Сервер реализован на Spring boot 2, использует базу данных H2.

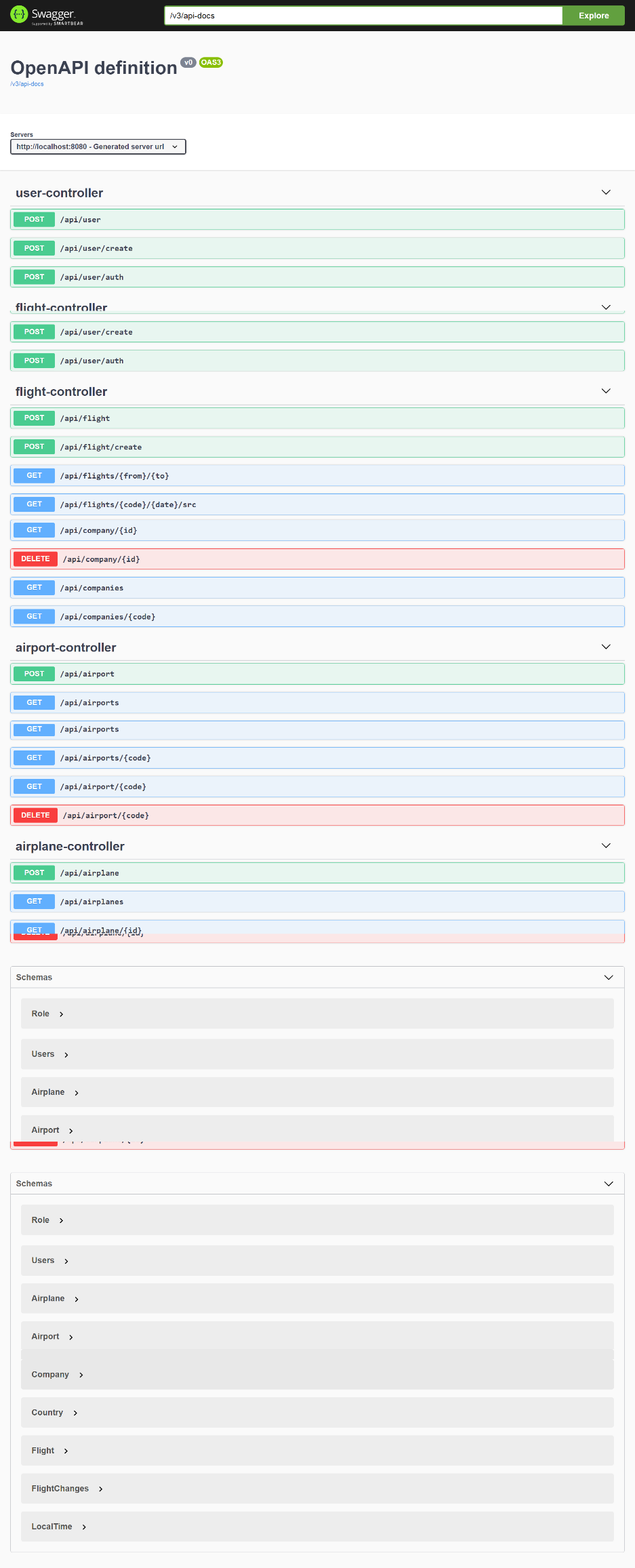
Доступ к базе данных после запуска сервера <http://localhost:8080/h2-console>

Для просмотра документации по ендпоинтам api нужно пройти авторизацию

<http://localhost:8080/login>

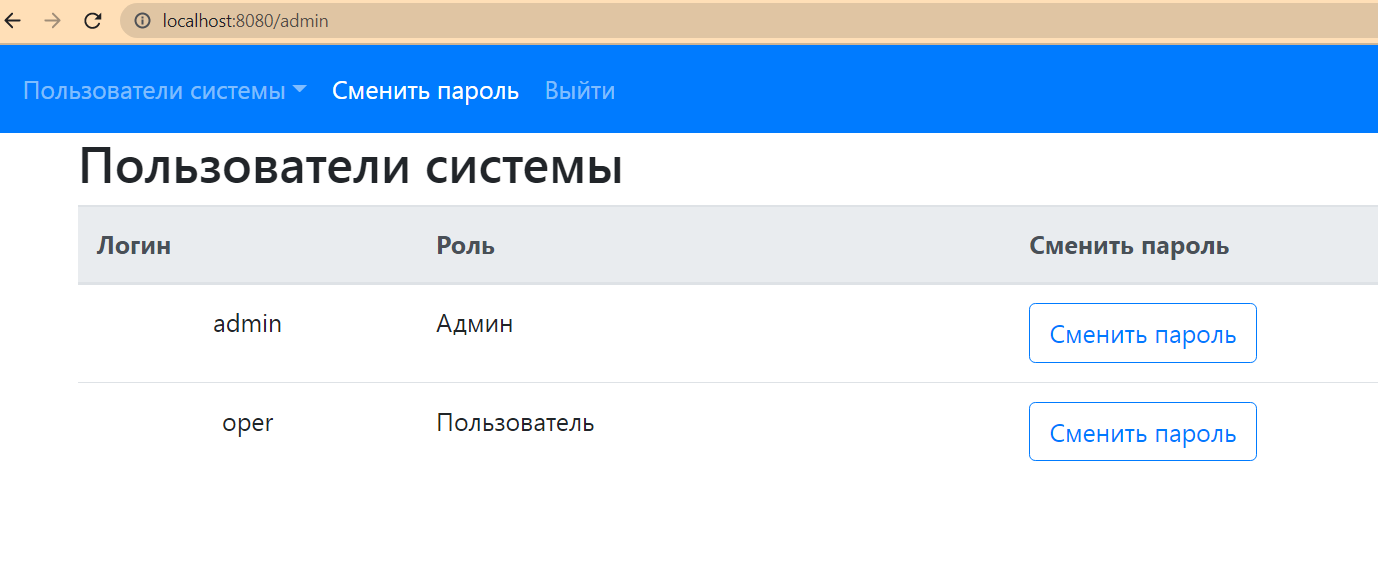
логин admin, пароль 123456

После чего будем перенаправлены на страницу документации



Так же после авторизации доступна форма с пользователями и возможностью сбросить пароль

<http://localhost:8080/admin>

Кл

По коду – rest контроллеры размещены в пакете com.app.api.controller

Они отвечают за реализацию rest api.

В пакете com.app.entity размещены сущности, которые проектируются на базу данных.

В пакете com.app.repos размещены репозитории, которые читают и пишут сущности в базу данных.

В пакете com.app.config размещены классы настройки, которые отвечают за авторизацию пользователей, разграничение доступа. При каждом запросе к api клиент должен передавать в заголовке логин/пароль в base64, что бы пройти называемую basic аутентификацию.

Невторизированый пользователь не сможет получтить информацию, а пользователь, у которого нет

прав на запись – тоже не сможет этого сделать.

В пакете com.app.mvc реализованые контроллеры админа api, который может заблокировать пользователя или сбросить ему пароль. Пароль хранится в базе в зашифрованном виде.

З проверку логина/пароль клиента отвечает rest контроллер UserController, он может

возрващает результат проверки логина/пароля пользователя.

@PostMapping(value = "/user/auth")  
ResponseEntity auth(@NotNull String login, @NotNull String password, HttpServletRequest request){  
 logger.info(String.*format*("call POST\\auth users %s %s", login,  
 request.getRemoteAddr()));  
 try {  
  
 Optional<Users> user = service.findUserByLogin(login);  
 if (user.isPresent() ){  
 // если пользователь заблокирован  
 if (!user.get().isActive()){  
 return ResponseEntity.*badRequest*().body("Пользователь заблокирован!");  
 }  
 if (passwordEncoder.matches(password, user.get().getPassword())){  
 return ResponseEntity.*ok*(user.get());  
 }  
 }  
  
 // если не угадали пароль  
 return ResponseEntity.*badRequest*().body("Не верный логин или пароль!");  
 }  
 catch (Exception e){  
 // если ошибка - выбросим 500 код и ошибку  
 logger.error(String.*format*("call POST\\flight from %s ,%s ",request.getRemoteAddr(),e.getMessage()));  
 // вернем сообщение об ошибке  
 if (e.getCause()!=null)  
 if (e.getCause().getCause()!=null)  
 {  
 String[] message = e.getCause().getCause().getMessage().split(":");  
 String msg=e.getCause().getCause().getMessage();  
 if (message.length>0) msg = message[message.length-1];  
 return ResponseEntity.*badRequest*().body(e.getMessage());  
  
 }  
 return ResponseEntity.*badRequest*().body(e.getCause().getMessage());  
 }  
  
}