

Выполнение тестового задания

В первую очередь определим, что является объектом тестирования, для чего оно нужно и как используется? Объектом тестирования является метод Edit an issue, предназначенный для редактирования issue (проблемы). Выполнять редактирование могут пользователи с соответствующими правами доступа.

Весь доступ к API осуществляется по протоколу HTTPS: <https://api.github.com>. При этом аутентификацию можно производить через token, отправляя его в заголовке запроса. Все данные отправляются и принимаются в формате JSON.

Для выполнения редактирования необходимо выполнить запрос вида:

PATCH /repos/:owner/:repo/issues/:number

Метод Edit an issue имеет следующий набор параметров:

Имя	Тип	Описание
title	string	Заголовок проблемы.
body	string	Описание проблемы.
assignee	string	Логин пользователя, которому назначена эта проблема. <u>Этот параметр устарел.</u>
state	string	Состояние проблемы: open (открыто) или closed (закрыто).
milestone	integer	Номер контрольной точки, с которой связана данная проблема, или null для удаления текущих связей. <i>ПРИМЕЧАНИЕ. Только пользователи, имеющие доступ, могут устанавливать контрольную точку. В противном случае этот параметр игнорируется.</i>
labels	array of strings	Метки, связанные с этой проблемой. Возможно передать одну или несколько меток, чтобы заменить текущий набор меток, или пустой массив ([]), чтобы удалить все метки. <i>ПРИМЕЧАНИЕ. Только пользователи, имеющие доступ, могут устанавливать метки. В противном случае этот параметр игнорируется.</i>
assignees	array of strings	Логин пользователей, которым назначена эта проблема. Возможно передать один или несколько логинов, чтобы заменить текущий набор назначенных лиц, или пустой массив ([]), чтобы очистить текущий набор назначенных лиц. <i>ПРИМЕЧАНИЕ. Только пользователи, имеющие доступ, могут назначать пользователей для решения проблем. В противном случае этот параметр игнорируется.</i>

Согласно этой таблице и официальной документации [github](https://help.github.com/articles/repository-permission-levels-for-an-organization/) (<https://help.github.com/articles/repository-permission-levels-for-an-organization/>) имеем следующую систему доступа к редактированию проблемы.

Создать issue может любой пользователь. Отредактировать issue может либо ее автор, либо владелец репозитория. Причем отдельные параметры (milestone, labels и assignees) изменяются только владельцем репозитория (если попытаться изменить их от имени автора issue, то данные параметры не изменятся, но само редактирование будет осуществлено успешно). Закрыть и переоткрыть issue может либо ее автор, либо владелец репозитория.

Стоит отметить, что в начале необходимо выполнять позитивное тестирование, а затем – негативное.

Рассмотрим каждый из параметров отдельно.

1. Параметр *title*

Воспользуемся разбиением на классы эквивалентности. Данный параметр имеет единственный тип – string. Таким образом, можно выделить два класса: string и любые другие типы (int, float, array и др.). Ограничение на максимальную длину строки и валидные символы не известны, поэтому будем руководствоваться здравым смыслом (в идеале – уточнить требования). Поле «заголовок» предполагает краткое, емкое описание проблемы. Длинные строки (тысячи символов) вряд ли будут использоваться в качестве заголовка. Рассмотрим заголовок обычной длины (порядка 50 символов) и большой длины (1000 символов). Минимальная возможная длина заголовка – 1 символ, так как согласно документации метода Create an issue (<https://developer.github.com/v3/issues/#create-an-issue>) заголовок не может быть пустым (0 символов). В соответствии с тестированием граничных значений необходимо протестировать значения слева и справа от границы, т.е. 0 и 2 символа. Будем считать, что заголовок может состоять из букв нижнего и верхнего регистра русского и английского алфавита, цифр (0-9) и спец. символов.

Таким образом, имеем:

- 1) 50 символов (рус. алфавит, смешанный регистр)
- 2) 1 символ (рус. алфавит)
- 3) 2 символа (рус. алфавит, верхний регистр)
- 4) 1000 символов (англ. алфавит и цифры, смешанный регистр)
- 5) "@#\$\$%^&.;?,>|\N"!"_{}[]<~" (спец. символы)
- 6) 0 символов (пустая строка)
- 7) int (неверный тип)

2. Параметр *body*

Выделим классы эквивалентности: string и любые другие типы (int, float, array и др.). Поле «описание» предполагает развернутое, подробное разъяснение проблемы. Поэтому могут использоваться строки большой длины (тысячи символов). Рассмотрим описание обычной длины (порядка 50 символов) и большой длины (10000 символов). Минимальная возможная длина строки – 0, так как описание может быть пустым. В соответствии с тестированием граничных значений необходимо протестировать -1 и 1 символ. Длина строки не может быть отрицательным числом, а значение 1, на мой взгляд, в данном случае является не критичным, так как уже рассмотрено значение 50. Будем считать, что описание может состоять из букв нижнего и верхнего регистра русского и английского алфавита, цифр (0-9) и спец. символов.

Таким образом, имеем:

- 8) 50 символов (рус. алфавит, смешанный регистр)
- 9) 10000 символов (англ. алфавит и цифры, смешанный регистр)
- 10) 0 символов (пустая строка)

- 11) "@#\$\$%^&.:?,>|\\№"!()_{}[<~" (спец. символы)
- 12) array (неверный тип)

3. Параметр *assignee*

Не будет рассматриваться, так как является устаревшим.

4. Параметр *state*

Выделим классы эквивалентности: string и любые другие типы (int, float, array и др.). Существует два возможных состояния: open и closed. Кроме того следует учесть, что закрыть и заново открыть issue может либо ее автор, либо владелец репозитория.

Получаем:

от имени владельца репозитория:

- 13) "closed" (закрыть)
- 14) "open" (открыть)
- 15) "cLoSEd" (смешанный регистр)
- 16) "OPEN" (верхний регистр)
- 17) "new state" (несуществующее состояние)
- 18) float (неверный тип)

от имени гостя:

- 19) закрытие проблемы от имени гостя, который не является ее автором
- 20) закрытие проблемы от имени гостя, который является ее автором (не будет автоматизироваться, так как сценарий данного кейса является более сложным по сравнению с другими, отложим его на ближайшее будущее)
- 21) переоткрытие проблемы от имени гостя, который не является ее автором
- 22) переоткрытие проблемы от имени гостя, который является ее автором (не будет автоматизироваться, так как не является критичным: при неудачном переоткрытии остается возможность создания новой issue)

5. Параметр *milestone*

Представляет собой целочисленный положительный номер существующей контрольной точки или null для удаления текущих связей. Следует помнить, что данный параметр успешно изменяется только владельцем репозитория.

от имени владельца репозитория:

- 23) существующий номер контрольной точки
- 24) null (удаление связей)
- 25) 0 (несуществующий номер, следует рассмотреть отдельно)
- 26) -1 (несуществующий отрицательный номер)
- 27) очень большое целое число (несуществующий номер, поддерживает ли его тип int)
- 28) string (неверный тип)

от имени гостя:

- 29) существующий номер контрольной точки

6. Параметр *labels*

Представляет собой список меток в формате string или пустой список для удаления существующих меток. Следует помнить, что данный параметр успешно изменяется только владельцем репозитория.

от имени владельца репозитория:

- 30) список из 1 метки в формате string (англ. алфавит, нижний регистр и цифры)
- 31) список из 2 меток в формате string (рус. и англ. алфавит, смешанный регистр)
- 32) пустой список (удаление меток)
- 33) список из 1 метки в формате string (спец. символы)
- 34) список из меток в формате string, int и float (неверный тип)

от имени гостя:

- 35) список из 1 метки в формате string (англ. алфавит, нижний регистр)

7. Параметр *assignees*

Представляет собой список логинов существующих пользователей в формате string или пустой список. Следует помнить, что данный параметр успешно изменяется только владельцем репозитория.

- 36) список из 1 существующего логина в формате string
- 37) список из 2 существующих логинов в формате string
- 38) пустой список (удаление назначенных пользователей)
- 39) список из 1 несуществующего логина в формате string
- 40) список из логинов в формате string, int и float (неверный тип)

от имени гостя:

- 41) список из 1 существующего логина в формате string

Кроме того следует протестировать редактирование issue от имени

- 42) владельца репозитория (изменяются все параметры)
- 43) ее автора, не являющегося владельцем репозитория (параметры milestone, labels и assignees остаются неизменными)

и рассмотреть ситуацию, когда запрос имеет неверный url:

- 44) PATCH /repos/:owner/:repo/issues////