@RestController

@RequestMapping("/fruits")

public class FruitController {

private final Logger LOG = LoggerFactory.getLogger(FruitController.class);

@Autowired

private FruitService fruitService;

@RequestMapping(method = RequestMethod.GET)

public ResponseEntity<List<Fruit>> getAll(@RequestParam(value = "offset", defaultValue = "0") int index,

@RequestParam(value = "numberOfRecord", defaultValue = "10") int numberOfRecord) {

LOG.info("Getting all fruits with index: {}, and count: {}", index, numberOfRecord);

List<Fruit> fruits = fruitService.getAll(index, numberOfRecord);

if (fruits == null || fruits.isEmpty()) {

return new ResponseEntity<List<Fruit>>(HttpStatus.NO\_CONTENT);

}

return new ResponseEntity<List<Fruit>>(fruits, HttpStatus.OK);

}

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

public ResponseEntity<Fruit> get(@PathVariable("id") int id) {

LOG.info("Getting fruit with id: {}", id);

Fruit fruit = fruitService.findById(id);

if (fruit == null) {

return new ResponseEntity<Fruit>(HttpStatus.NOT\_FOUND);

}

return new ResponseEntity<Fruit>(fruit, HttpStatus.OK);

}

@RequestMapping(method = RequestMethod.POST)

public ResponseEntity<Void> create(@RequestBody Fruit fruit, UriComponentsBuilder ucBuilder) {

LOG.info("Creating fruit: {}", fruit);

if (fruitService.exists(fruit)) {

return new ResponseEntity<Void>(HttpStatus.CONFLICT);

}

fruitService.create(fruit);

HttpHeaders headers = new HttpHeaders();

headers.setLocation(ucBuilder.path("/fruit/{id}").buildAndExpand(fruit.getId()).toUri());

return new ResponseEntity<Void>(headers, HttpStatus.CREATED);

}

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

public ResponseEntity<Fruit> update(@PathVariable int id, @RequestBody Fruit fruit) {

LOG.info("Updating fruit: {}", fruit);

Fruit currentFruit = fruitService.findById(id);

if (currentFruit == null) {

return new ResponseEntity<Fruit>(HttpStatus.NOT\_FOUND);

}

currentFruit.setId(fruit.getId());

currentFruit.setName(fruit.getName());

fruitService.update(fruit);

return new ResponseEntity<Fruit>(currentFruit, HttpStatus.OK);

}

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

public ResponseEntity<Void> delete(@PathVariable("id") int id) {

LOG.info("Deleting fruit with id: {}", id);

Fruit fruit = fruitService.findById(id);

if (fruit == null) {

return new ResponseEntity<Void>(HttpStatus.NOT\_FOUND);

}

fruitService.delete(id);

return new ResponseEntity<Void>(HttpStatus.OK);

}

}