# ВОПРОС № 5

Протокол RPC — это способ, при котором один сервис может как бы попросить другой сервис выполнить определённую функцию так, будто она вызывается локально, хотя на самом деле она запускается на удалённой машине. Когда проектируют такой протокол, главное сделать так, чтобы было удобно и понятно, какая именно функция вызывается, какие параметры ей нужно передать и как получить результат обратно. Всё должно быть максимально прозрачно, чтобы разработчику не нужно было заморачиваться, что там под капотом, а именно сеть, HTTP, сериализация и так далее.

Когда реализуют RPC поверх HTTP, то используют обычные HTTP-запросы, чаще всего POST, потому что туда удобно класть тело запроса. В теле указывается, какую функцию хотим вызвать и с какими параметрами. Часто это делается в формате JSON, потому что он читаемый и простой, можно прямо в виде объекта передать имя метода и аргументы. Например, в JSON можно отправить что-то вроде {"method": "add", "params": [5, 7]} — и это уже будет считаться RPC-запросом.

Данные передаются так, чтобы вызывающая сторона знала, что именно она хочет выполнить, а принимающая сторона могла точно понять, что за метод нужно запустить и с какими значениями. Обычно имя метода указывается как строка, а параметры — массивом или объектом. Когда запрос выполнен, обратно тоже прилетает JSON с результатом. В ответе может быть поле result — это значение, которое вернула функция, или error, если что-то пошло не так.

Если говорить о маршрутизации и балансировке RPC-запросов, то тут фишка в том, что маршрутизатор (например, NGINX) не смотрит на URL, как в REST, а смотрит на тело запроса. Он может по значению поля "method" понять, куда направить запрос. То есть, если вызывается метод "getTime", его можно отправить на один сервис, а метод "calculate" — на другой. Это удобно для микросервисов, где каждый сервис отвечает за свои функции.

Отличие от REST API в том, что в RPC чаще всего всё завязано на вызов конкретных методов, а не на работу с ресурсами. REST больше про работу с сущностями (например, получить список пользователей, добавить пользователя), и там используется чёткая структура URL, HTTP (GET, POST и т.д.). А в RPC всё завязано на метод и параметры, и структура URL вообще может быть одной и той же для всех запросов, вся суть находится в теле запроса.

Сериализация — это просто способ превратить данные (например, параметры запроса или ответ функции) в строку, которую можно передать по сети. Для сериализации чаще всего используют JSON, потому что он простой, легко читается человеком и поддерживается почти везде. В нём всё выглядит как обычные объекты или массивы, его легко читать и писать. Например, параметры [5, 7] превращаются в JSON-массив, а объект вроде { "username": "admin", "password": "1234" } — в JSON-объект.

В общем, RPC — это такой способ общения сервисов, когда хочется просто вызвать функцию на удалённой машине и получить результат, особо не парясь о том, что это сетевой запрос.