Backend Development

Table of Contents

1. Java Coding Standard 1

1.1 General 1

1.2 Strings 1

2. REST API 1

2.1 General 1

2.2 Naming rules: 1

2.3 Pluralization: 2

# Java Coding Standard

## General

Please follow the recommendation defined in [Google Java Style Guide](https://google.github.io/styleguide/javaguide.html)

## Strings

For string concatenatin rather than use String.concat() or StringBuilder/StringBuffer.append() please use concatenation operator + .

Based on <https://docs.oracle.com/javase/specs/jls/se8/html/jls-15.html#jls-15.18.1>, + operator interpreted by Java compiler as StringBuffer.append() so does not have a performance overhead

# REST API

## General

Defining a REST API methods please follow the OPENAPI Specification: <http://swagger.io/specification/>

## Naming rules:

URI should make sense and adequately describes the resource. URIs must follow a predictable, hierarchical structure to enhance understandability and, therefore, usability: predictable in the sense that they're consistent, hierarchical in the sense that data has structure—relationships.

REST API call should leverage all the HTTP verbs (GET/POST/PUT/ DELETE/etc.) rather than specifying requested operation in a path

e.g. use *GET http://www.c4sg.com/users/{id}*

instead of GET *http://www.c4sg.com/users/search/byID/{id}*

Please follow the rule of using two base URLs for each root resource

One for creation of the resource within a collection

e.g. *POST http://www.c4sg.com/users*

and the second for reading, updating and deleting the resource by its identifier

e.g. *GET|PUT|DELETE http://www.c4sg.com/users/{id}*

In case we need to return a collection of objects we can just use first case with GET verb

e.g. *GET http://www.c4sg.com/users*

## Pluralization:

The commonly-accepted practice is to always use plurals in node names to keep your API URIs consistent across all HTTP methods. While a group of resources is a collection within the service you can refer a particular resource in the collection using id.

For example:

*GET http://www.c4sg.com/****users****/5/****organizations****/1/****projects****/2*

with ‘users’, ‘organizations’, and ‘projects’ URI nodes all being their plural forms.

It's acceptable to use a singularized resource name when there can only be one of the resource—it's a singleton resource. For example, if there was a single, overarching configuration resource, you might use a singularized noun to represent that

*e.f. GET|PUT|DELETE http://www.c4sg.com/users/7/profile*