INDX documentation generation

Documentation is automatically generated by interpreting JavaScript and structured comments within the source code.

Config (config.js)

basePath	Where to look for files to document
filePaths	A list of files to parse. Order matters. If you have a class in file A that extends a class in file B, B should be before A. Globs may be used (e.g. js/*.js)
project	Attributes about the project put at the top of the documentation. • **Title** • **Version** • **Description** (in markdown)

Comment Style

JS source is parsed using regexps to extract classes and methods. For each of these, the preceding comment is parsed using a grammar.

Comment lines must begin with 3 slashes (///) to be used for documentation.

Files

Comments should be at top of file and contain any of the following (in this order):

- `@title`
- `@supplementary` - use to hide file from documentation but continue to refer to it from other files
- description

Example

```
/// @title Some Awesome Library
/// @supplementary
/// description
/// description continued
```

Classes

A class must have a name beginning with a capital letter and may be defined in any of the following forms:

- `Animal = function () {`
- `world.Animal = function () {`
- `function Animal () {`
- `something.awesome.Animal = SomeSuperClass.extend({ // Backbone's extend`

Methods

A method may be of the following form:

- `something.something = function () {`
- `something: function () {`

Comments should be above the definition and of the following form:

- `@arg <types> name: comment` - types are pipe (|) separated; types and comment are optional
- Result statement
    - Synchronous return:
      `@return <types> comment`
    - Asynchronous return:
      `@then <types> comment`
      `@fail <types> comment`
    - Chaining return:
      `@chain`
- description

For `@return`, `@then` and `@fail`, multiple result statements may be made. E.g.,

```
@return <string> if file was successfully read
        <boolean> if file failed to be read
```

Example (asynchronous method)

```
/// @arg <string|number> boxid: the id for the box
///
/// @then (<Box> the box)
/// @fail
///   (<{ code: 409 }> response) box already exists
///   (<{ code: -1, error: error obj }> response) other error
///
/// Attempts to create a box with the given ID
```

Building the docs

- `cd docs`
- `npm install`
- `node build.js`
  (if any errors are showing, nodejs might need upgrading -- v0.10.13 works for me)
- Everything will be built in ./build

Viewing/hosting the docs

host build in some way eg

- `cd build`
- `python -m SimpleHTTPServer 8080`

Abstracts

An abstract is a JS file which contains just the definitions of classes and methods. This is useful when you want to document parts of another library without modifying the library itself.

Grammars

The comments are parsed using Parsing Expression Grammars (PEG). Explaination of syntax can be found here: http://pegjs.majda.cz/documentation#grammar-syntax-and-semantics

Templates

Mustache templates are used to generate the documentation.

TODO

1. Go to line number
2. Superclass detail
3. Links to supplement classes
4. No returns if none provided

Properties

App	**File** name title description	**Class** name description	**Method** name description