This repository | Search or type a command | Explore Gist Blog Help | OliPelz

mgonto / **angular-wizard**

Watch 10 | Star 199 | Fork 8

Easy to use Wizard library for AngularJS

26 commits | **2** branches | **4** releases | **3** contributors

branch: **master** | **angular-wizard** /

0.2.1

mgonto authored 6 days ago | latest commit 1a2ba1a222

dist	0.2.1	6 days ago
src	already runs the code. Removed function calling	6 days ago
test	Finished configuration for Grunt	8 days ago
.gitignore	First creation of Angular wizard with dist completed	8 days ago
.travis.yml	Project created	10 days ago
CHANGELOG.md	Project created	10 days ago
CONTRIBUTE.md	Project created	10 days ago
Gruntfile.js	First creation of Angular wizard with dist completed	8 days ago
README.md	Updated readme	8 days ago
bower.json	0.2.1	6 days ago
karma.conf.js	Finished configuration for Grunt	8 days ago
karma.underscore.conf.js	Project created	10 days ago
license.md	Project created	10 days ago
package.json	0.2.1	6 days ago

Code

Issues 3

Pull Requests 0

Pulse

Graphs

Network

HTTPS clone URL

https://github.com/m

You can clone with HTTPS, SSH, or Subversion.

Clone in Desktop

Download ZIP

README.md

Angular-Wizard

Angular-wizard is a component that will make it easy for you to create wizards in your app. You can check a running example of the wizard by clicking here

How do I add this to my project?

You can download this by:

- Using bower and running bower install angular-wizard
- Using npm and running npm install angular-wizard
- Downloading it manually by getting the files from the dist folder
- Using JsDelivr CDN files:

```
<!-- Use LATEST folder to always get the latest version-->
<script type="text/javascript" src="http://cdn.jsdelivr.net/angular.wizard/latest/angular-wizard.js
<link rel="stylesheet" type="text/css" href="http://cdn.jsdelivr.net/angular.wizard/latest/angular-

<!-- Or use TAG number for specific version -->
<script type="text/javascript" src="http://cdn.jsdelivr.net/angular-wizard/0.1.0/angular.wizard.js"
<link rel="stylesheet" type="text/css" href="http://cdn.jsdelivr.net/angular.wizard/0.1.0/angular-w
```

The dist folder contains the following files:

- JS files needed for the directives and services
- CSS files with default styles for the directive
- LESS file with styles for the directive. If you have less in your project, I recommend using the less instead of the CSS since it has variables to configure Wizard colors.

Dependencies

Angular-wizard depends on Angular and Lodash (or Underscore).

Starter Guide

First example

The first thing we need to do is add a dependency to angular-wizard module which is called `mgo-angular-wizard`.

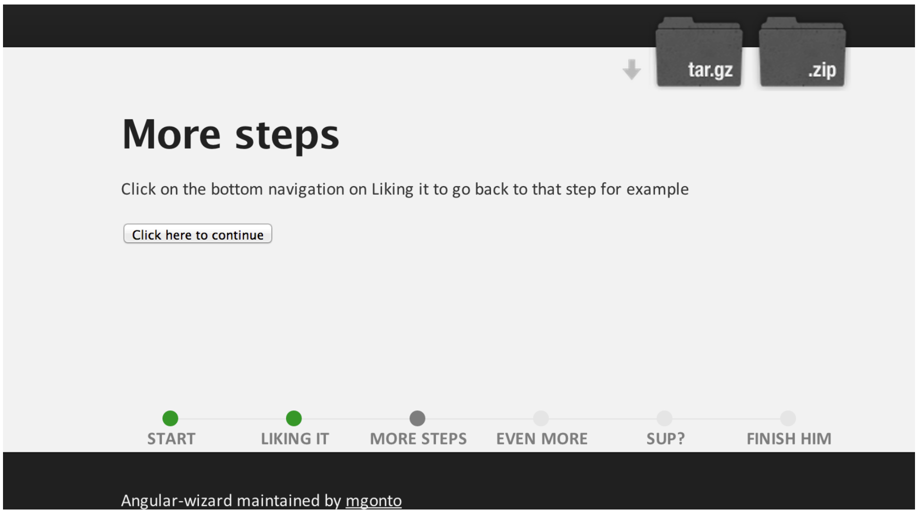
We can do this simply by doing:

```
angular.module('your-app', ['mgo-angular-wizard']);
```

Now, in some HTML for a controller, you can just add a wizard as follows:

```
<wizard on-finish="finishedWizard()">
    <step title="Starting">
        <h1>This is the first step</h1>
        <p>Here you can use whatever you want. You can use other directives, binding, etc.</p>
        <input type="submit" wz-next value="Continue" />
    </step>
    <step title="Continuing">
        <h1>Continuing</h1>
        <p>You have continued here!</p>
        <input type="submit" wz-next value="Go on" />
    </step>
    <step title="More steps">
        <p>Even more steps!!</p>
        <input type="submit" wz-next value="Finish now" />
    </step>
</wizard>
```

This will look like the following when you're in the second step:



More steps

Click on the bottom navigation on Liking it to go back to that step for example

Angular-wizard maintained by mgonto

Let's go step by step to see how this works.

1) You need to declare a master `wizard` directive. This wizard directive, has the following options as attributes:

- **on-finish**: Here you can put a function to be called when the wizard is finished. The syntaxis here is very similar to `ng-click`
- **name**: The name of the wizard. By default, it's called "Default wizard". It's used for the `WizardHandler` which we'll explain later.
- **edit-mode**: If set to true, this will set the wizard as edit mode. Edit mode means that all steps have been completed byt he can modify any of them now. Defaults to false.
- **hide-indicators**: If set to true, the indicators in the bottom of the page showing the current page and allowing navigation for the wizard will be hidden. Defaults to false.
- **current-step**: You need to set here a property from your scope (similar to `ng-model`) and that property will always have the name of the current step being shown on the screen.

2) Inside the wizard, we can have as many steps as we want. Each step MUST have a title which is going to be used to identify it. Inside each step, we can put whatever we want. Other directives, bindings, controls, forms, etc.

3) Inside the step, we now see a button which has a `wz-next` attribute. That means that clicking that button will send the user to the next step of wizard. Similar to `wz-next`, we have the following attributes:

- **wz-previous**: Goes to the previous step
- **wz-cancel**: Goes back to the first step
- **wz-finish**: Finishes the wizard and calls the on-finish later on. It's important to note that if we're in the last step and we put `wz-next` it'll be the same as putting `wz-finish` as the wizard will know we're at the last screen.

All of this attributes can receive an optional function to be called before changing the step. Something like:

```
<input type="button" wz-next="setMode(mode)" value="Next" />
```

In this case, the `setMode` function will be called before going to the next step.

Manipulating the wizard from a service

There're some times where we actually want to manipulate the wizard from the controller instead of from the HTML.

For those cases, we can inject the `WizardHandler` to our controller.

The main function of this service is the `wizard(name)` which will let you get the wizard to manipulate it. If you have just one wizard in the screen and you didn't set a name to it, you can just call it as `wizard()`. Let's see an example:

```
<step title="Cool step">
    <input type="submit" ng-click="changeLabelAndGoNext()" />
</step>
```

```
// In your controller
$scope.changeLabelAndGoNext = function() {
    $scope.model.label = "Hola Gonto";
    WizardHandler.wizard().next();
}
```

In this case, we're changing a label and moving forward on the steps. The functions available in the `wizard()` are:

- **next**: Goes to the next step
- **previous**: Goes to the previous step
- **finish**: Finishes the wizard.
- **goTo(number|title)**: This goes to the indicated step. It can receive either the number of the step (starting from 0) or the title of the step to go to.

Navigation bar

The navigation bar shown below works in the following way:

- Completed steps are painted as green
- Current step is painted as dark grey
- Future step is painted as light grey
- Editing step (Modifying a step already completed in the past) is painted as red
- You can click in any completed step to go back to that step. You can't click in the current step nor in the future ones unless you've already completed a future step before (for example in EditMode all steps are completed by default)

All of those colors are variables in the `angular-wizard.less`. You can easily change them by changing the colors in that file

Contributors

@sebazelonka helped me with all fo the styles of the Wizard.

Live sample

You can check out a live sample of the Wizard clicking here

Releases Notes

Releases notes are together with releases in GitHub at: https://github.com/mgonto/angular-wizard/releases

License