به طور کلی، شش قانون در توسعه یک وب سرویس با استفاده از معماری RESTful در نظر گرفته می‌شوند:

**- Client-Server:** زمانی که نحوه توسعه‌ کلاینت ساید با سرور ساید از یکدیگر کاملاً **مجزا** باشد

**- Cacheable:** به طور کلی، پاسخ به درخواست‌های کلاینت می‌تواند در سمت سرور کَش شود که این سیاست به منظور بهبود پرفورمنس می‌تواند اتخاذ گردد.

**- Stateless:** به طور خلاصه، Stateless حاکی از آن است که هر درخواست از سمت کلاینت **مستقل** از دیگر درخواست‌ها است و زمانی که کلاینت درخواستی از جنس اچ‌تی‌تی‌پی ایجاد می‌کند، چنین درخواستی حاوی کلیه جزئیات و اطلاعات مورد نیاز سرور برای هندل کردن تَسک مربوطه است و سرور **هرگز** به داده‌های قبلی برای این کار اتکا نمی‌کند.

**- Uniform Interface:** یکپارچکی در توسعهٔ یک ای‌پی‌آی از جنس RESTful یک **باید** است مضاف بر اینکه به منظور حفظ یکپارچگی، هر ریسورس باید از اصول ثابتی تبعیت کند که از آن جمله می‌توان به نحوه نام‌گذاری، فرمت دیتا و ... اشاره کرد. وقتی توسعه‌دهندگان با بخشی از ای‌پی‌آی آشنا شدند، باید بتوانند با رویکرد یکسانی از سایر بخش‌های ای‌پی‌آی مذکور استفاده نمایند.

**- Layered System: ا**ین نوع معماری که مبتنی بر لایه‌های انتزاعی مختلفی است امکانی را در اختیار سیستم می‌گذارد تا هر کامپوننت صرفاً لایه‌ای که در حال تبادل داده با آن است را دیده و درگیر کامپوننت‌های دیگر نگردد.

**- Code on Demand:** این اصل که **اختیاری** می‌باشد حاکی از آن است که به منظور اجرای یک درخواست مبتنی بر این معماری، کلاینت می‌تواند کدی را در قالب اِپلِت جاوا یا اسکریپتی خاص در دیگر زبان‌ها دانلود و اجرا کند.