



Как возвращать текст в JSON с форматирование markdown, что бы не сломать структуру JSON?

Чтобы возвращать текст с Markdown-форматированием в JSON, не нарушая его структуру, необходимо правильно экранировать строку с Markdown.

Основная проблема заключается в том, что специальные символы, используемые в Markdown (например, переносы строк, кавычки, обратные слэши), могут конфликтовать со синтаксисом JSON и приводить к ошибкам парсинга.

Решение: Экранирование спецсимволов

Самый надежный способ — это обработать весь ваш Markdown-текст как единую строку и экранировать в ней все необходимые символы.

Ключевые правила экранирования для JSON:

- **Кавычки (")**: Заменяются на `\`.
- **Обратный слэш (\)**: Заменяется на `\\`.
- **Переносы строк**: Заменяются на `\n`.
- **Другие спецсимволы**: Такие как возврат каретки (`\r`), табуляция (`\t`) и другие, также должны быть экранированы.

Пример

Допустим, у вас есть следующий Markdown-текст:

```
# Заголовок

Это жирный текст.

- Элемент списка 1
- Элемент списка 2

[Ссылка на Perplexity](https://perplexity.ai)
```

Чтобы встроить его в JSON, он должен быть преобразован в одну строку с экранированными символами.

Правильный JSON:

```
{
  "content": "# Заголовок\n\nЭто жирный текст.\n\n- Элемент списка 1\n- Элемент спис
```

Автоматизация с помощью языков программирования

Вручную экранировать строки не рекомендуется, так как это трудоемко и легко допустить ошибку. Практически во всех языках программирования есть встроенные функции для сериализации данных в формат JSON, которые автоматически выполняют правильное экранирование.

PHP

Используйте функцию `json_encode()`.

```
$data = [
    'content' => "# Заголовок\n\nЭто жирный текст."
];

$jsonString = json_encode($data, JSON_UNESCAPED_UNICODE | JSON_PRETTY_PRINT);

echo $jsonString;
```

Вывод будет корректно отформатированным JSON.

JavaScript (Node.js)

Используйте `JSON.stringify()`.

```
const data = {
  content: "# Заголовок\n\nЭто жирный текст."
};

const jsonString = JSON.stringify(data, null, 2);

console.log(jsonString);
```

Python

Используйте `json.dumps()`.

```
import json

data = {
    'content': '# Заголовок\n\nЭто жирный текст.'
}

json_string = json.dumps(data, indent=2, ensure_ascii=False)

print(json_string)
```

Использование этих стандартных функций является лучшей практикой, так как они гарантируют создание валидного JSON, который может быть без проблем прочитан любым парсером на стороне клиента.