

## butler Design

### recipe

`recipe` defines the schema recipes are stored with and provides a tool for reading a list of recipes from a directory called a recipe directory.

Recipes are stored in YAML files with structure:

```
name:          <NAME>

description:   <DESCRIPTION>

?ingredients:  [<INGREDIENT>|<COMPONENT>:<INGREDIENT>]]

?steps:        [<STEP>|<COMPONENT>:<STEP>]]

?notes:        [NOTE]
```

These YAML files are read into `Recipes` with structure:

```
class Recipe:

    Path          String          # Path to YAML file. Mandatory.

    Name          String          # 'name' in YAML file. Mandatory.

    Description    String          # 'description' in YAML file. Mandatory.

    Ingredients    NestableList    # 'ingredients' in YAML file. May be nested up to 2
                                   # levels enforced by the function that reads Recipes
                                   # from a directory. Nil if excluded.

    Steps          NestableList    # 'steps' in YAML file. May be nested up to 2 levels
                                   # enforced by the function that reads Recipes from a
                                   # directory. Nil if excluded.

    Notes          []String        # 'notes' in YAML file. Nil if excluded.

class NestableList []NestableItem

class NestableItem:

    Item string          # Value of the NestableItem.

    List NestableList    # Children of NestableItem.
```

YAML was chosen because it is human-readable, easily-parseable, and easily-writeable. A person not familiar with the format could see the YAML files and figure out how to write a well-formed recipe.

**ListFromDir** is a function accepts a directory name and returns a list of **Recipes** made from YAML files in that directory and an optional error. If a file couldn't be read, for reasons such as missing mandatory components or having too deeply nested lists, the error is non-nil. If the error is non-nil, none of the **Recipes** should be used. % TODO: Enumerate errors that can be returned from this.

**Path** is included in **Recipe** so that URLs or other identifiers can easily be generated to uniquely locate **Recipes**.

## Tests

Tests for **ListFromDir** assess that the correct output is provided for correctly and incorrectly formed YAML files. These test YAML files are kept in a directory called 'test' with sub-directories 'good' and 'bad'.

'good' contains well-formed recipe YAML files and verifies that these are properly read. 'good' also contains non-YAML files and makes sure they aren't included and no error is produced. Finally, 'good' includes sub-directories and makes sure they aren't recursed into.

'bad' contains mal-formed recipe YAML files with an additional field called 'err' that includes the name of the expected error to be returned.

An additional test verifies an **ErrNoDir** is returned if a directory name is passed which refers to a directory that doesn't exist.

% TODO: Why is it named recipe? % TODO: Errs for bad key types.

## Requirements

### Future Requirements

#### **book**

**book** is the recipe directory used for butler.

% TODO: Below should be integrated to recipe.

**book** is a collection of YAML files defining recipes. Each file has structure:

## **Testing**

The only testing will be proof-reading the recipes since the component only defines recipes.

## **Requirements**

3. Recipes all have a title, description, and optional ingredients, steps, and notes as enforced by `book`.

## **Future Requirements**

1. Pictures may have to be added to existing recipes.
2. Amounts may have to be added to existing recipes.

**gen**

**Components**

**Testing**

**Requirements**

**Future Requirements**

**source**

**Components**

**Testing**

**Requirements**

**Future Requirements**

**cmd/butler\_web**

**Components**

**Testing**

**Requirements**

**Future Requirements**

**cmd/butler\_server**

**Components**

**Testing**

**Requirements**

**Future Requirements**

**Scripts**