

1. **REpresentational State Transfer**.

Characteristics of a REST service:

* Client/Server
* **Stateless** – the server should not maintain client state. State is **sent** with each request.
* Cacheable – resources decide for themselves if they are cachable or not (and how long)
* Hypertext driven (in an ideal world)

Level 0: A **single url**, often accessible with **ONLY POST.**

Level 1: **Resources** – app defines different URIs for each resource.

Level 2: HTTP Verbs – the application uses GET/POST/PUT/PATCH/DELETE instead of putting the verb in the URL.

Level 3: Hypermedia controls – the application only exposes a single URI, and the client doesn‘t really need to know anything about the available resources, other than the root resource..

1. Can put version in the URL..

I.e. /api/v1/courses

So when v2.. we can still maintain v1 for some time.

Can also put it in **Accept header**: application/json;v2 or make a special header.

URL represents the resource (represent a course, not a version of it), **should not change.**

Headers describe how you‘d like the data (in HTTP).

Accept header: Harder to test/configure accept header properly.

