

REST?



CREATE

READ

UPDATE

DELETE

Uniform Interface and Easy Access:

HTTP Methods:

POST

GET

PUT

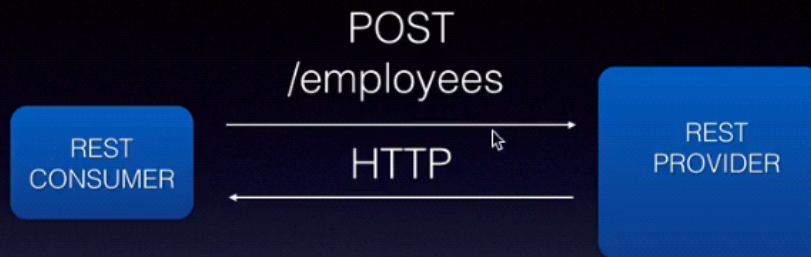
DELETE

VERBS

URI: /employees

NOUNS

CREATE



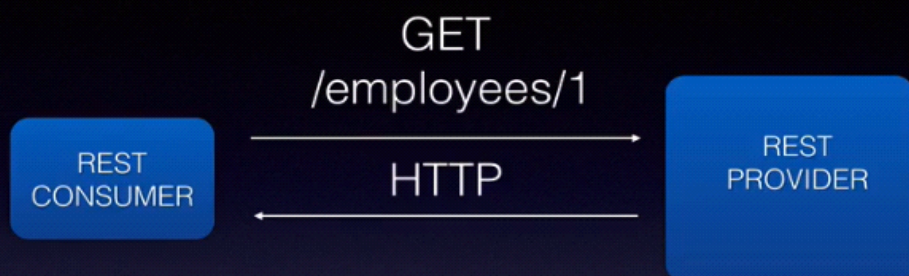
POST /employees

```
{  
  "name" : "bharath"  
  "dept" : "IT"  
}
```

201 Created

```
{  
  "name" : "bharath"  
  "dept" : "IT"  
}
```

READ



GET /employees/1

200 OK

```
{  
  "name" : "bharath"  
  "dept" : "IT"  
}
```


UPDATE



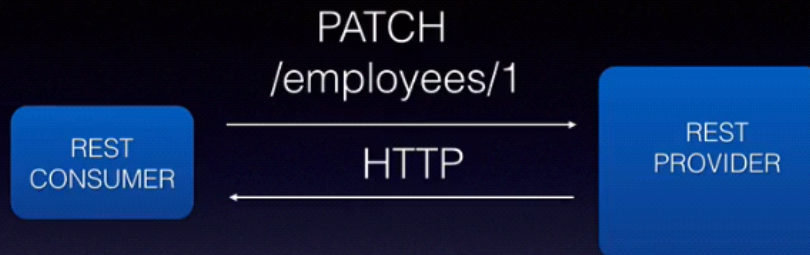
PUT /employees

```
{  
  "name" : "bharath"  
  "dept" : "psychology"  
}
```

200 OK

```
{  
  "name" : "bharath"  
  "dept" : "psychology"  
}
```

PATCH

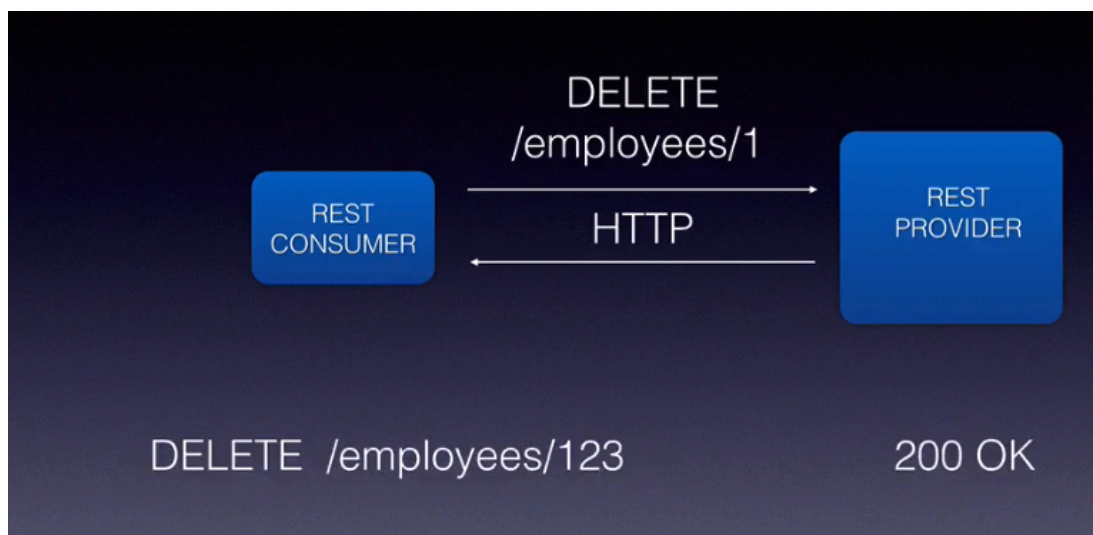


PATCH /employees

```
{  
  "dept" : "psycology"  
}
```

200 OK

```
{  
  "name" : "bharath"  
  "dept" : "psychology"  
}
```

MULTIPLE FORMATS:



```
<employee>  
  <name>john</name>  
</employee>
```

text/xml

```
employee:{name:john}
```

application/json

```
name=john
```

text/plain

RECAP:

Key REST Principles

SINGLE HTTP INTERFACE

EASY TO ACCESS

Multiple Representations

SPRING MVC or Spring WEB

spring-boot-starter-web

@RestController

@RequestMapping("/products")

Controller

Add Methods

@GetMapping

@PostMapping

@PutMapping

@DeleteMapping

Product

id

name

description

price

MySQL

JpaRepository

ProductRepository

/products

GET

findAll()

/products/{id} GET

findProduct(int id)

/products POST

createProduct(Product product)

/products PUT

updateProduct(Product)