

REST یک مدل معماری برای طراحی برنامه‌های کاربردی شبکه است که ترکیبی از چند مدل معماری مبتنی بر شبکه می‌باشد و محدودیت‌هایی جهت تعریف یک واسط اتصال یکنواخت برای آن در نظر گرفته شده است.

REST مخفف شده‌ی عبارت **Representational State Transfer** و به معنای "انتقال بازنمودی حالت" است.

ایده اصلی معماری REST این است که به جای استفاده از مکانیزم‌های پیچیده‌ای مانند CORBA، RPC یا SOAP برای اتصال ماشین‌ها، از HTTP ساده برای برقراری ارتباط بین ماشین‌ها استفاده شود.

❖ سرویس‌هایی که معماری REST را پیاده‌سازی می‌کنند، وب سرویس‌های RESTful نامیده می‌شوند.

✓ این سرویس‌های سبک، قابل نگهداری و مقیاس‌پذیر، با ارائه روشی برای خواندن، ایجاد کردن، آپدیت کردن یا حذف اطلاعات از روی سرور، امکان همکاری بین سیستم‌های کامپیوتری روی شبکه اینترنت را فراهم می‌کنند.

به عنوان یک دستورالعمل برای مدیریت ارتباطات در یک شبکه پیچیده مانند اینترنت ایجاد شد

❖ مدل REST شش قید برای معماری برنامه‌های شبکه تعریف می‌کند: کلاینت سرور (client-server)

باشد. بدون حالت (stateless) باشد. قابلیت cache داشته باشد. (cacheable) سیستم لایه‌بندی شده

(layered system) داشته باشد. واسط یکنواخت (uniform interface) داشته باشد. دارای قابلیت کد

در صورت نیاز (code on demand) باشد. (که این محدودیت اختیاری می‌باشد) به سیستمی که این

قید را رعایت نماید. RESTful می‌گویند.

❖ **سادگی و قابل فهم بودن:**

یکی از بزرگ‌ترین مزایای REST، سادگی آن است. با استفاده از استانداردهای وب مثل HTTP، توسعه و استفاده از API ها آسان‌تر می‌شود. این سادگی به توسعه‌دهندگان امکان می‌دهد سریع‌تر و با خطای کمتری پروژه‌های خود را پیش ببرند.

❖ **انعطاف‌پذیری و مقیاس‌پذیری:** REST

❖ اجازه می‌دهد سرویس‌هایی با انعطاف‌پذیری بالا طراحی شوند که به راحتی می‌توانند با تغییر نیازها، توسعه یابند. این معماری به خصوص برای پروژه‌هایی که قرار است در مقیاس بزرگ اجرا شوند، بسیار مفید است.

❖ **سازگاری با انواع داده‌ها:** REST

❖ از انواع مختلف فرمت‌های داده پشتیبانی می‌کند. از جمله JSON، XML و HTML. این امر باعث می‌شود REST برای توسعه API هایی که قرار است با انواع مختلفی از کلاینت‌ها کار کنند، ایده‌آل باشد.

❖ **استفاده از متدهای HTTP معمول:**

❖ استفاده از متدهای HTTP استاندارد مانند GET، POST، PUT و DELETE در REST، توسعه API ها را برای توسعه‌دهندگانی که با پروتکل‌های وب آشنایی دارند، آسان‌تر می‌کند

❖ **قابلیت کش‌پذیری بالا:**

❖ با توجه به کش‌پذیر بودن درخواست‌ها در REST، عملکرد و کارایی سیستم به شکل قابل توجهی بهبود می‌یابد. این ویژگی به کاهش بار روی سرور و کاهش زمان لازم برای پاسخ‌گویی به کلاینت کمک می‌کند

❖ **استقلال زبانی:**

❖ توسعه‌دهندگان می‌توانند برای ساخت کلاینت‌ها و سرورهای REST از هر زبان برنامه‌نویسی که ترجیح می‌دهند استفاده کنند. زیرا این معماری تنها بر روی ارتباطات HTTP متمرکز است و به زبان خاصی وابسته نیست

سوالات

1. soap و rest تفاوت چیست؟

rest چند مورد است؟ قوانین و مفاد

چیست؟ Restful