Tutorial Listando Chaves Disponiveis

Projeto: Controle de empréstimo de chaves

Tecnologias: Spring Boot Rest e Svelte

Alunos:
Gabriela Silva Rodrigues;
Gustavo Machado Pontes;
Mateus Alves Silva;
Nathan Rodrigues dos Santos.

Iniciar o container

- Abra o terminal e digite o seguinte comando para iniciar o container com o mongo:

  sudo docker start mongo

Configuração de acesso ao banco e modificação da porta do server

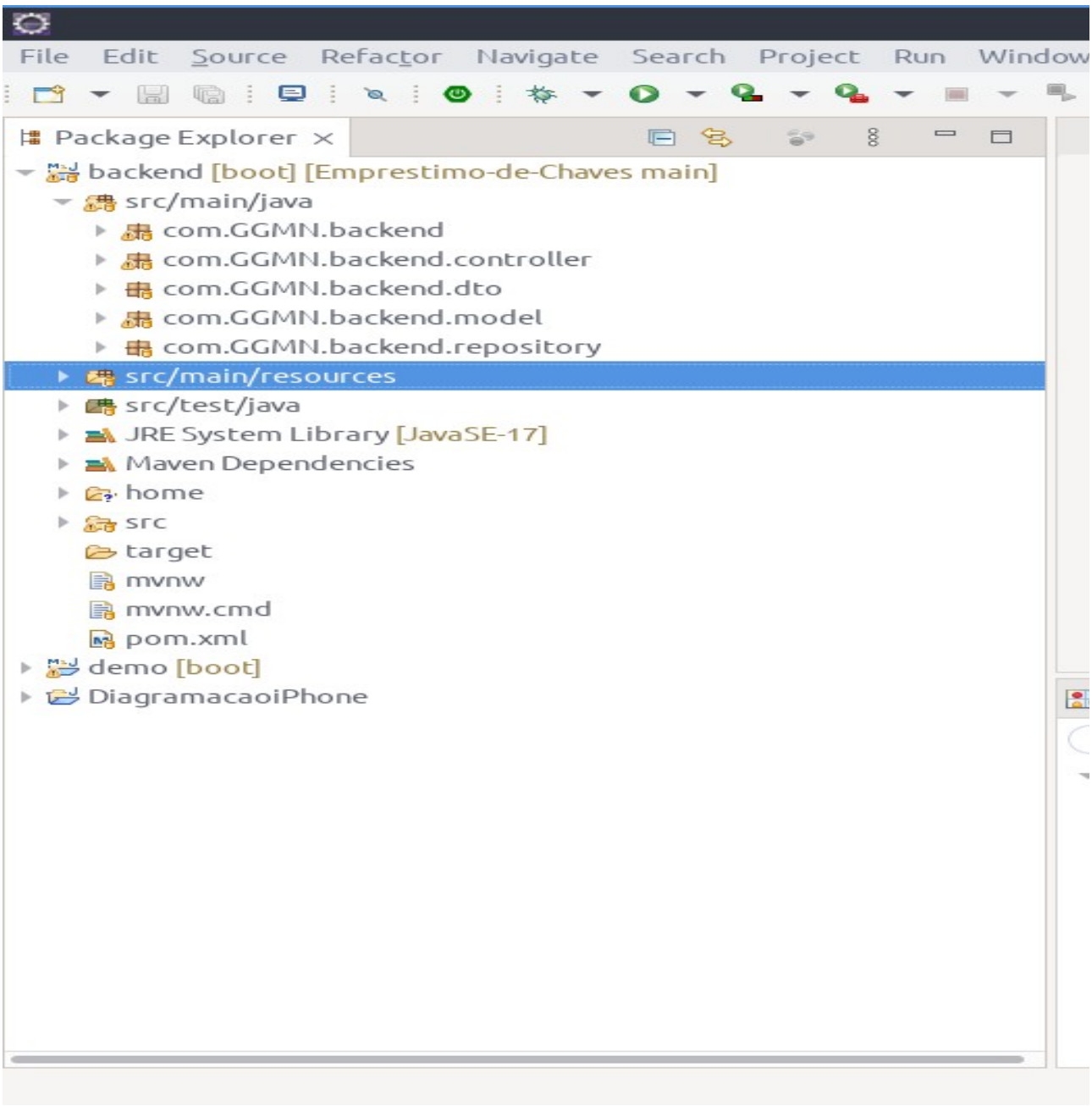
- Abra o projeto no eclipse e procure o arquivo application.properties localizados na pasta src/main/resources e digite:

```
spring.data.mongodb.host=localhost
spring.data.mongodb.port=27017
spring.data.mongodb.database=emprestimo_chaves
server.port=8081
```

- Essas configurações configuram o host, porta e a database do Banco de dados MongoDB a última linha modifica a porta do servidor Tomcat que está embutido no SpringBoot para que a backend e frontend não utilizem a mesma porta;

Criação do DTO, Repository e Controller

- Crie os seguinte arquivos nos packages



- Crie os seguinte arquivos dentro dos packages:
  - Um Record dentro do package dto com o nome ChaveRecordoDto para armazenar temporariamente os dados com o seguinte conteudo:

```
package com.GGMN.backend.dto;
public record ChaveRecordDto(String nome, String situacao,  boolean status) {

}
```

  - Uma interface no package repository com o nome ChaveRepository que herda os métodos do MongoRepositoy que contem os comandos de persistência e busca no mongodb, digite o seguinte conteúdo:

```
package com.GGMN.backend.repository;

import java.util.List;

import org.springframework.data.mongodb.repository.MongoRepository;
```

```
import org.springframework.stereotype.Repository;

import com.GGMN.backend.model.Chave;

@Repository
public interface ChaveRepository extends MongoRepository<Chave, String>{
    List<Chave> findByStatus(boolean status);
    List<Chave> findBySituacao(String situacao);
}
```

- Uma classe com o nome de ChaveController com o seguinte conteúdo:

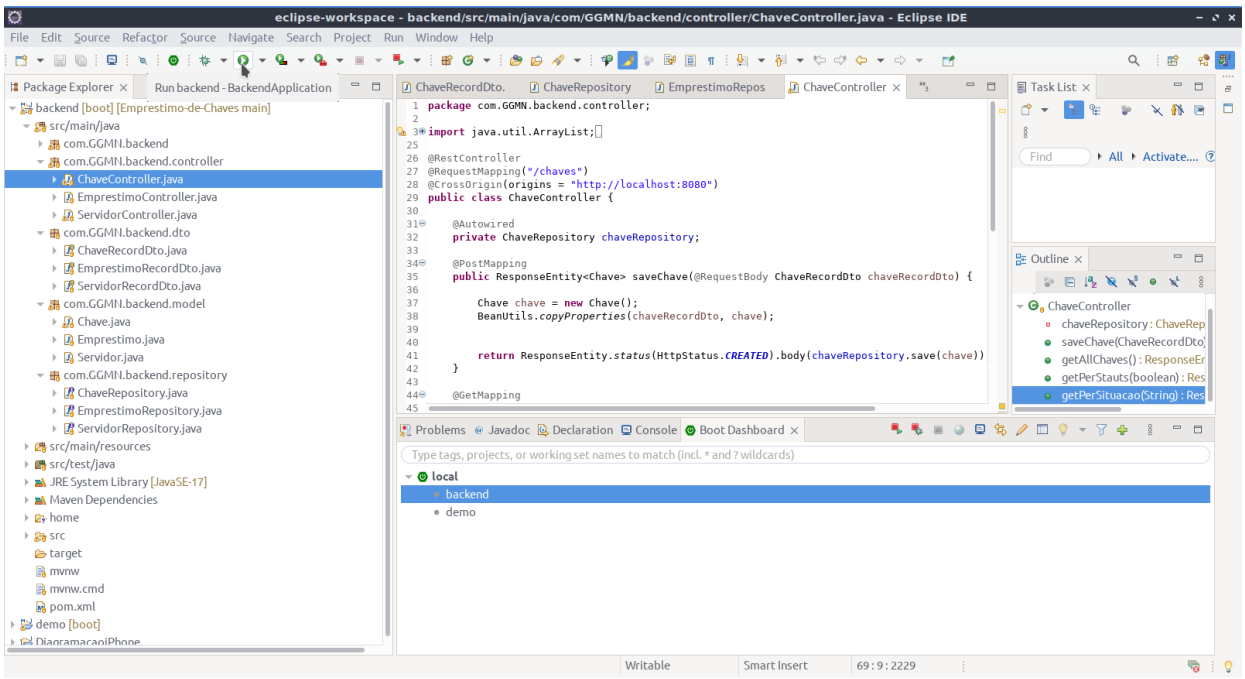
```
package com.GGMN.backend.controller;

import java.util.ArrayList;
import java.util.List;

import org.springframework.beans.BeanUtils;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.HttpStatusCode;
import org.springframework.http.ResponseEntity;
import org.springframework.stereotype.Controller;
import org.springframework.stereotype.Service;
import org.springframework.validation.annotation.Validated;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.GGMN.backend.dto.ChaveRecordDto;
import com.GGMN.backend.model.Chave;
import com.GGMN.backend.repository.ChaveRepository;

@RestController
@RequestMapping("/chaves")
@CrossOrigin(origins = "http://localhost:8080")
public class ChaveController {

    @Autowired
    private ChaveRepository chaveRepository;

    @PostMapping
    public ResponseEntity<Chave> saveChave(@RequestBody ChaveRecordDto
chaveRecordDto) {

            Chave chave = new Chave();
            BeanUtils.copyProperties(chaveRecordDto, chave);
```

```
            return
ResponseEntity.status(HttpStatus.CREATED).body(chaveRepository.save(chave));
    }

    @GetMapping
    public ResponseEntity<List<Chave>> getAllChaves() {

            return
ResponseEntity.status(HttpStatus.OK).body(chaveRepository.findAll());
    }

    @GetMapping("/status/{status}")
    public ResponseEntity<List<Chave>> getPerStauts(@PathVariable(value="status")
boolean status) {


            return
ResponseEntity.status(HttpStatus.OK).body(chaveRepository.findByStatus(status));
    }

    @GetMapping("/situacao/{situacao}")
    public ResponseEntity<List<Chave>>
getPerSituacao(@PathVariable(value="situacao") String situacao) {

            List<Chave> chaves = new ArrayList<>();

            for (Chave chave : chaveRepository.findBySituacao(situacao)) {
                    if(chave.getStatus() == true) {
                            chaves.add(chave);
                    }
            }



            return ResponseEntity.status(HttpStatus.OK).body(chaves);
    }
}
```

- Esse arquivo contem o tratamento de cada chamado da API fazendo a ligação do frontend com o banco de dados.

iniciando o servidor

- clique o ícone de play na barra superior senão funcionar clique na seta ao lado e selecione seu projeto:


eclipse-workspace - backend/src/main/java/com/GGMN/backend/controller/ChaveController.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
Package Explorer
Run backend - BackendApplication
backend [boot] [Emprestimo-de-Chaves main]
src/main/java
com.GGMN.backend
com.GGMN.backend.controller
ChaveController.java
EmprestimoController.java
ServidorController.java
com.GGMN.backend.dto
ChaveRecordDto.java
EmprestimoRecordDto.java
ServidorRecordDto.java
com.GGMN.backend.model
Chave.java
Emprestimo.java
Servidor.java
com.GGMN.backend.repository
ChaveRepository.java
EmprestimoRepository.java
ServidorRepository.java
src/main/resources
src/test/java
JRE System Library [JavaSE-17]
Maven Dependencies
home
src
target
mvnw
mvnw.cmd
pom.xml
demo [boot]
DiagramacaoIPhone
ChaveRecordDto.
ChaveRepository
EmprestimoRepos
ChaveController
1 package com.GGMN.backend.controller;
2
3 import java.util.ArrayList;
25
26 @RestController
27 @RequestMapping("/chaves")
28 @CrossOrigin(origins = "http://localhost:8080")
29 public class ChaveController {
30
31 @Autowired
32 private ChaveRepository chaveRepository;
33
34 @PostMapping
35 public ResponseEntity<Chave> saveChave(@RequestBody ChaveRecordDto chaveRecordDto) {
36
37 Chave chave = new Chave();
38 BeanUtils.copyProperties(chaveRecordDto, chave);
39
40
41 return ResponseEntity.status(HttpStatus.CREATED).body(chaveRepository.save(chave))
42 }
43
44 @GetMapping
45
Task List
Find
All
Activate....
Outline
ChaveController
chaveRepository : ChaveRep
saveChave(ChaveRecordDto)
getAllChaves() : ResponseEr
getPerStauts(boolean) : Res
getPerSituacao(String) : Res
Problems
Javadoc
Declaration
Console
Boot Dashboard
Type tags, projects, or working set names to match (incl. * and ? wildcards)
local
backend
demo
Writable
Smart Insert
69 : 9 : 2229