**1 Тест кейс**

**Идея:**

Получение списка пользователей из базы данных (БД)

**Приоритет:**

High

**Описание:**

Получение списка пользователей из БД с помощью Postman

**Предусловие:**

* Публичный сервис по адресу 129.146.247.102 работоспособен
* Открыт внешний доступ к порту 5000

**Шаги:**

|  |  |  |
| --- | --- | --- |
| № | Описание | Ожидаемый результат |
| 1 | В Postman создать новый запрос типа GET | Создан новый GET запрос |
| 2 | Задать URL «129.146.247.102:5000/users» и отправить запрос | Статус код 200 ОК, получен список пользователей из БД в виде json содержащий:  - "username" – имя пользователя  - "dateOfBirth" – дата рождения |

**Ожидаемый результат:**

Статус код 200 ОК, получен список пользователей из БД

**2 Тест кейс**

**Идея:**

Сохранение нового пользователя и даты его рождения в БД

**Приоритет:**

High

**Описание:**

Добавление пользователя и его даты рождения в БД с помощью Postman

**Предусловие:**

* Публичный сервис по адресу 129.146.247.102 работоспособен
* Открыт внешний доступ к порту 5000

**Шаги:**

|  |  |  |
| --- | --- | --- |
| № | Описание | Ожидаемый результат |
| 1 | В Postman создать новый запрос типа POST | Создан новый POST запрос |
| 2 | Прописать в тело json дату рождения вида <{"dateOfBirth": "YYYY-MM-DD"}>, где: YYYY-MM-DD – дата, до текущей даты | Json содержит дату рождения в верном формате |
| 3 | Задать URL «129.146.247.102:5000/hello/<имя\_пользователя>» и отправить запрос, где: <имя\_пользователя> - желаемое имя пользователя содержащее только буквы, например “anatoliy”, “самойлов” и т.п. | Статус код 201, получено ответное сообщение в виде json содержащее:  - "message": "New user '<имя\_пользователя>' was added successfully." |
| 4 | Выполнить GET запрос по URL «129.146.247.102:5000/users» и найти пользователя <имя\_пользователя> заданного в шаге 3 | В БД имеется пользователь <имя\_пользователя> |

**Ожидаемый результат:**

В БД создана новая запись содержащая имя пользователя и дату его рождения

**3 Тест кейс**

**Идея:**

Добавление уже существующего пользователя

**Приоритет:**

Medium

**Описание:**

Добавление пользователя с именем, уже существующем в БД, с помощью Postman

**Предусловие:**

* Публичный сервис по адресу 129.146.247.102 работоспособен
* Открыт внешний доступ к порту 5000

**Шаги:**

|  |  |  |
| --- | --- | --- |
| № | Описание | Ожидаемый результат |
| 1 | В Postman создать новый запрос типа POST | Создан новый POST запрос |
| 2 | Прописать в тело json дату рождения вида <{"dateOfBirth": "YYYY-MM-DD"}>, где: YYYY-MM-DD – дата, не превышающая текущую | Json содержит дату рождения в верном формате |
| 3 | Задать URL «129.146.247.102:5000/hello/<имя\_пользователя>» и отправить запрос | Статус код 201, получено ответное сообщение в виде json содержащее:  - "message": "New user '<имя\_пользователя>' was added successfully." |
| 4 | Повторить шаг 3 с тем же <имя\_пользователя> | Статус код 409, получено ответное сообщение в виде json содержащее:  - "error": "User already exists. Use 'PUT' method for updating the date of birth." |
| 5 | Повторить шаг 2, 3 изменив дату рождения сохраняя корректный формат YYYY-MM-DD | Статус код 409, получено ответное сообщение в виде json содержащее:  - "error": "User already exists. Use 'PUT' method for updating the date of birth." |
| 6 | Выполнить GET запрос по URL «129.146.247.102:5000/users» и найти пользователя <имя\_пользователя> заданного в шаге 3, проверить изменения | В БД имеется один пользователь <имя\_пользователя>, изменений нет |

**Ожидаемый результат:**

Пользователь не добавлен, получено сообщение о том, что пользователь уже существует

**4 Тест кейс**

**Идея:**

Добавление пользователя с некорректным форматом имени

**Приоритет:**

Medium

**Описание:**

Добавление пользователя с именем, состоящим не только из буквенных символов, с помощью Postman

**Предусловие:**

* Публичный сервис по адресу 129.146.247.102 работоспособен
* Открыт внешний доступ к порту 5000

**Шаги:**

|  |  |  |
| --- | --- | --- |
| № | Описание | Ожидаемый результат |
| 1 | В Postman создать новый запрос типа POST | Создан новый POST запрос |
| 2 | Прописать в тело json дату рождения вида <{"dateOfBirth": "YYYY-MM-DD"}>, где: YYYY-MM-DD – дата, не превышающая текущую | Json содержит дату рождения в верном формате |
| 3 | Задать URL «129.146.247.102:5000/hello/<имя\_пользователя>» и отправить запрос, где: <имя\_пользователя> - формат имени пользователя отличный от буквенного, например числовой “12345”, “72382” и т.п. | Статус код 400, получено ответное сообщение в виде json содержащее:  - "error": "Username must be a string containing only letters." |
| 4 | Повторить шаг 3, но с другим форматом имени пользователя, например числовым с буквенным “and123”, “1q2w3e” и т.п. | Статус код 400, получено ответное сообщение в виде json содержащее:  - "error": "Username must be a string containing only letters |
| 5 | Повторить шаг 3, но с другим форматом имени пользователя, например буквенный с символами “and\_”, “-=king=-” и т.п. | Статус код 400, получено ответное сообщение в виде json содержащее:  - "error": "Username must be a string containing only letters |
| 6 | Повторить шаг 3, но с другим форматом имени пользователя, например только символы “\_-=”, “№;:%?” и т.п. | Статус код 400, получено ответное сообщение в виде json содержащее:  - "error": "Username must be a string containing only letters |
| 7 | Выполнить GET запрос по URL «129.146.247.102:5000/users» и попытаться найти пользователя <имя\_пользователя> заданного в шагах 3 - 6, проверить изменения | В БД нет пользователя <имя\_пользователя> |

**Ожидаемый результат:**

Пользователь не добавлен, получено сообщение о том, что имя пользователя должно содержать только буквы

**5 Тест кейс**

**Идея:**

Добавление пользователя с некорректным форматом даты рождения

**Приоритет:**

Medium

**Описание:**

При некорректном формате даты рождения не должна создаваться новая запись в БД.

**Предусловие:**

* Публичный сервис по адресу 129.146.247.102 работоспособен
* Открыт внешний доступ к порту 5000

**Шаги:**

|  |  |  |
| --- | --- | --- |
| № | Описание | Ожидаемый результат |
| 1 | В Postman создать новый запрос типа POST | Создан новый POST запрос |
| 2 | Прописать в тело json дату рождения вида <{"dateOfBirth": "<дата>"}>, где: <дата> – не соответствует формату YYYY-MM-DD, например “1 января 1995”, “2 августа 1983 г.” | Json содержит дату рождения в неверном формате |
| 3 | Задать URL «129.146.247.102:5000/hello/<имя\_пользователя>» и отправить запрос, где: <имя\_пользователя> - желаемое имя пользователя содержащее только буквы, например “alarionov”, “alexander” и т.п. | Статус код 400, получено ответное сообщение в виде json содержащее:  "error": "JSON field 'dateOfBirth' is missing or value not in 'YYYY-MM-DD' format." |
| 4 | Повторить шаг 2, но <дата> является несуществующей датой, например, “2020-06-52”; и отправить POST запрос согласно шагу 3 | Статус код 400, получено ответное сообщение в виде json содержащее:  "error": "JSON field 'dateOfBirth' is missing or value not in 'YYYY-MM-DD' format." |
| 5 | Повторить шаг 2, но <дата> не является датой, например, “123456”, “йцукен”; и отправить POST запрос согласно шагу 3 | Статус код 400, получено ответное сообщение в виде json содержащее:  "error": "JSON field 'dateOfBirth' is missing or value not in 'YYYY-MM-DD' format." |
| 6 | Повторить шаг 2, но <дата> формата YYYY-MM-DD, является високосной датой, тогда как год не високосный, например, “2021-02-29”, “2019-02-29”; и отправить POST запрос согласно шагу 3 | Статус код 400, получено ответное сообщение в виде json содержащее:  "error": "JSON field 'dateOfBirth' is missing or value not in 'YYYY-MM-DD' format." |
| 7 | Повторить шаг 2, но <дата> формата YYYY-MM-DD, старше текущей даты, например “2035-01-01”, “2045-12-30” и т.п.; и отправить POST запрос согласно шагу 3 | Статус код 422, получено ответное сообщение в виде json содержащее:  "error": " Date of birth must be a date before the today date." |
| 8 | Выполнить GET запрос по URL «129.146.247.102:5000/users» и попытаться найти пользователя <имя\_пользователя> заданного в шаге 3 | В БД нет пользователя <имя\_пользователя> |

**Ожидаемый результат:**

Пользователь не добавлен, при добавлении получено сообщение о неверном формате даты рождения

**6 Тест кейс**

**Идея:**

Обновление даты рождения пользователя в БД

**Приоритет:**

Medium

**Описание:**

По заданному имени пользователя должна быть обновлена его дата рождения

**Предусловие:**

* Публичный сервис по адресу 129.146.247.102 работоспособен
* Открыт внешний доступ к порту 5000

**Шаги:**

|  |  |  |
| --- | --- | --- |
| № | Описание | Ожидаемый результат |
| 1 | Создать нового пользователя через POST запрос по URL «129.146.247.102:5000/hello/<имя\_пользователя>», где: <имя\_пользователя> - желаемое имя пользователя содержащее только буквы, например “anatoliy”, “самойлов” и т.п. | Создан пользователь <имя\_пользователя> |
| 2 | В Postman создать запрос PUT | Создан PUT запрос |
| 3 | Изменить тело json на дату рождения вида <{"dateOfBirth": "YYYY-MM-DD"}>, где: YYYY-MM-DD – дата, не превышающая текущую и отличная от указанной в шаге 1 | Json содержит дату рождения в верном формате |
| 4 | Задать URL «129.146.247.102:5000/hello/<имя\_пользователя>» и отправить запрос, где: <имя\_пользователя> - имя пользователя указанное в шаге 1 | Статус код 204 No content, дата рождения пользователя <имя\_пользователя> обновлена |
| 5 | В Postman поменять тип запроса на GET и задать URL «129.146.247.102:5000/users» и отправить запрос | Дата рождения <имя\_пользователя> обновлена согласно шагу 3 |

**Ожидаемый результат:**

Дата рождения заданного пользователя изменена

**7 Тест кейс**

**Идея:**

Обновление данных по имени несуществующего пользователя

**Приоритет:**

Low

**Описание:**

Попытка обновить дату рождения пользователя, которого еще нет в БД

**Предусловие:**

* Публичный сервис по адресу 129.146.247.102 работоспособен
* Открыт внешний доступ к порту 5000

**Шаги:**

|  |  |  |
| --- | --- | --- |
| № | Описание | Ожидаемый результат |
| 1 | В Postman создать новый запрос типа PUT | Создан новый PUT запрос |
| 2 | Прописать в тело json дату рождения вида <{"dateOfBirth": "YYYY-MM-DD"}>, где: YYYY-MM-DD –дата заданного формата, не старше текущей даты | Json содержит дату рождения в верном формате |
| 3 | Задать URL «129.146.247.102:5000/hello/<имя\_пользователя>» и отправить запрос, где: <имя\_пользователя> - имя не существующего пользователя содержащее только буквы, например “anatoliy”, “самойлов” и т.п. | Статус код 201, получено ответное сообщение в виде json содержащее:  - "message": "New user '<имя\_пользователя> ' was added successfully." |
| 4 | В Postman поменять тип запроса на GET и очистить тело json | Задан GET тип запроса, json пуст |
| 5 | Задать URL «129.146.247.102:5000/users» и отправить запрос | Создан пользователь с <имя\_пользователя> и датой рождения согласно шагам 2, 3 |

**Ожидаемый результат:**

Создан новый пользователь, получено сообщение «New user '<имя\_пользователя> ' was added successfully.»

**8 Тест кейс**

**Идея:**

Обновление данных не существующего пользователя при некорректном формате даты рождения

**Приоритет:**

Medium

**Описание:**

Создание/обновление пользователя при некорректном формате даты рождения

**Предусловие:**

* Публичный сервис по адресу 129.146.247.102 работоспособен
* Открыт внешний доступ к порту 5000

**Шаги:**

|  |  |  |
| --- | --- | --- |
| № | Описание | Ожидаемый результат |
| 1 | В Postman создать новый запрос типа PUT | Создан новый PUT запрос |
| 2 | Прописать в тело json дату рождения вида <{"dateOfBirth": "<дата>"}>, где: <дата> – не соответствует формату даты YYYY-MM-DD, например “1 января 1995”, “2 августа 1983 г.” | Json содержит дату рождения в неверном формате |
| 3 | Задать URL «129.146.247.102:5000/hello/<имя\_пользователя>» и отправить запрос, где: <имя\_пользователя> - желаемое имя пользователя содержащее только буквы, например “alarionov”, “alexander” и т.п. | Статус код 400, получено ответное сообщение в виде json содержащее:  "error": "JSON field 'dateOfBirth' is missing or value not in 'YYYY-MM-DD' format." |
| 4 | Повторить шаг 2, <дата> является несуществующей датой, например, “2020-06-52”; и отправить POST запрос согласно шагу 3 | Статус код 400, получено ответное сообщение в виде json содержащее:  "error": "JSON field 'dateOfBirth' is missing or value not in 'YYYY-MM-DD' format." |
| 5 | Повторить шаг 2, <дата> не является датой, например, “123456”, “йцукен”; и отправить POST запрос согласно шагу 3 | Статус код 400, получено ответное сообщение в виде json содержащее:  "error": "JSON field 'dateOfBirth' is missing or value not in 'YYYY-MM-DD' format." |
| 6 | Повторить шаг 2, <дата> формата YYYY-MM-DD, является високосной датой, тогда как год не високосный, например, “2021-02-29”, “2019-02-29”; и отправить POST запрос согласно шагу 3 | Статус код 400, получено ответное сообщение в виде json содержащее:  "error": "JSON field 'dateOfBirth' is missing or value not in 'YYYY-MM-DD' format." |
| 7 | Повторить шаг 2, <дата> формата YYYY-MM-DD, старше текущей даты, например “2035-01-01”, “2045-12-30” и т.п.; и отправить POST запрос согласно шагу 3 | Статус код 400, получено ответное сообщение в виде json содержащее:  "error": "JSON field 'dateOfBirth' is missing or value not in 'YYYY-MM-DD' format." |
| 8 | Выполнить GET запрос по URL «129.146.247.102:5000/users» и попытаться найти пользователя <имя\_пользователя> заданного в шаге 3 | В БД нет пользователя <имя\_пользователя> с шага 3 |

**Ожидаемый результат:**

Пользователь не создан, получено сообщение о неверном формате даты, получено сообщение об ошибке «JSON field 'dateOfBirth' is missing or value not in 'YYYY-MM-DD' format.»

**9 Тест кейс**

**Идея:**

Обновление данных существующего пользователя при некорректном формате даты рождения

**Приоритет:**

Medium

**Описание:**

Не должно происходить обновление данных существующего пользователя при некорректном формате даты рождения

**Предусловие:**

* Публичный сервис по адресу 129.146.247.102 работоспособен
* Открыт внешний доступ к порту 5000

**Шаги:**

|  |  |  |
| --- | --- | --- |
| № | Описание | Ожидаемый результат |
| 1 | Создать нового пользователя через POST запрос по URL «129.146.247.102:5000/hello/<имя\_пользователя>», где: <имя\_пользователя> - желаемое имя пользователя содержащее только буквы, например “anatoliy”, “самойлов” и т.п. | Создан пользователь <имя\_пользователя> |
| 2 | В Postman создать новый запрос типа PUT | Создан новый PUT запрос |
| 3 | Прописать в тело json дату рождения вида <{"dateOfBirth": "<дата>"}>, где: <дата> – не соответствует формату даты YYYY-MM-DD, например “1 января 1995”, “2 августа 1983 г.” | Json содержит дату рождения в неверном формате |
| 4 | Задать URL «129.146.247.102:5000/hello/<имя\_пользователя>» и отправить запрос | Статус код 400, получено ответное сообщение в виде json содержащее:  "error": "JSON field 'dateOfBirth' is missing or value not in 'YYYY-MM-DD' format." |
| 5 | Повторить шаг 2, <дата> является несуществующей датой, например, “2020-06-52”; и отправить POST запрос согласно шагу 3 | Статус код 400, получено ответное сообщение в виде json содержащее:  "error": "JSON field 'dateOfBirth' is missing or value not in 'YYYY-MM-DD' format." |
| 6 | Повторить шаг 2, <дата> не является датой, например, “123456”, “йцукен”; и отправить POST запрос согласно шагу 3 | Статус код 400, получено ответное сообщение в виде json содержащее:  "error": "JSON field 'dateOfBirth' is missing or value not in 'YYYY-MM-DD' format." |
| 7 | Повторить шаг 2, <дата> формата YYYY-MM-DD, является високосной датой, тогда как год не високосный, например, “2021-02-29”, “2019-02-29”; и отправить POST запрос согласно шагу 3 | Статус код 400, получено ответное сообщение в виде json содержащее:  "error": "JSON field 'dateOfBirth' is missing or value not in 'YYYY-MM-DD' format." |
| 8 | Повторить шаг 2, <дата> формата YYYY-MM-DD, старше текущей даты, например “2035-01-01”, “2045-12-30” и т.п.; и отправить POST запрос согласно шагу 3 | Статус код 400, получено ответное сообщение в виде json содержащее:  "error": "JSON field 'dateOfBirth' is missing or value not in 'YYYY-MM-DD' format." |
| 9 | Выполнить GET запрос по URL «129.146.247.102:5000/users» и попытаться найти пользователя <имя\_пользователя> заданного в шаге 1 | Дата рождения пользователя <имя\_пользователя> не обновлена |

**Ожидаемый результат:**

Данные пользователя не обновлены, получено сообщение о неверном формате даты получено сообщение об ошибке «JSON field 'dateOfBirth' is missing or value not in 'YYYY-MM-DD' format.»

**10 Тест кейс**

**Идея:**

Получить поздравление с днем рождения для пользователя

**Приоритет:**

High

**Описание:**

Если текущая дата совпадает с датой дня рождения пользователя, то должно появиться поздравление пользователя с днем рождения

**Предусловие:**

* Публичный сервис по адресу 129.146.247.102 работоспособен
* Открыт внешний доступ к порту 5000

**Шаги:**

|  |  |  |
| --- | --- | --- |
| № | Описание | Ожидаемый результат |
| 1 | Создать нового пользователя у которого день рождения – текущая дата, с годом предшествующем текущему, через POST запрос по URL «129.146.247.102:5000/hello/<имя\_пользователя>», где: <имя\_пользователя> - желаемое имя пользователя содержащее только буквы, например “anatoliy”, “самойлов” и т.п. | Создан пользователь у которого дата дня рождения совпадает с текущей датой |
| 2 | В Postman сменить тип запроса на GET | Запрос сменен на тип GET |
| 3 | Задать URL «129.146.247.102:5000/hello/<имя\_пользователя>» и отправить запрос, где: <имя\_пользователя> - имя пользователя из шага 1 | Статус код 200, получено ответное сообщение в виде json содержащее:  - "message": "Hello, <имя\_пользователя>! Happy birthday!" |

**Ожидаемый результат:**

Получено сообщение с поздравлением для указанного пользователя вида «Hello, <имя\_пользователя>! Happy birthday!»

**11 Тест кейс**

**Идея:**

Получить информацию о количестве дней до дня рождения пользователя

**Приоритет:**

High

**Описание:**

Если текущая дата не является датой дня рождения пользователя, то должна появиться информации о количестве дней до дня рождения

**Предусловие:**

* Публичный сервис по адресу 129.146.247.102 работоспособен
* Открыт внешний доступ к порту 5000

**Шаги:**

|  |  |  |
| --- | --- | --- |
| № | Описание | Ожидаемый результат |
| 1 | Создать нового пользователя у которого день рождения – предшествует текущей дате на <N> дней, например 360 дней, через POST запрос по URL «129.146.247.102:5000/hello/<имя\_пользователя>», где: <имя\_пользователя> - желаемое имя пользователя содержащее только буквы, например “anatoliy”, “самойлов” и т.п. | Создан пользователь у которого дата дня рождения предшествует текущей дате на 7 дней |
| 2 | В Postman создать новый запрос типа GET | Создан новый GET запрос |
| 3 | Задать URL «129.146.247.102:5000/hello/<имя\_пользователя>» и отправить запрос, где: <имя\_пользователя> - имя пользователя с шага 1 | Статус код 200, получено ответное сообщение в виде json содержащее:  - "message": "Hello, <имя\_пользователя>! Your birthday is in <N> day(s).", где: <N> - количество дней до дня рождения согласно шагу 1, т.е. 5 для високосного и 6 дней для другого года |

**Ожидаемый результат:**

Получена информация о количестве дней до дня рождения пользователя, выводится сообщение «Hello, <имя\_пользователя>! Your birthday is in <N> day(s).»

**12 Тест кейс**

**Идея:**

Получить поздравление для несуществующего пользователя

**Приоритет:**

Low

**Описание:**

Не должно быть информации о дне рождения пользователя, которого нет в БД

**Предусловие:**

* Публичный сервис по адресу 129.146.247.102 работоспособен
* Открыт внешний доступ к порту 5000

**Шаги:**

|  |  |  |
| --- | --- | --- |
| № | Описание | Ожидаемый результат |
| 1 | В Postman создать новый запрос типа GET | Создан новый GET запрос |
| 2 | Задать URL «129.146.247.102:5000/hello/<имя\_пользователя>» и отправить запрос, где: <имя\_пользователя> - имя не существующего пользователя | Статус код 404, получено ответное сообщение в виде json содержащее:  - "error": "User with username '<имя\_пользователя>' does not exist." |

**Ожидаемый результат:**

При запросе поздравления для несуществующего пользователя выводится сообщение об ошибке «User with username '<имя\_пользователя>' does not exist.»

**13 Тест кейс**

**Идея:**

Удалить существующего пользователя из БД

**Приоритет:**

Low

**Описание:**

Пользователь с заданным в запросе именем должен быть удален из БД

**Предусловие:**

* Публичный сервис по адресу 129.146.247.102 работоспособен
* Открыт внешний доступ к порту 5000

**Шаги:**

|  |  |  |
| --- | --- | --- |
| № | Описание | Ожидаемый результат |
| 1 | Создать нового пользователя через POST запрос по URL «129.146.247.102:5000/hello/<имя\_пользователя>», где: <имя\_пользователя> - желаемое имя пользователя содержащее только буквы, например “anatoliy”, “самойлов” и т.п. | Создан пользователь <имя\_пользователя> |
| 2 | В Postman создать новый запрос типа DELETE | Создан новый DELETE запрос |
| 3 | Задать URL «129.146.247.102:5000/hello/<имя\_пользователя>» и отправить запрос, где: <имя\_пользователя> - имя пользователя с шага 1 | Статус код 200, получено ответное сообщение в виде json содержащее:  - "message": "User '<имя\_пользователя>' was deleted successfully." |
| 4 | Выполнить GET запрос по URL «129.146.247.102:5000/users» и попытаться найти пользователя <имя\_пользователя> заданного в шаге 1 | В БД нет пользователя <имя\_пользователя> с шага 1 |

**Ожидаемый результат:**

Пользователь удален из БД

**14 Тест кейс**

**Идея:**

Попытаться удалить пользователя, не существующего в БД

**Приоритет:**

Low

**Описание:**

Нельзя удалить несуществующего пользователя

**Предусловие:**

* Публичный сервис по адресу 129.146.247.102 работоспособен
* Открыт внешний доступ к порту 5000

**Шаги:**

|  |  |  |
| --- | --- | --- |
| № | Описание | Ожидаемый результат |
| 1 | В Postman создать новый запрос типа DELETE | Создан новый DELETE запрос |
| 2 | Задать URL «129.146.247.102:5000/hello/<имя\_пользователя>» и отправить запрос, где: <имя\_пользователя> - имя не существующего пользователя, например “tallechpochmak”, “zebralevsha” и т.п. | Статус код 404, получено ответное сообщение в виде json содержащее:  - "error": "User with username '<имя\_пользователя>' does not exist." |

**Ожидаемый результат:**

При запросе удаления несуществующего пользователя выводится сообщение об ошибке «"User with username '<имя\_пользователя>' does not exist."»