

# Contents

<b>1</b>	<b>Contributing</b>	<b>2</b>
1.1	Coding Guidelines . . . . .	2

# 1 Contributing

Please read through this document before contributing to Doubtfire.

Before continuing, **please read the contributing document<sup>1</sup> of the API**, as this outlines the Git workflow you should be following.

## 1.1 Coding Guidelines

For extendability and maintenance purposes, following these guidelines:

- Name a directive with it's role in mind (i.e., as a **Agent Noun**<sup>2</sup>) to give a small summary as to what the directive *does*:
- when *viewing* a project or task, the directive is project-**viewer** and task-**viewer**
- when *assessing* task submissions, the directive is task-submission-**assessor**
- when *editing* a unit's tutorials, the directive is unit-tutorial-**editor**
- Name directives that show lots of data in one directive in a table a list: e.g.:  
unit-student-list, group-member-list, project-top-task-list
- Name directives with a series of steps to perform a goal a **wizard**, e.g.:  
project-portfolio-wizard, new-user-wizard
- Always name modals in Pascal Case **SomeModal** and create them as a factory/controller pair CoffeeScript file which can then be easily created on the fly:

```
# foo/modals/create-foo-modal.coffee
angular.module('doubtfire.foo.modals.create-foo-modal', [])

#
# Prompts the user to create a Foo using a bar and qux variable
#
.factory('CreateFooModal', ($modal) ->
  CreateFooModal = {}

  CreateFooModal.show = (bar, qux) ->
```

---

<sup>1</sup>See <https://github.com/doubtfire-lms/doubtfire-api/blob/develop/CONTRIBUTING.md>

<sup>2</sup>See [https://en.wikipedia.org/wiki/Agent\\_noun](https://en.wikipedia.org/wiki/Agent_noun)

```
$modal.open
  templateUrl: 'foo/modals/create-foo-modal.tpl.html'
  controller: 'CreateFooModalCtrl'
  resolve:
    bar: -> bar
    qux: -> qux

CreateFooModal
)

.controller('CreateFooModalCtrl', ($scope, bar, qux) ->
  # Does stuff with bar and qux to create a foo
  $scope.bar = bar
  $scope.qux = qux
)

# foo/states/foo-view/foo-view.coffee
# ...
.controller('FooViewCtrl', ($scope, CreateFooModal) ->
  # ...
  $createNewFoo = ->
    CreateFooModal.show($scope.bar, $scope.qux)
)
```

- Always name non-anonymous controllers with a `Ctrl` suffix
- Case correctly:
- `directiveName` should be camelCase - refer to this Angular documentation<sup>3</sup>
- `ServiceName`, `ControllerNameCtrl` should be in PascalCase
- Regardless of abbreviations, stick to these conventions (e.g., `pdfPanelViewer` directive works, but `PDFPanelViewer` won't work as it needs to be camelCase)
- Place modals and states in a `modals` and `states` folder under the root. All else can be in their own folders unless they are of a related concept (see the `project-portfolio-wizard` folder under `project`, `stats` under `tasks` and `units`)

---

<sup>3</sup>See <https://docs.angularjs.org/guide/directive#normalization>

- The name of a module should follow the directory structure of where it has been placed (i.e., in the above example, the template file was at `foo/modals/create-foo-modal.tpl.html`, the CoffeeScript file was at `foo/modals/create-foo-modal.coffee`, and thus the module is `doubtfire.foo.modals.create`
- Try to give a brief summary of what the directive, state or factory does. E.g., the comment in the example above for `CreateFooModal` is sufficient.
- Try to abstract as much code inside a model class as possible. At present a lot of this code is in a model's service, and it should be moved into the model's resource definition as much as possible:

```
Unit.addTutorial tutorialData
```

instead of:

```
unitService.addTutorial unit, tutorialData
```