Endpoint Objects allow greater abstraction in test cases

```python
class ExampleApiTests(object):

    # create instances of endpoint
    color_endpoint = color.ColorEndpoint()

    def test_get_color(self):
       # setup: make API request
        data = '000000'
        res = self.color_endpoint.get_color(data)

        # test point: verify the correct response for a correct api call
        assert res.status_code == 200

        # test point: verify that we get back the same hex that we requested
        assert res.json()[0]['hex'] == data

        # test point: verify that the json keys are correct
        assert self.color_endpoint.verify_keys_in_response(res.json()[0].keys())
```

Custom Exceptions Support Functional Tests

```
class UnexpectedStatusCodeException(Exception):
    """
        Raise this exception when the server response code from an http request is not
        a success code.
        Capture the requests response object and make that available.

        # pass the response object to the exception
        >>> raise UnexpectedStatusCodeException(response=res)

        # then catch and introspect the exception's property
        >>> except UnexpectedStatusCodeException as e:
            ... print(e.response.status_code)
        400
    """
    def __init__(self, response):
        Exception.__init__(self, response)
        self.response = response


class PageIdentityException(Exception):
    """
        Raise this exception when a page fails its self validation of identity.
    """
    pass
```

Endpoint Objects allow greater abstraction in test cases

```
class ExampleApiTests(object):

    # create instances of endpoint
    color_endpoint = color.ColorEndpoint()

    def test_get_color(self):
       # setup: make API request
        data = '000000'
        res = self.color_endpoint.get_color(data)

        # test point: verify the correct response for a correct api call
        assert res.status_code == 200

        # test point: verify that we get back the same hex that we requested
        assert res.json()[0]['hex'] == data

        # test point: verify that the json keys are correct
        assert self.color_endpoint.verify_keys_in_response(res.json()[0].keys())
```