

Тестовое задание "Тестирование API"				
Задание 1.				
Напишите позитивные и негативные тест-кейсы на проверку ручки регистрации api/v1/register				
№	Проверки	Тестовые данные	Шаги	Ожидаемый результат
1	Позитивные проверки с валидными данными			
1.1	Регистрация нового пользователя в системе с валидными данными поля "Логин"	Request Body: { "login": "login", "email": "email@email.com", "password": "password" "confirmPassword": "password" }, где login - валидное значение для поля "Логин" (См. лист "Данные")	1. Регистрация нового пользователя в системе: Отправить запрос POST api/v1/register с телом {login: String; email: String; password: String; confirmPassword: String}	Запрос успешно отправлен на сервер, новый пользователь зарегистрирован Status code: 200 OK/ 201 Created
			2. Проверка статуса кода	{ code: int; result: String}
			3. Проверить структуру ответа	Ответ получен в разумные ожидаемые сроки
		Request: api/v1/getUser/login	4. Проверка производительности	
			5. Проверка наличия пользователя в системе: Отправить запрос GET api/v1/getUser/{login}, где login - логин регистрируемого пользователя	Пользователь есть в системе Status code: 200 OK Response: {code: int; result: String}
1.2	Регистрация нового пользователя в системе с валидными данными поля "Email"	Request Body: { "login": "login", "email": "email", "password": "password" "confirmPassword": "password" }, где email - валидное значение для поля "Email" (См. лист "Данные")	1. Регистрация нового пользователя в системе: Отправить запрос POST api/v1/register с телом {login: String; email: String; password: String; confirmPassword: String}	Запрос успешно отправлен на сервер, новый пользователь зарегистрирован Status code: 200 OK/ 201 Created
			2. Проверка статуса кода	{ code: int; result: String}
			3. Проверить структуру ответа	Ответ получен в разумные ожидаемые сроки
		Request: api/v1/getUser/login	4. Проверка производительности	
			5. Проверка наличия пользователя в системе: Отправить запрос GET api/v1/getUser/{login}, где login - логин регистрируемого пользователя	Пользователь есть в системе Status code: 200 OK Response: {code: int; result: String}
2	Негативные проверки с валидными данными			
2.1	Регистрация нового пользователя в системе с уже существующим логином	Request Body: { "login": "login_1", "email": "email@email.com", "password": "password" "confirmPassword": "password" }	1. Регистрация нового пользователя в системе: Отправить запрос POST api/v1/register с телом {login: String; email: String; password: String; confirmPassword: String}	Запрос успешно отправлен на сервер, новый пользователь зарегистрирован Status code: 200 OK/ 201 Created Response: {code: int; result: String}
		Request: api/v1/getUser/login_1	2. Проверка наличия пользователя в системе: Отправить запрос GET api/v1/getUser/{login}, где login - логин регистрируемого пользователя	Пользователь есть в системе Status code: 200 OK Response: {code: int; result: String}
		Request Body: { "login": "login_1", "email": "new@email.com", "password": "password_new" "confirmPassword": "password_new" }	3. Регистрация нового пользователя в системе с логином из шага 1 Отправить запрос POST api/v1/register с телом {login: String; email: String; password: String; confirmPassword: String}	Невозможно зарегистрировать пользователя с уже существующим логином, сообщение об ошибке Response: {code: int; result: String}
2.2	Регистрация нового пользователя в системе с уже существующим email	Request Body: { "login": "login", "email": "mail@email.com", "password": "password" "confirmPassword": "password" }	1. Регистрация нового пользователя в системе: Отправить запрос POST api/v1/register с телом {login: String; email: String; password: String; confirmPassword: String}	Запрос успешно отправлен на сервер, новый пользователь зарегистрирован Status code: 200 OK/ 201 Created Response: {code: int; result: String}
		Request: api/v1/getUser/login	2. Проверка наличия пользователя в системе: Отправить запрос GET api/v1/getUser/{login}, где login - логин регистрируемого пользователя	Пользователь есть в системе Status code: 200 OK Response: {code: int; result: String}
		Request Body: { "login": "login_new", "email": "mail@email.com", "password": "password_new" "confirmPassword": "password_new" }	3. Регистрация нового пользователя в системе с email из шага 1 Отправить запрос POST api/v1/register с телом {login: String; email: String; password: String; confirmPassword: String}	Невозможно зарегистрировать пользователя с уже существующим email, сообщение об ошибке Response: {code: int; result: String}
2.3	Регистрация нового пользователя в системе с одинаковыми значениями пароля и логина	Request Body: { "login": "login", "email": "mail@email.com", "password": "login" "confirmPassword": "login" }	1. Регистрация нового пользователя в системе: Отправить запрос POST api/v1/register с телом {login: String; email: String; password: String; confirmPassword: String}	Оповещение о небезопасном пароле.
3	Негативные проверки с невалидными данными			
3.1	Регистрация нового пользователя с невалидным логином	Request Body: { "login": "login", "email": "email@email.com", "password": "password" "confirmPassword": "password" }, где login - невалидное значение для поля "Логин" (См. лист "Данные")	1. Регистрация нового пользователя в системе: Отправить запрос POST api/v1/register с телом {login: String; email: String; password: String; confirmPassword: String}	Регистрация невозможна, сообщение об ошибке Response: {code: int; result: String}
3.2	Регистрация нового пользователя с невалидным email	Request Body: { "login": "login", "email": "email", "password": "password" "confirmPassword": "password" }, где email - невалидное значение для поля "Email" (См. лист "Данные")	1. Регистрация нового пользователя в системе: Отправить запрос POST api/v1/register с телом {login: String; email: String; password: String; confirmPassword: String}	Регистрация невозможна, сообщение об ошибке Response: {code: int; result: String}
3.3	Регистрация нового пользователя с неверным подтверждением пароля	Request Body: { "login": "login", "email": "email", "password": "password" "confirmPassword": "new_password" }	1. Регистрация нового пользователя в системе: Отправить запрос POST api/v1/register с телом {login: String; email: String; password: String; confirmPassword: String}	Регистрация невозможна, сообщение об ошибке Response: {code: int; result: String}
3.4	Регистрация нового пользователя с пустыми значениями в обязательных полях	Request Body: { "login": "", "email": "", "password": "", "confirmPassword": "" }	1. Регистрация нового пользователя в системе: Отправить запрос POST api/v1/register с телом с пустыми значениями в обязательных строках	Регистрация невозможна, сообщение об ошибке Response: {code: int; result: String}
3.5	Регистрация нового пользователя без обязательных полей	Request Body: { "login": "login", "password": "password" } }	1. Регистрация нового пользователя в системе: Отправить запрос POST api/v1/register с телом без строк "email" и "confirmPassword"	Регистрация невозможна, сообщение об ошибке Status code: 400 Bad Request
3.6	Регистрация нового пользователя с пустым телом запроса	Request Body: { }	1. Регистрация нового пользователя в системе: Отправить запрос POST api/v1/register с пустым телом запроса	Регистрация невозможна, сообщение об ошибке Status code: 400 Bad Request
Дополнительный вопрос:				
Как бы вы проверили работу данного функционала на сайте?				

Помимо проведения функционального тестирования на валидацию данных в поля ввода или тех или иных сценариев регистрации, при тестировании на сайте можно провести ШУДХ – тестирование, которое учитывать удобства формы регистрации для клиентов. Различные поля ввода, их расположение, а также удобство их заполнения; появляются ли подсказки по поводу заполнения поля, есть ли маски; наличие чек-боксов «Запомнить меня», опции «скрыть/показать пароль». Саму форму регистрации должно быть легко найти на сайте. Обязательные поля должны быть выделены с помощью звездочки, пока все они не заполнены, кнопка отправки не должна быть активной. После отправки валидных данных пользователю нужно показать, что регистрация прошла успешно.