Обработка ошибок API

При разработке чат-бота на платформе Metabot, а также при интеграции с чат-ботом из внешних систем, советуем придерживаться описанных ниже рекомендаций.

Использование конструкций try..catch не поддерживается платформой на данный момент. Вы, конечно, можете так оборачивать код, но это ни к чему не приведет. Платформа сама, на системном уровне, обеспечивает безопасное выполнение кода и обработку исключительных ситуаций. Мы позаботились об этом: "под капотом" все безопасно, обернуто try..catch. Не нужно беспокоится, что код упадет по непонятным вам причинам.

Ошибки в чат-боте будут только там, где они должны возникать (например, ошибка в коде, синтаксисе, в обработке данных). Для всех случаев платформа Metabot вернет вызывающей стороне 500 ошибку.

При разработке прикладной бизнес-логики вы можете сами возвращать нужную ошибку, когда вам будет необходимо. Например, когда нужно сообщить о проблемах с валидацией или не валидности запроса. В этом случае платформа вернет 200 ответ, а в теле JSON ответа нам нужно добавить информацию об ошибке. Если не уверены, как лучше сформировать тело ответа, можете воспользоваться нашим рекомендованным стандартом, которые приведен ниже.

Все возможные коды ответа сервера

Код ответа HTTP	**Причина**	**Возможны ли доп. поле в тела ответа**
200	Все хорошо. В теле запроса возможны дополнительные ответы.	Да
401	Отсутствуют данные для авторизации или не верный токен авторизации.	Нет
403	Нет прав для выполнения запроса. Например, у пользователя API, для которого выдан токен, нет прав доступа к чат-боту.	Нет
404	Неправильный URL запроса. Возникает, когда точки интеграции нет или URL совсем не верный. Если URL верный, то значит кастомный внутренний эндпоинт не найден по алиасу.	Нет

Код ответа HTTP	Причина	Возможны ли доп. поле в тела ответа
500	Внутренняя ошибка. Например, ошибка в JavaScript коде чат-бота.	Нет

Тело ответа для 200 ответа

В случае успеха, чат-бот возвращает код ответа **200 OK**, а в теле запроса, содержится информация об успешном выполнении, например:

```
{
  "success": true
}
```

Если есть ошибка, то также возвращается **200 OK** и дополнительно возвращается текст ошибки в поле **message**, например:

```
{
  "success": false,
  "message": "Сообщение об ошибке"
}
```

В случае, если необходимо возвращать код ошибки, будет добавлен код ошибки, например:

```
{
  "success": false,
  "errorCode": 422,
  "message": "Ошибка валидации"
}
```

Версия #2
Юрий Гарашко создал 2023-06-12 10:59:07 UTC
Ирина Петрова обновил 2026-03-24 07:09:51 UTC