HTTP has a few important verbs.

* POST - Create a new resource : Should create new resource. Ideally return JSON with link to newly created resource. Same return codes as get possible. In addition : Return code 201 (CREATED) is possible.
* GET - Read a resource : Should not update anything. Should be idempotent (same result in multiple calls). Possible Return Codes 200 (OK) + 404 (NOT FOUND) +400 (BAD REQUEST)
* PUT - Update an existing resource : Update a known resource. ex: update client details. Possible Return Codes : 200(OK)
* DELETE - Delete a resource

HTTP also defines standard response codes.

* 200 - SUCESS
* 404 - RESOURCE NOT FOUND
* 400 - BAD REQUEST
* 201 - CREATED
* 401 - UNAUTHORIZED
* 415 - UNSUPPORTED TYPE - Representation not supported for the resource
* 500 - SERVER ERROR

GET

@GetMapping("/students/{id}")

public Student retrieveStudent(@PathVariable long id)

* URL - http://localhost:8080/students/10002

@RestController

@RequestMapping("api/v1/")

public class ShipwreckController {

@RequestMapping(value = "shipwrecks", method = RequestMethod.GET)

public List<Shipwreck> list(){

return ShipwreckStub.list();

}

@RequestMapping(value = "shipwrecks", method = RequestMethod.POST)

public Shipwreck create(@RequestBody Shipwreck shipwreck){

return ShipwreckStub.create(shipwreck);

}

@RequestMapping(value = "shipwrecks/{id}", method = RequestMethod.GET)

public Shipwreck get(@PathVariable Long id){

return ShipwreckStub.get(id);

}

@RequestMapping(value = "shipwrecks/{id}", method = RequestMethod.PUT)

public Shipwreck update(@PathVariable Long id, @RequestBody Shipwreck shipwreck){

return ShipwreckStub.update(id, shipwreck);

}

@RequestMapping(value = "shipwrecks/{id}", method = RequestMethod.DELETE)

public Shipwreck delete(@PathVariable Long id){

return ShipwreckStub.delete(id);

}

}