* **REST** - este un set de constrangeri arhitecturale, nu un Protocol. Fiecare poate implementa REST in foarte multe feluri
* **REST** este un stil arhitectural pentru a oferi standarte intre computer systems in web facand posibila comunicarea usoara intre sisteme.
* Cand un client face un request prin REST API, el primeste sau transfera state al unui object prin JSON, XML etc.
* Informatia e trimisa mereu prin HTTP
* Un REST API este un API ce se supune principiilor arhitecturale ale REST
* O aplicatie RESTful expune inormatii despre ea insasi, deci despre resursele sale. El ofera si posibilitatea clientului de a face modificari la resources, de a adauga etc.
* REST se cheama asa, deoarece se trimie o reprezentare la statusul unui object sau resursa
* REST se bazeaza pe 2 concepte cheie:
* Serverul va face o actiune in dependenta de **Endpoint** si **HTTP Method**

- **Client** - persoana sau soft care folosste API

- **Resource** - orice obiect despre care API ofera informatii

**Principii arhitecturale la REST**

1. **Uniform Interface** - acelasi requst, indiferent de cine e folosit si de unde, mereu returneaza acelasi rezultat, adica in aceeasi forma. Poate contine, optional, si pe HATEOAS

2. **Client-Server decoupling** - Client si Server sunt componente separate. Client nu poate interactiona direct cu componentele serverului si nici serverul cu clientul. Ei doar comunica prin requesturi

3. **Statelessness** - server nu memoreaza nimic despre client care foloseste API. Nu memoreaza gen daca user a mai faucut un GET inainte.

4. **Cacheability** - server returneaza date si ele pot fi cacheable, iar server trebuie sa informeze clientul daca ele sunt cacheable sau nu. Asta poate fi facut prin a trimite un verison number ca client sa stie daca datele ar putea fi vechi.

5. **Layered system architecture** - calls si responses pot sa treaca prin diferite layers. Clientul posibil sa nu comunice cu aplicatia finala, ci cu niste layere intermediare. Clientul nu trebuie sa stie ca comunica cu niste layers sau direct cu aplicatia. Aceste layers pot fi security layer, chaching layer etc.Aceste layers nu afecteaza response.

6**. Code on demand**(optional) - uneori, unele REST API pot trimite si vreun script executabil.