Kanbanzilla Documentation

Release 1.0

Derek Ries, Peter Bengtsson

Sep 27, 2017

Contents

1 Running (and writing) tests 3

2 Compiling (making a distribution) 5

3 Writing code 7

3.1 Getting started 7

3.2 Actually Writing 8

4 Indices and tables 9

More details available on: https://wiki.mozilla.org/Kanbanzilla

This documentation is more about the code. How to run tests, how to make deployments and how to get started contributing.

CHAPTER 1

Running (and writing) tests

Tests are written in Jasmine and run through the Karma test-runner.

From the kanbanzilla directory, run the command:

```
grunt test
```

If you want to setup tests to run on save, from the kanbanzilla directory you can run the command:

```
karma start karma.conf.js
```

Configuration for the Karma task runner is handled in `karma.conf.js`

To start writing tests, place them in the appropriate architectural directory in `/test/spec/:type`

When writing tests in angular there's a few things to include. First you must load the module that you are running tests for, which can be accomplished with the function `module`. Also you will need to inject the appropriate services in whatever component you are testing with the `inject` function.

Example testing a controller:

```
beforeEach(module('kanbanzillaApp'));
var MyCtrl, scope;
beforeEach(inject(function ($controller, $rootScope) {
  scope = $rootScope.$new();
  MyCtrl = $controller('MyCtrl', {
    $scope: scope
  })
}));

// write your tests here
it('should do something', function () {
  expect(scope.something).toBe('something');
})
```

CHAPTER 2

Compiling (making a distribution)

When running Kanbanzilla in production, all the client side files are in the *./server/dist/* directory. This directory is never manually maintained. Instead it's generated with `grunt build` or alternatively `grunt`. The instructions for building a distribution of Kanbanzilla are contained in `Gruntfile.js` under:

```
grunt.registerTask('build', [
  // tasks
]);
```

The distribution is included with the git repository as a convenience feature, to deploy without having to have all of the proper project dependencies installed.

CHAPTER 3

Writing code

Getting started

To get started developing on Kanbanzilla you will need to install all of the application dependencies to run it locally. This can be accomplished by running:

```
bower install
npm install

cd server
pip install -r requirements.txt
```

Now you should have everything you need to run it locally. Kanbanzilla was built using Yeoman which includes many nice features, but is not without it's own problems. Yeoman and grunt use a node server to do live-reloading, compass compilation, and cleaning of directories, but the Kanbanzilla backend uses Python and Flask. For this reason you will need to run both servers:

```
grunt proxy server
```

and from the server directory:

```
DEBUG=true python api.py
```

there are several other options you can override when starting the flask server as well. To override the memcache URL use:

```
MEMCACHE_URL=128.0.0.2:8989 python api.py
```

To override what database to use, set `DATABASE_URI` like this:

```
DATABASE_URI="postgresql://localhost:5432/kanbanzilla" python api.py
```

Actually Writing

Every angular component has its' own sub-directory under the `app/scripts/` path.

If you installed yeoman and generator-angular, you can use the CLI to create new components. Their documentation is here: https://github.com/yeoman/generator-angular

CHAPTER 4

Indices and tables

- genindex
- modindex
- search

Index

C

compiling, 3

R

runningtests, 1

W

writingcode, 5