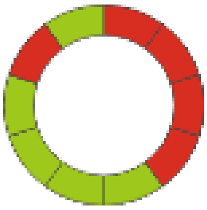
Better Code Hub[BETA]

SIGN OUT



rubenwiersma/thefirstorder

Last updated: May 27, 2016

YOUR TASKLIST

Write Short Units of Code

✕

Guideline

- Small units are easier to understand, reuse, and test.
- When writing new units, don't let them grow above 15 lines of code.
- When a unit grows beyond 15 lines of code, you need to shorten it by splitting it in smaller units of no longer than 15 lines of code.
- The list on the right side contains a sorted by severity selection of units that violate this guideline and the colors indicate the severity of the violation: □ more than 60 lines of code, □ more than 30 lines of code, □ more than 15 lines of code.

Refactoring candidates

- □ `material.js:DefaultUnit`
- □ `gulpfile.js:DefaultUnit`
- □ `project.state.js:stateConfig`
- □ `player.state.js:stateConfig`
- □ `camera.state.js:stateConfig`
- □ `map.state.js:stateConfig`
- □ `camera-action.state.js:stateConfig`
- □ `cue.state.js:stateConfig`
- □ `script.state.js:stateConfig`

Write Simple Units of Code

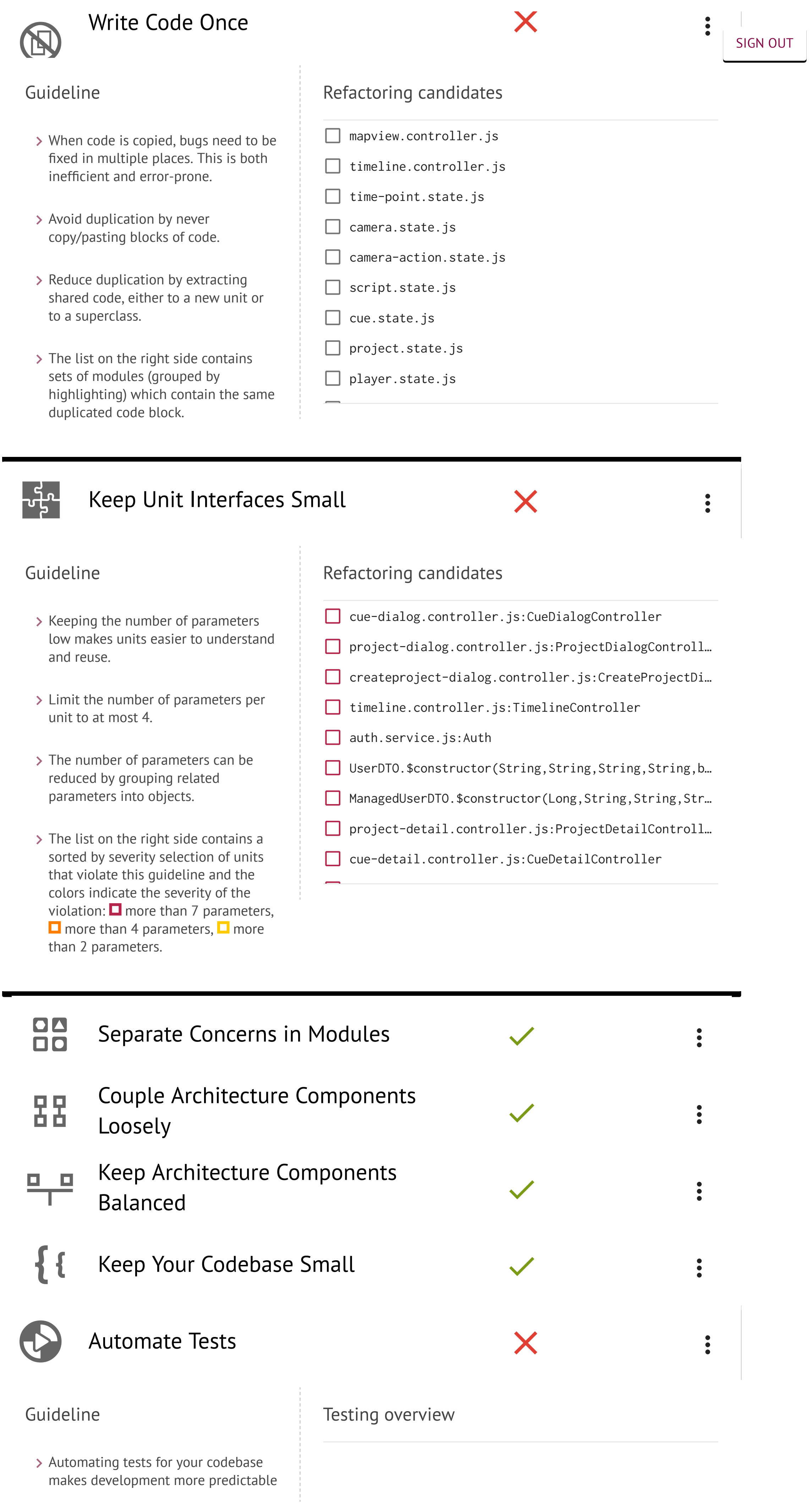
✕

Guideline

- Keeping the number of branch points (if, for, while, etc.) low makes units easier to modify and test.
- Try to keep the number of branch points in a unit below 5.
- You can reduce complexity by extracting sub-branches to separate units of no more than 5 branch points.
- The list on the right side contains a sorted by severity selection of units that violate this guideline and the colors indicate the severity of the violation: □ more than 25 branch points, □ more than 10 branch points, □ more than 5 branch points.

Refactoring candidates

- □ `material.js:DefaultUnit`
- □ `mapview.controller.js:MapviewController.drawCamera…`
- □ `alert-error.directive.js:jhiAlertErrorController`
- □ `auth.service.js:Auth.authorize.authThen`
- □ `material.js:_isChar`
- □ `ripples.js:Ripples.init`
- □ `state.handler.js:stateHandler.initialize`
- □ `AngularCookieLocaleResolver.parseLocaleCookieIfNec…`
- □ `JSR310LocalDateDeserializer.deserialize(JsonParser…`



Write Code Once ✗

Guideline

- When code is copied, bugs need to be fixed in multiple places. This is both inefficient and error-prone.
- Avoid duplication by never copy/pasting blocks of code.
- Reduce duplication by extracting shared code, either to a new unit or to a superclass.
- The list on the right side contains sets of modules (grouped by highlighting) which contain the same duplicated code block.

Refactoring candidates

□ mapview.controller.js
□ timeline.controller.js
□ time-point.state.js
□ camera.state.js
□ camera-action.state.js
□ script.state.js
□ cue.state.js
□ project.state.js
□ player.state.js

Keep Unit Interfaces Small ✗

Guideline

- Keeping the number of parameters low makes units easier to understand and reuse.
- Limit the number of parameters per unit to at most 4.
- The number of parameters can be reduced by grouping related parameters into objects.
- The list on the right side contains a sorted by severity selection of units that violate this guideline and the colors indicate the severity of the violation: 🟥 more than 7 parameters, 🟧 more than 4 parameters, 🟨 more than 2 parameters.

Refactoring candidates

□ cue-dialog.controller.js:CueDialogController
□ project-dialog.controller.js:ProjectDialogControll…
□ createproject-dialog.controller.js:CreateProjectDi…
□ timeline.controller.js:TimelineController
□ auth.service.js:Auth
□ UserDTO.$constructor(String,String,String,String,b…
□ ManagedUserDTO.$constructor(Long,String,String,Str…
□ project-detail.controller.js:ProjectDetailControll…
□ cue-detail.controller.js:CueDetailController

Separate Concerns in Modules ✓

Couple Architecture Components Loosely ✓

Keep Architecture Components Balanced ✓

Keep Your Codebase Small ✓

Automate Tests ✗

Guideline

- Automating tests for your codebase makes development more predictable

Testing overview

and less risky.

- Write unit tests that amount to at least 80% coverage.
- Add tests for existing code every time you change it.
- To update the status on the right, regularly send coverage information to Coveralls.io

72 %
Test coverage

Write Clean Code ✓