

Запомните

Как вы заметили из лекции, тестовый запрос выглядел вот так

```
{ "jsonrpc": "2.0", "method": "list_pets", "params": [ 1 ], "id": 0 }
```

а не так - { "jsonrpc": "2.0", "method": "list_pets", "params": { "limit": 1 }, "id": 0 }

То есть, хоть в документации и есть параметр limit, мы его не указали, и всё равно получили результат. Почему?

В JSON-RPC 2.0 существует два способа передачи параметров - позиционные параметры и именованные параметры. Оба варианта являются допустимыми и корректными, однако они предполагают разные структуры данных.

- 1. Позиционные параметры передаются в виде массива, и они предполагают, что сервер и клиент заранее знают порядок, в котором должны следовать параметры. В данном случае, параметры передаются в виде массива [1], и сервер должен знать, что это означает значение параметра limit .

```
{ "jsonrpc": "2.0", "method": "list_pets", "params": [ 1 ], "id": 0 }
```

- 2. Именованные параметры передаются в виде объекта, где каждому параметру присваивается конкретное имя. Это позволяет передавать параметры в произвольном порядке, и сервер и клиент заранее должны знать названия параметров.

```
{ "jsonrpc": "2.0", "method": "list_pets", "params": { "limit": 1 }, "id": 0 }
```

В обоих случаях сервер должен обрабатывать параметры соответствующим образом. Использование именованных параметров действительно помогает избежать путаницы с порядком параметров и делает код более понятным и читаемым. Однако для позиционных параметров нужно писать меньше кода.