

тестирование API через POSTMAN https://developer.atlassian.com/cloud/jira/platform/rest/v3/api-group-issues/ метод Create issue	Ожидаемый результат/статус
0 Проверка позитивного запроса из документации 0.1 POST Создание задачи 0.2 GET Проверка созданной задачи	Created - 201 ok - 200
1. Проверка заголовка URL 1.1. Поменять запрос https через http 1.1.1. для POST 1.1.2. для GET 1.2. Ключ задачи / id объекта 1.2.1. Указан существующий ключ задачи (id объекта) 1.2.2. Не указан ключ задачи (id объекта) 1.2.3. Указан несуществующий ключ задачи (id объекта) 1.2.4. Указан ключ задачи (id объекта), к которым нет доступа 1.2.5. Указан ключ задачи (id объекта), который был удалён 1.2.5.1. DELETE Удалить задачу с текущим ключом 1.2.5.2. GET Проверка удалённой задачи 1.2.5.3. POST Создание новой задачи взамен удалённой 1.2.6. Проверка регистронезависимости ключа 1.2.7. Передать параметры и значения в url 1.2.8. Проверка длины запроса, так как могут быть ограничения на сервере 1.2.8.1. Apache: по умолчанию 8 килобайт (8192 символов) 1.2.8.2. Microsoft Internet Information Server: ограничена пределом в 16 килобайт (16384 символов)	Method Not Allowed - 405 ok - 200 ok - 200 Method Not Allowed - 405 not found - 404 not found - 404 No Content - 204 not found - 404 Created - 201 ok - 200 Unsupported Media Type - 415 Created - 201 Created - 201
2. Проверка типа запроса 2.1. А если изменить POST на GET? 2.2. А если изменить POST на PUT?	Method Not Allowed - 405 Method Not Allowed - 405
3. Проверка авторизации 3.1. Basic-авторизация пройдена успешно 3.2. Запросы без авторизации 3.2.1. POST-запрос на создание задачи без авторизации 3.2.2. GET-запрос проверка созданной задачи без авторизации 3.3. Неверные параметры авторизации 3.3.1. не ввести пароль 3.3.2. ввести неправильный пароль 3.3.3. не ввести e-mail	Created - 201 Bad Request - 400 not found - 404 Unauthorized - 401 Unauthorized - 401 Bad Request - 400

3.3.4. ввести неправильный e-mail	Bad Request – 400
3.4. Проверка регистронезависимости логина в basic-авторизации	Created – 201
4. Проверка заголовка «Асепт»	
4.1. Не передаём заголовок	Created – 201
4.2. Передаём заголовок «Асепт: application/json»	Created – 201
4.3. Передаём заголовок с типом не json	Not Acceptable - 406
4.4. Проверка регистронезависимости заголовка Accept aPpLiCAtIoN/JsOn	Created – 201
5. Проверка тела (запроса)	
5.1. Правильное тело – пример из документации	Created – 201
5.2. ошибка не well formed xml/Json	
5.2.1. убрать запятую между параметрами	Bad Request – 400
5.2.2. лишняя скобка	Bad Request – 400
5.2.3. параметр без кавычек	Bad Request – 400
5.2.4. без фигурных скобок	Bad Request – 400
5.2.5. без закрывающей скобки	Bad Request – 400
5.2.6. закрывающая скобка от массива	Bad Request – 400
5.2.7. запятая перед закрывающей скобкой	Bad Request – 400
5.2.8. массив в фигурных скобках	Bad Request – 400
5.3. перестановка мест слагаемых	Created – 201
5.4. Регистрозависимость	Bad Request – 400
5.5. Различные параметры	
5.5.1. обязательность	
5.5.1.1. отсутствие значений параметров	Bad Request – 400
5.5.1.2. свои несуществующие параметры	Bad Request – 400
5.5.2. дублирование параметров (Одно имя с разными значениями)	Created – 201
5.6. Проверка Бизнес-логики (позитив, негатив)	Created – 201
5.6.1. Проверяем поле summaгу в рамках бизнес логики	или
5.6.1.1. Размер/длина (в рамках тестирования полей)	Bad Request – 400
5.6.1.1.1. Минимум – одна буква	
5.6.1.1.2. Ноль – пустое поле (Меньше минимального)	
5.6.1.1.3. Максимум – 255 символов	
5.6.1.1.4. Больше максимального – 256 символов	
5.6.1.2. цифры	
5.6.1.2.1. Ноль – 0	
5.6.1.3. символы	
5.6.1.3.1. специальные символы (!@»#№\$;%^:~&*()_{}[]“:~?><.,\ /)	
5.6.1.3.2. Не ASCII –	
5.6.1.3.3. Символы конца строки (\n, \r)	

5.6.1.4.	текст с пробелом	
5.6.1.4.1.	В начале строки (текст)	
5.6.2.	Проверяем поле <code>issuetype / id</code> в рамках бизнес-логики	
5.6.3.	Проверяем поле <code>project / id</code> в рамках бизнес-логики	
5.7.	Проверки аспектов безопасности:	
5.7.1.	SQL-инъекции <code>' OR '1'='1'</code>	
5.7.2.	XSS-инъекции <code><script>alert("xss!")</script></code>	
5.7.3.	HTML-инъекции <code></body></body></code>	
5.7.3.1	<code><form action="http://live.hh.ru"><input type="submit"></form></code>	
6.	Проверка получаемого ответа	
	Какие поля вернулись в ответе	
	Значения в полях	
	Текст ошибок	
	Статус код	
	Поле ответа	
	сравнить с ТЗ	
	сравнить SOAP с RESTом	