# Video 1:

## Tổng kết kiến thức series sau khai giảng

* Dùng @controller để biến java class thành 1 cái resfulapi webservice
* Truy xuất vào webservice thông qua url và method phải định nghĩa @RequestMapping
* @RequestParam nhận request đầu vào
* @ResponseBody trả json cho clien
* @RequestBody convert json sang object có thể chuyển hóa sang json
* Bên phía client có thể gửi nhiều type đầu vào ví dụ: text, json, html,…
* Response về cho client cũng có thể nhiều dạng dữ liệu

## Giới thiệu về @RestController

* Để build webservice ngoài @Controller còn có @RestController
* Khi mà dùng @RestController thì không cần phải sử dụng @ResponseBody(Để convert object sang json) nữa vì ở bên trong @RestController đã có @ResponseBody rồi

## Cách khác khi dùng @RequestMapping để định nghĩa url và http method

* Ngoài cách @RequestMapping(value, method)
* Còn có cách ngắn gọn hơn là GetMapping(url), PostMapping(url), PutMapping(url), DeleteMapping(url).

## Giới thiệu rule 8: don’t nest resourses

* Thay vì làm url kiểu cha con như /authors/12/articles/ tức là bên authors là 1 bên articles là nhiều và lấy id của authors là 12
* Thì làm url /articles/?author\_id=12 người khác nhìn vào dễ hiểu hơn

## Giới thiệu rule 4: dùng dạng số nhiều

## Khi đi làm phải thống nhất URL

* Bên frontend và backend phải thống nhất với nhau
* url tên là gì, method là gì, data trả ra có không

## Viêt 1 api xử lý nhiều chức năng

/api/users?buildingId=10&role=staff -> api load nhân viên trong giao tòa nhà

/api/users?role=manager -> api load toàn tộ ds quản lý

/api/users -> api load toàn bộ user