

We've built a simple app based on Apple's XCode template to show the most recent images available from Flickr's API.

We want to add 2 features to the `DetailsViewController` taking a test driven development (TDD) approach.

## Task: Preparation – Add tests to

### `DetailViewController`

`DetailViewController` currently has no tests at all. Add tests of `DetailViewController` to validate to following...

- Ensure that a `DetailViewController` instantiated from the storyboard has views.
- Ensure setting `detailItem` updates `detailDescriptionLabel`.
- Ensure setting `detailItem` initiates a download.
- Ensure successful image download updates `imageView`.

To achieve these goals you may wish to:

- Modify `DetailViewController` so it's `imageFetcher` is injected.
- Provide a protocol for `ImageFetching` to which `ImageFetcher` conforms.
- Implement a test mock for `ImageFetching`
- You don't need to add tests of `ImageFetcher` or other components now.

## Task: New feature – Show download failure message

Users would like to know when detail image download has failed.

Our design team have provided an image asset to be shown in the image view when an image download fails.

Following a TDD approach...

- Add test coverage that unsuccessful image download shows the failed download image.
- Fix the tests.

## Task: New feature – Add retry support

Users would like to be able to retry a failed image download in the detail view.

Discuss a design for this feature – simple is good.

Following a TDD approach...

- Add test coverage for the retry feature.
  - Fix the tests.
- 

## Discussion points

### `ImageFetcher` testing

---

In the current code, `ImageFetcher` has no tests. How would you add tests for it?

## Project architecture

---

How would you improve the architecture of this application so to better support ongoing:

- Testing
- Reuse of components
- Decoupling