



REST Constraints

REST Constraints

1. Uniform Interface
2. Stateless Interaction
3. Client-Server
4. Cacheable
5. Layered System
6. Code on Demand (optional)

1. Uniform Interface

Defines the **interface** between client and server
Simplifies and decouples the architecture
Fundamental to RESTful design

??

HTTP verbs : GET, DELETE, PUT, POST

URI : Resource name

HTTP response : status and body

2. Stateless

Server contains **no client state**

Each request contains enough context to process

Self-Descriptive messages

Any session state is held on the client

3. Client-Server

Assume a disconnected system

Separation of concern (SoC)

Uniform interface is the link between the two

4. Cacheable

Server responses (representations) are cacheable

- Implicit
- Explicit
- Negotiate

5. Layered system

Client can't assume direct connection to server
SW / HW intermediaries between client and server

Improves scalability

Layered architecture !!

6. Code on demand

Server can temporarily extend client

By transfer logic to client

Client executes logic

Example => JavaScript

Summary

Compliance with REST constraints allow :

- Scalability
- Simplicity
- Modifiability
- Visibility
- Portability
- Reliability

<https://github.com/up1/course-rest-in-practice>