MANAGING CONFIGURATION

Storing user configuration options and sharing them between sub-generators is a common task. For example, it is common to share preferences like the language (does the user use CoffeeScript?), style options (indenting with spaces or tabs), etc.

These configurations can be stored in the `.yo-rc.json` file through the Yeoman Storage API (https://yeoman.github.io/generator/Storage.html). This API is accessible through the `generator.config` object.

Here are some common methods you'll use.

Methods

`this.config.save()`

This method will write the configuration to the `.yo-rc.json` file. If the file doesn't exist yet, the `save` method will create it.

The `.yo-rc.json` file also determines the root of a project. Because of that, even if you're not using storage for anything, it is considered to be a best practice to always call `save` inside your `:app` generator.

Also note that the `save` method is called automatically each time you `set` a configuration option. So you usually won't need to call it explicitly.

`this.config.set()`

`set` either takes a key and an associated value, or an object hash of multiple keys/values.

Note that values must be JSON serializable (String, Number or non-recursive objects).

`this.config.get()`

`get` takes a `String` key as parameter and returns the associated value.

`this.config.getAll()`

Returns an object of the full available configuration.

The returned object is passed by value, not reference. This means you still need to use the `set` method to update the configuration store.

`this.config.delete()`

Deletes a key.

`this.config.defaults()`

Accepts a hash of options to use as defaults values. If a key/value pair already exist, the value will remain untouched. If a key is missing, it will be added.

`.yo-rc.json` structure

The `.yo-rc.json` file is a JSON file where configuration objects from multiple generators are stored. Each generator configuration is namespaced to ensure no naming conflicts occur between generators.

This also means each generator configuration is sandboxed and can only be shared between sub-generators. You cannot share configurations between different generators using the storage API. Use options and arguments during invocation to share data between different generators.

Here's what a `.yo-rc.json` file looks like internally:

```
{
  "generator-backbone": {
    "requirejs": true,
    "coffee": true
  },
  "generator-gruntfile": {
    "compass": false
  }
}
```

The structure is pretty comprehensive for your end user. This means, you may wish to store advanced configurations inside this file and ask advanced users to edit the file directly when it doesn't make sense to use prompts for every option.