

Чек-лист базовых проверок для API:

### URL

- *Позитивный сценарий:* корректно ли срабатывает запрос с URL согласно документации?
- Что произойдет, если запросить не тот домен?
- Что будет, если поменять тип протокола?
- Является ли доменная часть регистрозависимой?
- Что будет, если изменить путь?

**Пример:** мы отправили два запроса — сайт с доменом [this-is-testing.com](http://this-is-testing.com) и сайт с доменом [THIS-is-TeStInG.com](http://THIS-is-TeStInG.com). Ожидаемый результат: запросы обрабатываются одинаково. Фактический результат: второй запрос возвращает ошибку — это баг.

### Методы

- *Позитивный сценарий:* корректно ли срабатывает запрос с методом согласно документации? Например, для теста post-запроса нам нужно не просто получить 201 статус, но и отправить дополнительно get-запрос, который покажет, что нужная сущность создавалась.
- Что получится, если изменить метод запроса?

**Пример:** есть корректно работающий get-запрос. Если мы поменяем метод на patch, должен возвращаться статус 405 Method Not Allowed, но возвращается 404 Not Found — это баг.

### Хедеры

- *Позитивный сценарий:* корректно ли срабатывает запрос с хедерами согласно документации?
- Все ли хедеры, указанные в документации, являются обязательными для передачи?
- Являются ли хедеры регистрозависимыми? Предварительно стоит посмотреть в документации или выяснить у разработчика, каким задуман регистр хедеров.
- Что будет, если передать несуществующий хедер?
- А если допустить ошибку в существующем?

**Пример:** разработчик говорит, что все хедеры должны передаваться только в нижнем регистре. При этом успешно отправляется запрос с хедером *CONTENT-TYPE* — это баг.

### Query-параметры (если есть)

- *Позитивный сценарий:* корректно ли срабатывает запрос с параметрами согласно документации?
- Если у параметров есть ограничения, соблюдаются ли они?
- Что будет, если передать не указанный в документации параметр?
- Как влияют на запрос ошибки в параметрах?

- Являются ли параметры регистрозависимыми? Предварительно стоит посмотреть в документации или выяснить у разработчика, каким задуман регистр параметров.

**Пример:** согласно документации, в get-запросе обязательно передавать параметры *limit* и *page*, чтобы не выгрузить всю базу данных целиком. Фактически запрос отправляется без этих параметров, приходит статус 500 Internal Server Error — это баг.

### Боди (если есть)

- *Позитивный сценарий:* корректно ли срабатывает запрос с боди согласно документации?
- Что произойдет, если не передать боди?
- Как передается боди в неверном формате? В абсолютном большинстве случаев мы имеем дело с боди в JSON.
- Что будет, если отправить JSON без обязательных ключей?
- Если у значений в боди есть ограничения, соблюдаются ли они?
- А если передать лишний, несуществующий ключ?
- Что произойдет, если допустить ошибку в формате JSON? Например:

- поставить неверные скобки,
- добавить запятую после последнего значения,
- убрать кавычки или двоеточия,
- поменять пары ключ-значение местами,
- не закрыть скобку,
- пропустить запятую,
- передать пустое значение (например, {“key”: } ).

**Пример:** согласно документации, в post-запросах отправляется JSON с боди формата:

```
{  
"id": "1ui3gh56nb"  
}
```

Напишем:

```
{  
"id": "1ui3gh56nb",  
}
```

Получим ошибку 403 Forbidden вместо 400 Bad Request с сообщением *JSON parse error* — это баг.

## Ошибки

- Корректный ли статус у каждого типа ошибок?
- Понятно ли из текста ошибки, что именно мы сделали неверно и как нам исправить запрос?

**Пример:** есть get-запрос <https://this-is-testing.com/applications>. Если запросить адрес <https://this-is-testing.com/TESTapplications>, возвращается ошибка 500 Internal Server Error вместо 404 Not Found — это баг.

## Доступы

- Чаще доступ к API предоставляется через токен, поэтому указанные проверки актуальнее всего для токенов. Однако похожие по логике тесты можно применить и к парам логин-пароль и другим типам авторизации.
- *Позитивный сценарий:* корректно ли срабатывает запрос с актуальным токеном?
- Что будет, если передать запрос без токена?
- Если у токена установлен срок действия, можно ли передать запрос по истечении этого срока?
- Что произойдет, если передать токен с ошибкой?
- Если есть разные типы токенов для разных запросов, можно ли отправить запросы с не соответствующими им токенами?

**Пример:** в системе есть два уровня доступов: admin и user. Только администратор со своим токеном может видеть все присланные на сайт заявки — через ручку <https://this-is-testing.com/applications>. При этом когда user, используя свой токен, отправляет запрос, он получает статус 200 Success вместо 403 Forbidden — это баг.

## Документация

- *Позитивный сценарий:* корректно ли срабатывают запросы, составленные строго по документации?
- Содержит ли документация все сведения, необходимые для запросов (путь, параметры, ключи-значения в боди)?
- Корректно ли указаны типы передаваемых и получаемых данных?
- Соответствуют ли запросы и ответы схеме, указанной в документации?

**Пример:** в документации указано, что для получения списка всех одобренных заявок необходимо запросить <https://this-is-testing.com/applications>. По факту этот запрос возвращает все заявки — как успешные, так и отклоненные. Чтобы получить только успешные заявки, нужна ручка <https://this-is-testing.com/applications/success>. В документации пропущена часть пути — это баг.