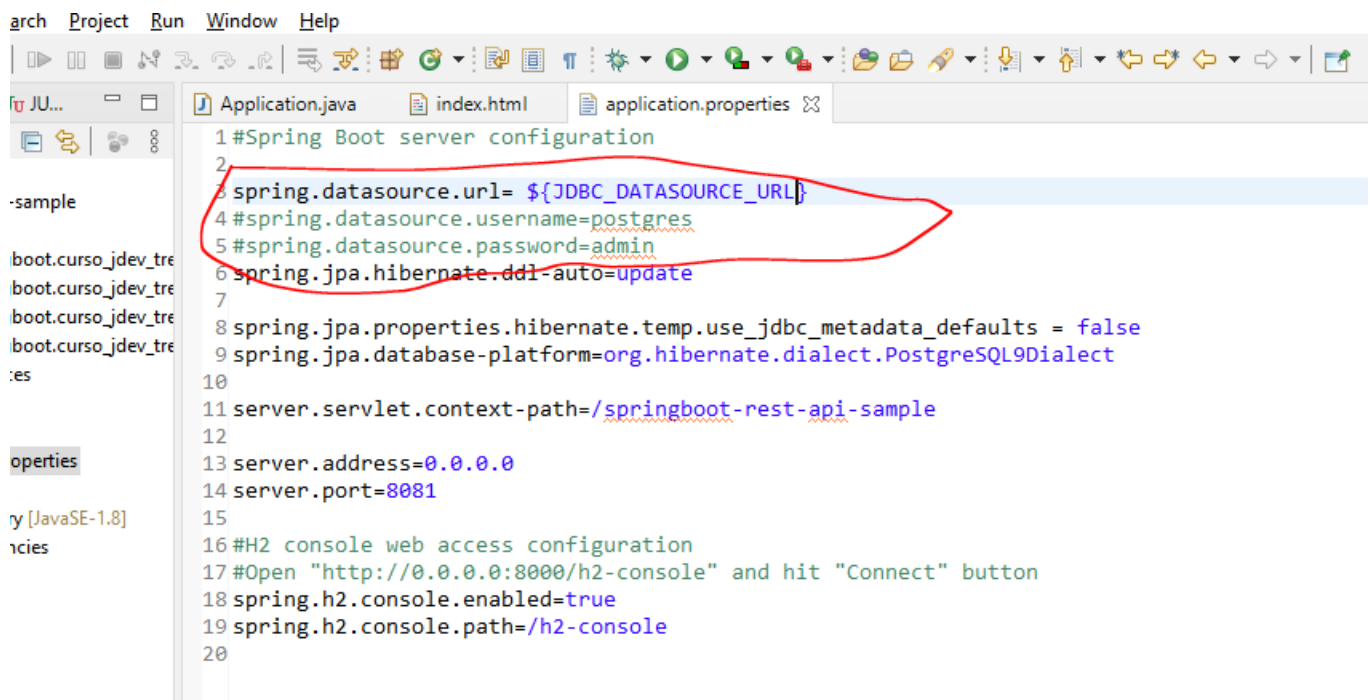
Configurando o banco de dados.

```
heroku addons:create heroku-postgresql:hobby-dev --app springboot-rest-api-sample
```

Configurando o arquivo application.properties.

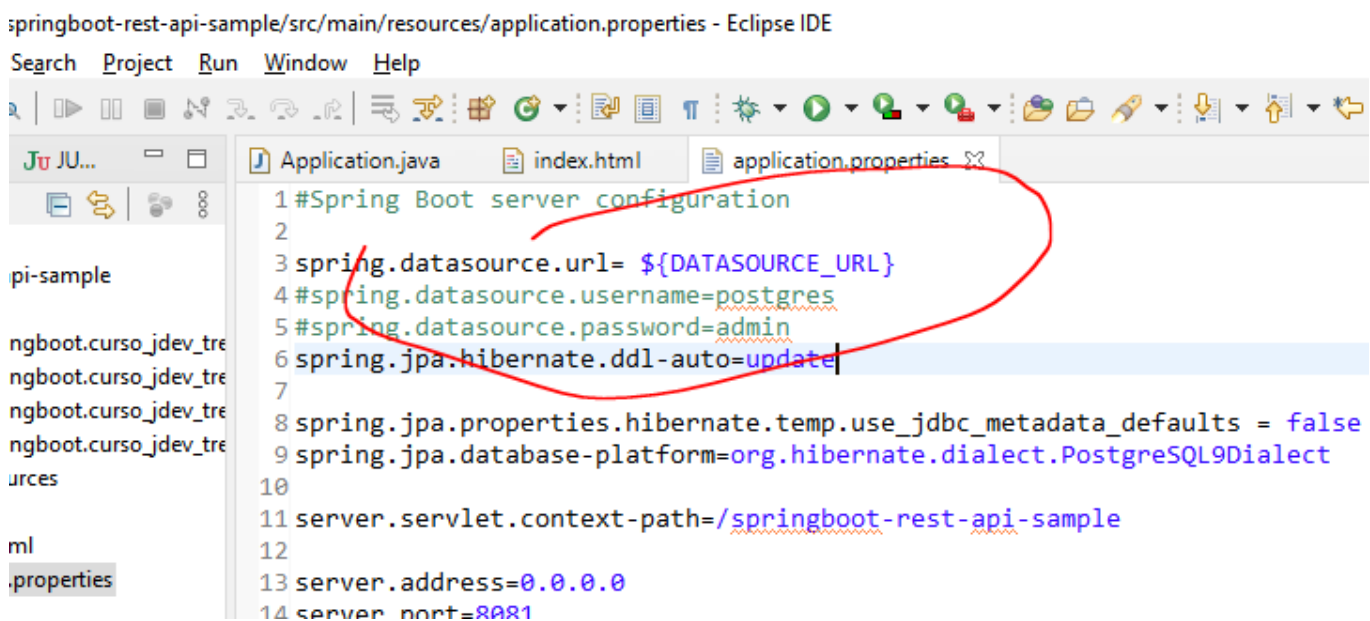
Antes de enviar o sistema para o Heroku temos que configurar o arquivo **application.properties** para identificar a URL do banco de dados que o Heroku vai disponibilizar automaticamente.

E também deixar comentado a linha de **user** e **senha** porque o próprio Heroku vai disponibilizar para a aplicação.



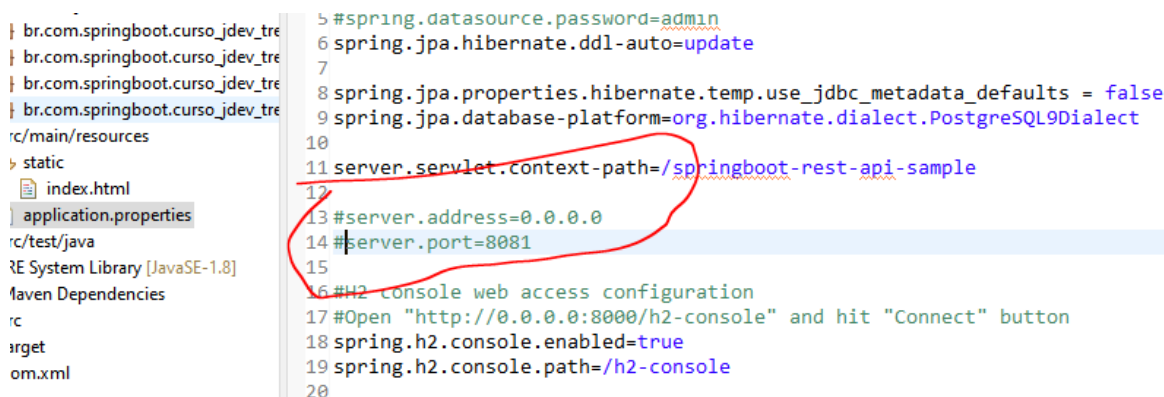
```
1 #Spring Boot server configuration
2
3 spring.datasource.url= ${JDBC_DATASOURCE_URL}
4 #spring.datasource.username=postgres
5 #spring.datasource.password=admin
6 spring.jpa.hibernate.ddl-auto=update
7
8 spring.jpa.properties.hibernate.temp.use_jdbc_metadata_defaults = false
9 spring.jpa.database-platform=org.hibernate.dialect.PostgreSQL9Dialect
10
11 server.servlet.context-path=/springboot-rest-api-sample
12
13 server.address=0.0.0.0
14 server.port=8081
15
16 #H2 console web access configuration
17 #Open "http://0.0.0.0:8080/h2-console" and hit "Connect" button
18 spring.h2.console.enabled=true
19 spring.h2.console.path=/h2-console
20
```

Ou



```
1 #Spring Boot server configuration
2
3 spring.datasource.url= ${DATASOURCE_URL}
4 #spring.datasource.username=postgres
5 #spring.datasource.password=admin
6 spring.jpa.hibernate.ddl-auto=update
7
8 spring.jpa.properties.hibernate.temp.use_jdbc_metadata_defaults = false
9 spring.jpa.database-platform=org.hibernate.dialect.PostgreSQL9Dialect
10
11 server.servlet.context-path=/springboot-rest-api-sample
12
13 server.address=0.0.0.0
14 server.port=8081
15
16 #H2 console web access configuration
17 #Open "http://0.0.0.0:8080/h2-console" and hit "Connect" button
18 spring.h2.console.enabled=true
19 spring.h2.console.path=/h2-console
20
```

E também comentar os as configurações de **servidores** no arquivo **application.properties**.

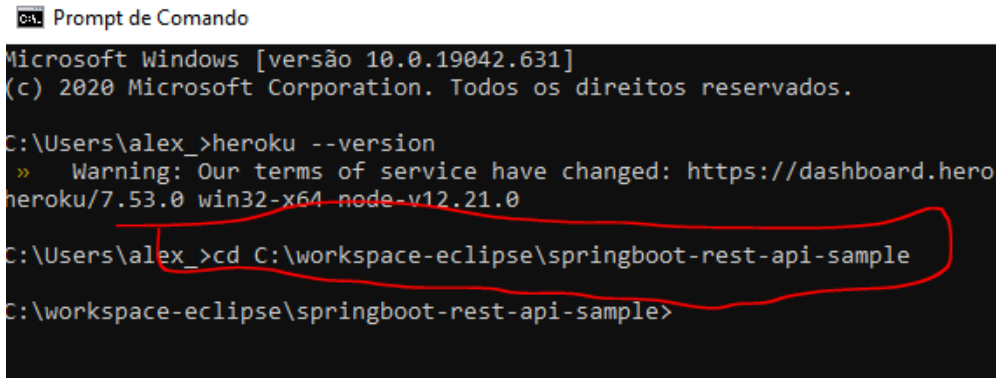


```
5 #spring.datasource.password=admin
6 spring.jpa.hibernate.ddl-auto=update
7
8 spring.jpa.properties.hibernate.temp.use_jdbc_metadata_defaults = false
9 spring.jpa.database-platform=org.hibernate.dialect.PostgreSQL9Dialect
10
11 server.servlet.context-path=/springboot-rest-api-sample
12
13 #server.address=0.0.0.0
14 #server.port=8081
15
16 #H2 console web access configuration
17 #Open "http://0.0.0.0:8080/h2-console" and hit "Connect" button
18 spring.h2.console.enabled=true
19 spring.h2.console.path=/h2-console
20
```

Commit da aplicação no Git

Vamos entrar na pasta do projeto pelo terminal CMD.

Cd C:\workspace-eclipse\springboot-rest-api-sample

A screenshot of a Windows Command Prompt window. The title bar says 'Prompt de Comando'. The text inside shows the Windows version (10.0.19042.631) and copyright (c) 2020 Microsoft Corporation. The user 'alex' has entered the command 'cd C:\workspace-eclipse\springboot-rest-api-sample', which has been executed, and the prompt has moved to the directory 'C:\workspace-eclipse\springboot-rest-api-sample'. A red circle highlights the command and the resulting directory path.

```
C:\Users\alex>heroku --version
» Warning: Our terms of service have changed: https://dashboard.heroku/7.53.0 win32-x64 node-v12.21.0

C:\Users\alex>cd C:\workspace-eclipse\springboot-rest-api-sample
C:\workspace-eclipse\springboot-rest-api-sample>
```

Iniciar o Git pelo CMD

git init

Adicionar todo o projeto ao Git

git add .

Comitando o projeto.

Git commit -m "Commit aplicacao"

Acionando o git remoto

heroku git:remote -a springboot-rest-api-sample

Efetutando o Push para o Heroku

git push heroku master

Abrir a aplicação

Abrir no navegador

heroku open

Log de Erro

Para vermos os log de erro no Heroku.

heroku logs --tail

Removendo aplicação

heroku apps:destroy --app springboot-rest-api-sample --confirm springboot-rest-api-sample

heroku git:remote -a springboot-rest-api-sample

<https://http://springboot-rest-api-sample.herokuapp.com/springboot-rest-api-sample/index.html>

<https://git.heroku.com/springboot-rest-api-sample.git>