Coding Standard

Basic Coding Rules:

* Use descriptive names for all identifiers (variables, method names, class names, constants, etc.).
* Comment every 3-7 lines of code.
* Avoid having comments for obvious parts of code

Identifier Naming and Capitalization:

* Use descriptive names for all identifiers (variables, method names, class names, constants, etc.).
* Avoid using Java reserved words or names that are similar to the reserved words.
* Use single letter identifiers only for the counter in loops.
* Class names start with an uppercase letter.
* Variable names start with a lowercase letter.
* Constants variable names should be all uppercase letters.
* Method names start with a lowercase letter.
* Multi-word identifiers are internally capitalized. (For example: sumOfSquares)
* Do not use hyphens or underscores to separate multi-word identifiers (For example: sum-of-squares)
* Boolean variables should start with “is”.

Comments: General

* Begins with a slash, followed by two asterisks (/\*\*)
* Begins with one asterisk (\*) on every comment
* At the end of the comment, add a new line and use (\*/)

Comments: Classes

* The comment should describe its purpose of the class.
* Class names start with an uppercase letter.

Example

| /\*\*  \* Stores the first, middle, and last names for a president.  \*/  class President {  //code...  } |
| --- |

Comments: Methods

* The comment should describe its purpose of the method.
* Method names start with a lowercase letter.

Naming Conventions for REST APIs

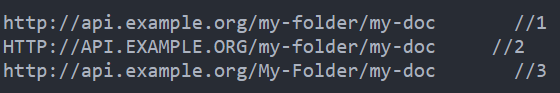
* Use forward slash ( / ) to indicate hierarchical relationships b/w resources



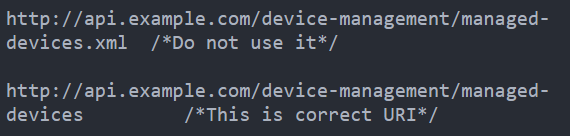
* Do NOT use trailing forward slash ( / ) in URIs



* Use hyphen ( - ) to improve readability of names in line path segments
* Do NOT use underscores ( \_ ) as replacements for hyphens ( - )

(1 & 2 same; 3 diff)

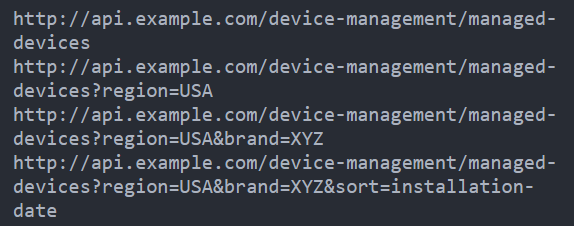
* Use lowercase letters frequently (rarely use capital letters)



* Do not use file extensions (instead specify it in Content-Type header to specify how to process body’s content)



* Never use CRUD (create, read, update, delete) function names in URIs; URIs only used to uniquely identify resources & NOT ANY ACTION UPON THEM
  + use HTTP request methods to specify CRUD function



* Use a query component to filter URI collection (DO NOT CREATE NEW APIs!!!)
* INSTEAD, enable sorting, filtering, pagination capabilities in resource collection API & pass input parameters as query parameters)