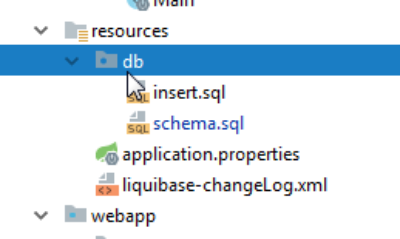
**0. Spring Liquibase [#309058]**

В этом уроке мы подключим liquibase к проекту forum.

Добавим зависимости.

<**dependency**>  
 <**groupId**>org.liquibase</**groupId**>  
 <**artifactId**>liquibase-core</**artifactId**>  
 <**version**>3.6.2</**version**>  
</**dependency**>

Перенесем папку db в директорию resources/

[](https://job4j.ru/TrackStudio/download/task/302330/0873958f726a9988017288986e300cad?type=image)

schema.sql

**create** **table** posts (  
 **id** serial primary **key**,  
 **name** varchar(2000),  
 description text,  
 created timestamp **without** time zone **not** **null** **default** **now**()  
);

insert.sql

**insert** **into** posts (**name**) **values** (**'О чем этот форум?'**);  
**insert** **into** posts (**name**) **values** (**'Правила форума.'**);

liquibase-changeLog.xml

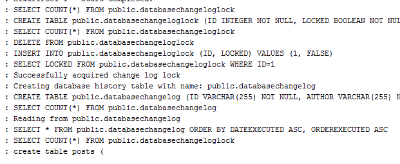
<?xml version="1.0" encoding="UTF-8"?>  
<**databaseChangeLog**  
 xmlns=**"http://www.liquibase.org/xml/ns/dbchangelog"**  
 xmlns:xsi=**"http://www.w3.org/2001/XMLSchema-instance"**  
 xmlns:ext=**"http://www.liquibase.org/xml/ns/dbchangelog-ext"**  
 xsi:schemaLocation=**"http://www.liquibase.org/xml/ns/dbchangelog http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-3.1.xsd**  
 **http://www.liquibase.org/xml/ns/dbchangelog-ext http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-ext.xsd"**>  
 <**include** file=**"db/schema.sql"** relativeToChangelogFile=**"true"**/>  
 <**include** file=**"db/insert.sql"** relativeToChangelogFile=**"true"**/>  
</**databaseChangeLog**>

Liquibase будет запускаться каждый раз при старте приложения.

Этого эффекта мы добьемся через Spring.

**package** ru.job4j.forum;  
  
**import** liquibase.integration.spring.SpringLiquibase;  
**import** org.springframework.boot.SpringApplication;  
**import** org.springframework.boot.autoconfigure.SpringBootApplication;  
**import** org.springframework.boot.builder.SpringApplicationBuilder;  
**import** org.springframework.boot.web.servlet.support.SpringBootServletInitializer;  
**import** org.springframework.context.annotation.Bean;  
  
**import** javax.sql.DataSource;  
  
@SpringBootApplication  
public **class** Main **extends** SpringBootServletInitializer {  
 @Override  
 **protected** SpringApplicationBuilder configure(SpringApplicationBuilder application) {  
 **return** application.sources(Main.**class**);  
 }  
 @Bean  
 **public** SpringLiquibase liquibase(DataSource ds) {  
 SpringLiquibase liquibase = **new** SpringLiquibase();  
 liquibase.setChangeLog(**"classpath:liquibase-changeLog.xml"**);  
 liquibase.setDataSource(ds);  
 **return** liquibase;  
 }  
 **public** **static** **void** main(String[] args) {  
 SpringApplication.run(Main.**class**, args);  
 }  
}

Перед запуском удалите старую таблицу posts из базы данных.

[](https://job4j.ru/TrackStudio/download/task/302330/0873958f726a9988017288986e7b0cae?type=image)

**1. Развернуть приложение на heroku [#309056]**

В этом уроке мы запустим приложение "форум " в облаке.

Для этого мы будем использовать сервис Hekoru.

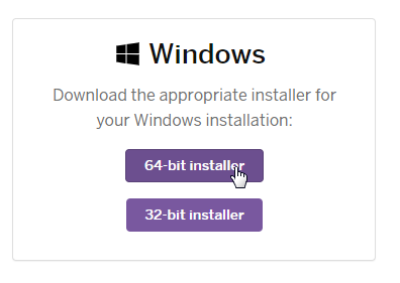
Зарегистрируйте на этом сайте.

<https://devcenter.heroku.com/articles/getting-started-with-java> 

**Клиент hekoru.**

Установите клиента hekoru. Это консольное приложение.

<https://devcenter.heroku.com/articles/getting-started-with-java#set-up>

[](https://job4j.ru/TrackStudio/download/task/49382/0873958f726a99880172889b2e360dc2?type=image)

После установки выполните команду

**heroku** login

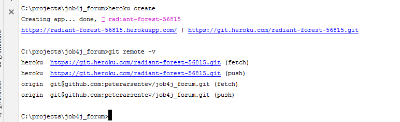
Команда откроет браузер. Далее введите логин и пароль.

**Запуск приложения.**

Откройте терминал IDEA и полните команду

heroku create

Это команда создаст git репозиторий для heroku.

[](https://job4j.ru/TrackStudio/download/task/49382/0873958f726a9988017288e4bd5e1dc2?type=image)  
  
Загрузим изменения.

**git** push heroku master

**Создание базы данных.**

В сервис hekoru подключена база данных postgresql.

**heroku** addons

[](https://job4j.ru/TrackStudio/download/task/49382/0873958f726a9988017288e4bdba1dc3?type=image)

Получим URL для работы с этой базой.

heroku config

[https://job4j.ru/TrackStudio/TSImageServlet?attId=0873958f726a9988017288e4bdda1dc4&width=400&height=400](https://job4j.ru/TrackStudio/download/task/49382/0873958f726a9988017288e4bdda1dc4?type=image)

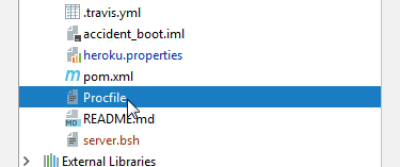
В корне проекта создайте файл heroku.properties

spring.main.banner-mode=**off**  
  
*## boot*  
*spring.mvc.view.prefix: /WEB-INF/views/*  
*spring.mvc.view.suffix: .jsp*  
  
*## datasource*  
*spring.datasource.url=postgres://*[*qyruajrfbgrmnj:bc4289b1b3f36024cd04b8db3774a51fc8f3692f5a857b18c31225eb9821936e@ec2-3-222-150-253.compute-1.amazonaws.com*](mailto:qyruajrfbgrmnj:bc4289b1b3f36024cd04b8db3774a51fc8f3692f5a857b18c31225eb9821936e@ec2-3-222-150-253.compute-1.amazonaws.com)*:5432/d6dtgdstch3uk5*  
  
*server.port=${PORT:5000}*

url - это ссылка DATA\_BASE\_URL из команды heroku config

**Настройка загрузки.**

Создайте в корне проекта файл Procfile.

[](https://job4j.ru/TrackStudio/download/task/49382/0873958f726a9988017288e4bdf31dc5?type=image)

hekoru использует этот файл для запуска приложения.

Содержимое файл Procfile

web: java -jar target/forum-1.jar --spring.config.location=heroku.properties

Приложение готова к запуска в облаке. Выполните команды

git add .  
git commit -m **"add Procfile"**  
git push origin master  
git push heroku master

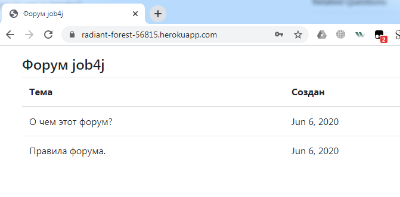
После этого выполните команду, которая покажет процесс лог загрузки.

**heroku** logs

Выполните команду

heroku **open**

Ваше приложение готово.

[](https://job4j.ru/TrackStudio/download/task/49382/0873958f726a9988017288e4be0b1dc6?type=image)

**2. Travis и Heroku [#309057]**

В этом уроке мы интегрируем Travis и Heroku.

Вначале нужно поставить Travis CLI - это консольное приложения для travis.

Инсталляционного пакета этого приложения нет. Поэтому его нужно собрать из исходников.

Написан он на Ruby.

Сперва поставим компилятор Ruby.

<https://rubyinstaller.org/>

Выполним команду

**gem** install travis

Теперь нужно получить ключи доступа к Heroku.

Создайте в корне проекта файл .travis.yml

language: java  
sudo: false  
  
install: true  
  
script: mvn clean package -DskipTests=true  
  
notifications:  
 email:  
 recipients:  
[-parsentev@yandex.ru](mailto:-parsentev@yandex.ru)  
 on\_success: never # default: change  
 on\_failure: always # default: always

Важно. Генерацию ключа нужно делать только в клиенте Git Bash.

Перейдите в приложении job4j\_forum выполните команду.

travis encrypt $(heroku auth:token) --add deploy.api\_key

Откройте .travis.yml и в блоке deploy добавьте ключи: provider, app.

language: java  
sudo: false  
install: true  
script: mvn clean package -DskipTests=true  
notifications:  
 email:  
 recipients: "[-parsentev@yandex.ru](mailto:-parsentev@yandex.ru)"  
 on\_success: never  
 on\_failure: always  
deploy:  
 provider: heroku  
 api\_key:  
 secure:   
 app: radiant-forest-56815

Имя приложения - это имя репозиторий на heroku

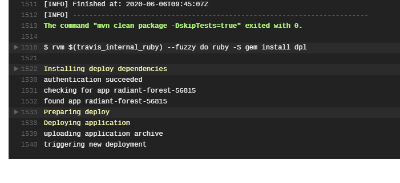
git remote -v

[](https://job4j.ru/TrackStudio/download/task/296072/0873958f726a9988017288fd7a4d2502?type=image)

Зайдите на сайт travis и включите интеграцию репозиторий job4j\_forum с Travis.

[https://job4j.ru/TrackStudio/TSImageServlet?attId=0873958f726a9988017288fd7a922503&width=400&height=400](https://job4j.ru/TrackStudio/download/task/296072/0873958f726a9988017288fd7a922503?type=image)

Загрузите изменения на github. Откройте лог сборки в Travis.

[](https://job4j.ru/TrackStudio/download/task/296072/0873958f726a9988017289065e982805?type=image)

Теперь приложение обновляется автоматически.