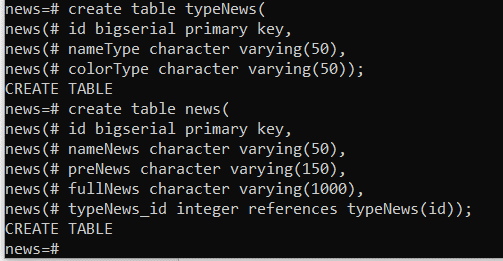
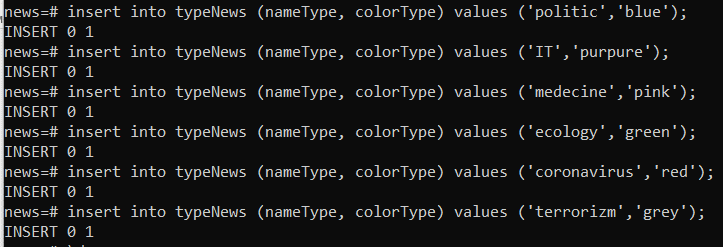
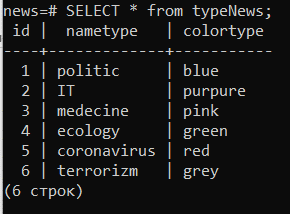
Перед запуском приложения необходимо создать 2 таблицы (таблица typeNews & таблица news) со следующими параметрами (можно их не заполнять)  


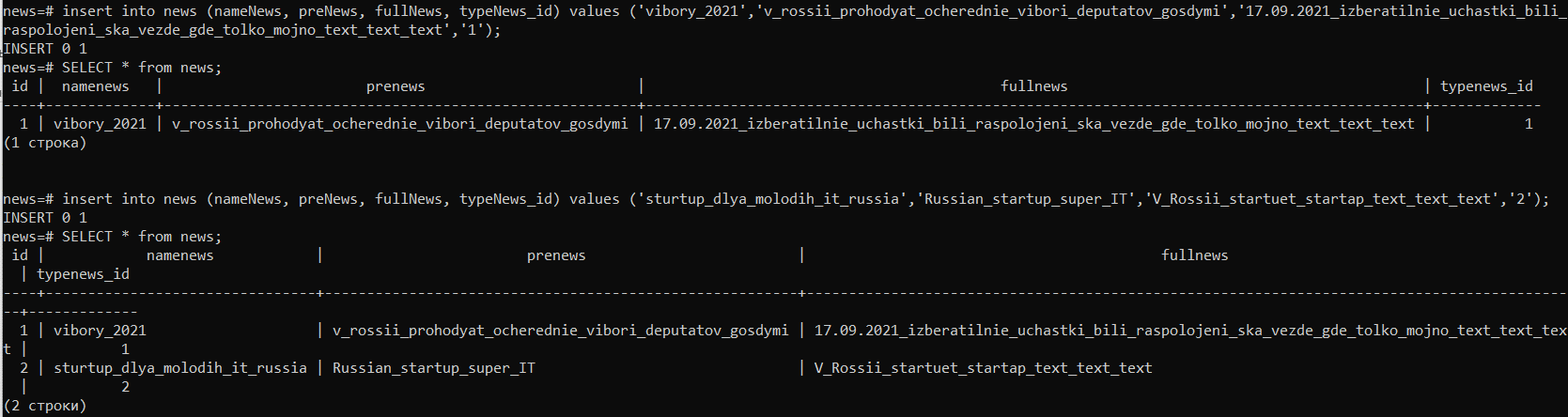
Пояснения:

Для создания таблицы был использован postgresql\_12

Чтобы видеть работоспособность, для наглядности я сразу заполнил таблицы (пока что хламом)







//@CrossOrigin(origins = "http://localhost:8080")  
@RestController  
public class NewsController {  
  
 @Autowired  
 private NewsService newsService;  
  
 @Autowired  
 private NewsRepo newsRepo;  
  
 @GetMapping("/news")  
 public List<News> getAllNews(@PathVariable String news) {  
 return newsRepo.findAll();  
 // return newsService.findAll();  
 }  
  
 @GetMapping("/news/{id}")  
 public News getNews(@PathVariable String news, @PathVariable long id) {  
 return newsRepo.findById(id).get();  
 // return newsService.findById(id);  
 }  
  
 @PutMapping("/news")  
 public ResponseEntity<News> updateNews(@RequestBody News news) {  
 // News updatedNews = newsService.save(news);  
 News updatedNews = newsRepo.save(news);  
 return new ResponseEntity<>(updatedNews, HttpStatus.*OK*);  
 }  
  
 @PostMapping("/news")  
 public ResponseEntity<Void> createNews(@RequestBody News news) {  
 // News createdNews = newsService.save(news);  
 News createdNews = newsRepo.save(news);  
  
 URI uri = ServletUriComponentsBuilder.*fromCurrentRequest*()  
 .path("/{id}").buildAndExpand(createdNews.getId()).toUri();  
  
 return ResponseEntity.*created*(uri).build();  
 }  
  
 @DeleteMapping("/news/{id}")  
 public ResponseEntity<Void> deleteNews(@PathVariable long id) {  
 // News news = newsService.deleteById(id);  
  
 newsRepo.deleteById(id);  
  
 return ResponseEntity.*noContent*().build();  
 }  
}