Rest

This is used to carry state (so it is said representation state) from the client side to server.
The connection is stateless this mean client has to send whole lot of data each time.
If server has to remember something then it creates session. It allocates the token so that next time the request is made then it should come with the token to remember we made a contact.
Rest is mainly used to transfer data in format (like json, xml, etc.) which is convenient instead of sending the design.
The url cannot have noun to make more sense.
It simplifies the design by using get, post, delete, put etc. action verbs.

Jackson XML:

Normal Jackson core supports only json conversion with rest api.
So if we send in header: Accept : application/xml, we will get no result.

So add dependency Jackson data format xml.
Then spring mvc will do the job for you.

But we can restrict that we will return in specific format.

```
@GetMapping(path = “alien”, produces = {“application/xml”})
```

So if asked for other format it will throw 406, not acceptable.

We can also restrict what format of data we will accept.

We can also say like what type of content we are sending through the request from the client in header:
Content-type: Application/json

We can use @RequestBody in parameter which converts the coming data in json format or xml format if Jackson xml is used.

But we can restrict what we will receive using consumes attribute:

```
@PostMapping(path=“alien”,
consumes={MediaType.APPLICATION.JSON.VALUE})
Public String addAlien(@RequestBody Alien a) {
	repo.save(a);
	return a;
}
```

Or

```
@PostMapping(path=“alien”, consumes={“application/json”})
Public String addAlien(@RequestBody Alien a) {
        repo.save(a);
        return a;
}
```

If we send now in xml, we will get 405 (content type not supported).