

Coding conventions

"Master programmers think of systems as stories to be told rather than programs to be written"

Must Follow:

1. All Groovy class names should be nouns, starting with capital letters.
2. Class name should not be plural.
3. Class name should be self explanatory (longer names are better than crisp, but non understandable names).
4. Abbrevated class names should be sensible
5. Standard Class structure should be followed

```
class MyClass{  
  
    /*Dependency Injections (e.g. Services) */  
  
    /*Fields */  
  
    /*Transients */  
  
    /*Relations */  
  
    /*Constraints */  
  
    /*Mappings */  
  
    /*Hooks */  
  
    /*Named queries */  
  
    /*Transient Methods */  
  
    /*Methods */  
  
    /*Static Methods */  
}
```

6. Package name should be in small case with structure as **'com.companyname.projectname.packagename'**
7. Variable names should be self explanatory, pronounceable and in camel case
8. Variable holding a collection should be plural.
9. Give appropriate Data Types for known variables rather than using def .e.g., def x = 10, is wrong

10. Use data type of variable holded by collection variable e.g., `List<User> users = User.list()`
11. Default value of a variable, if any, should be assigned at the time of declaration
12. Constant variables should be in capital letters and separated by an underscore e.g., `public static final int MAX_HEIGHT = 100`
13. Method name should be able to explain the intent of the method (longer names are better than crisp, but non understandable names). It should do just what its name suggests
14. Method name should be in camel case, and should start with a verb. Avoid using def as its return type
15. Smaller methods are better than longer ones
16. Step Down Rule – Code read from top to bottom, means all the called methods should be written after the calling method
17. DRY (DO NOT REPEAT YOURSELF)
18. AVOID more than 20 lines in methods
19. Pass well defined parameters (with descriptive names) instead of a map as parameter
20. Each line should contain at most one statement.
21. Wrap more than 4 args. to a function in a class.
22. Use binding by CO or params itself rather than updating each value
23. AVOID more than 10 lines in actions
24. Do all the manipulations using methods of services rather than in action itself
25. The controller action should sent the readily usable data to the view, i.e., any manipulation in the data (e.g., sorting, formatting, etc.) should be done in the VO rather than in the view.
26. Pass the the object rather than the whole params to the service or other methods
27. Pick one word per concept. e.g., it's confusing to have fetch, retrieve, and get as equivalent methods for different classes.
28. Follow standard domain structure
29. Write all the javascript in js files.
30. Make extensive use of taglib rather than doing calculations in gsp.
31. DO NOT write any database query in GSP, especially in the layout
32. gsp file names should be in camel case
33. Avoid in-line styling
34. Avoid writing "it" in closures
35. Always give attribute "var" in g:each
36. All the css statements should be clubbed into one block similarly all js statements
37. While using a tag, DO NOT send object id and then get the object again in tag, its good to send whole object in this case
38. Follow the indentation of 4,4,8, if there is no project specific indentation. Format you code as you write it
39. Do not write more than 120 character in one line (Line should be visible at one glance). If the line is more than 120 characters then break it to next line.
40. Line break should be logical like comma, brackets etc.
41. One blank line should always be used in the following circumstances:

- a. Between methods
 - b. Between the local variables in a method and its first statement
 - c. Between logical sections inside a method to improve readability
42. If your application need to do bootstrap it should be done by bootstrap service
 43. The application level constants should be in a separate file named as `ProjectNameConstants.groovy`
 44. Any Meta magic in bootstrap should be done via `MetaClassHelper.groovy` class in util package
 45. Remove Idea generated template
 46. DO NOT query in loops.
 47. DO NOT write negated if else conditions
 48. Use Ternary/Elvis Operators instead of simple if-else.
 49. Avoid multiple return statements.
 50. Remove all unused variables, methods, imports etc. Don't have anything in the code which is not used.
 51. If you have modified any jquery/javascript/grails plugin, rename the plugin by modified-pluginName.
 52. AVOID leaving printlns in you code
 53. Do not leave commented code. You can always get the code from history.
 54. Group logically related functions at one place. e.g., service for Security should have methods only related to security.
 55. DO NOT use magic numbers or words. Put such things in Constants which make it more readable
 56. Never ever do any operation in a println

Nice To Have

1. vo, co, enums etc., should be in `src/groovy` with their respective package name as `com.companyname.projectname.vo/co/enums` etc.
2. Declare variables inside methods just before its use
3. Follow The Thirty-Second Rule (Your method should be readable and its intend should be understandable wtithin 30 seconds)
4. Make extensive use of VO
5. Try to make Rich Domain classes
6. Make templates rather than having single big gsp file.
7. Application level js file should be named as `all.js` / `common.js` *
8. If a tag requires 2,3 parameters than send only those parameters, instead of the whole object
9. Avoid them as much as possible. They should be used if your method does multiple logical things at a time.
10. Avoid multiple if, else, else if. It shows the program is not object oriented.
11. Listen to your IDE. There are cases where your IDE suggest you the better way, unused things, unused assignments.

12. Always write a return statement, i.e., avoid using default last line returns