How To Create Single Page Application in minutes! with AngularJs and Yii 2.0

By Hafid Mukhlasin



How To Create Single Page Application in minutes! with AngularJs and Yii 2.0

# Open Your Mind

I will show You about How To Create Single Page Application (SPA) in minutes with using AngularJs and Yii Framework 2.0. This client server web application will simulate a Create Read Update Delete (CRUD) data processing. Where user interface of this application will full control by AngularJs, then data will provide by Yii Framework 2.0 as web services.

So, it will be two applications:

1. Web Client Application
2. Web Service Application

# Technology Behind Scene

**For web client app**



I use 3 web standard language, they are HTML, JS, and CSS.



I use AngularJs as Javascript Framework and Twitter Bootstrap as CSS Framework.

**For web service**

 I use PHP and MySQL. Yii Framework 2.0 as PHP Framework

Development tools, I use XAMPP and Notepad++ or Netbeans.

# Preparing Our Gun

1. For Web Client

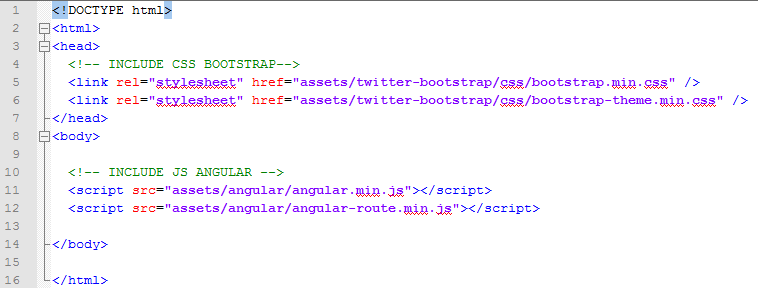
I create folder “web-client” in htdocs, then create 4 folder inside it. They are assets, controllers, models, views.

* Assets folder for core angularJs and TwitterBootstrap,
* Controllers folder for javascript controller of angularJs
* Models folders for services, connecting to Restful CRUD URL Yii 2.0
* Views folder for partial page, is like views in Yii ☺

Download http://getbootstrap.com/, and extract to folder assets

Download <https://angularjs.org/>, and extract to folder assets

Create file index.html in web root (web-client folder)



Done

1. For Web Service

Install Yii 2.0 Basic https://github.com/yiisoft/yii2-app-basic

composer global require "fxp/composer-asset-plugin:1.0.0"

composer create-project --prefer-dist yiisoft/yii2-app-basic web-service

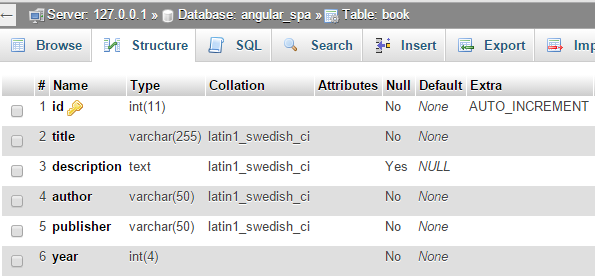
Prefered way using composer, but You can install manual checkout in http://yiiframework.com

# Getting Started

## Create Database Structure with PHPMyAdmin

Create database “angular\_spa”

Create “book” table with following structure

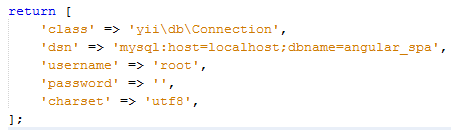


Input some datas

## Create Web Service with Yii 2.0

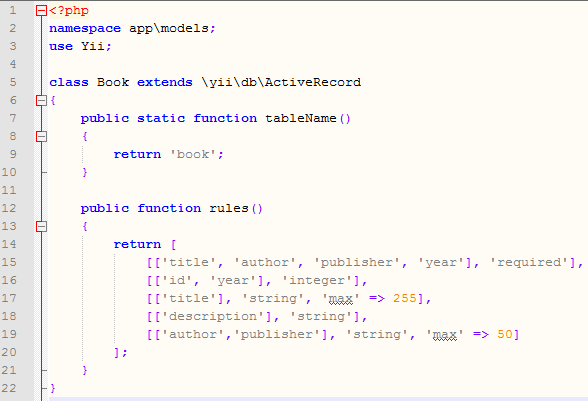
### Config Yii to connect database

Open config/db.php in root of Yii application (web service), modify db configuration



### Create model of book table

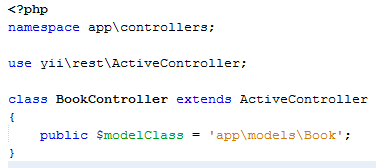
Create file Book.php in folder models, You can create via Gii or manual..



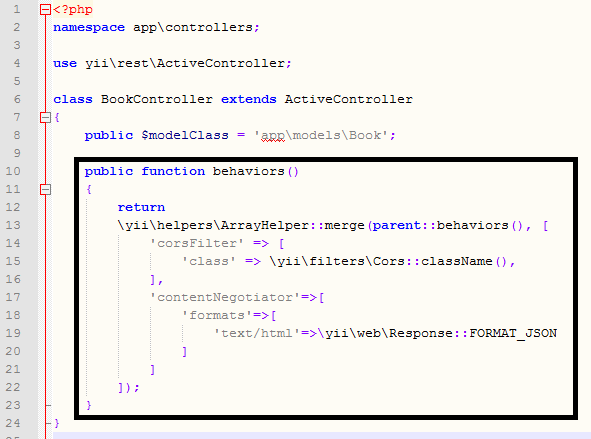
### Config Yii as Restfull App

Please read from this guide <https://github.com/yiisoft/yii2/blob/master/docs/guide/rest-quick-start.md>, its good start, to explore restfull in Yii

#### Create BookController.php in folder controllers

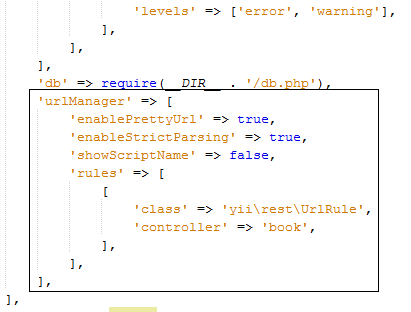


Add behavior for CORS to unblock cross origin, and set default response of REST to be JSON (optional).

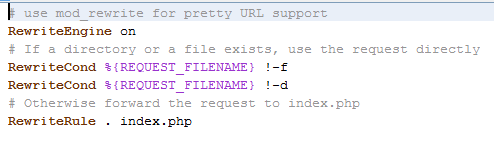


#### Configuring URL Rules – config\web.php

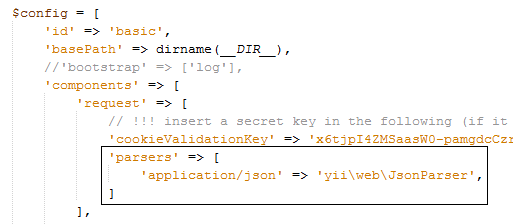
In rules You must add class yii\rest\UrlRule and define controller rest (its array), controller => [‘book’,’blabla’,’’]



#### Add .htaccess in web root (htdocs/web-service/web/), to hide index.php from url



#### Enabling JSON Input

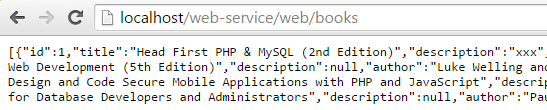


#### Trying it Out

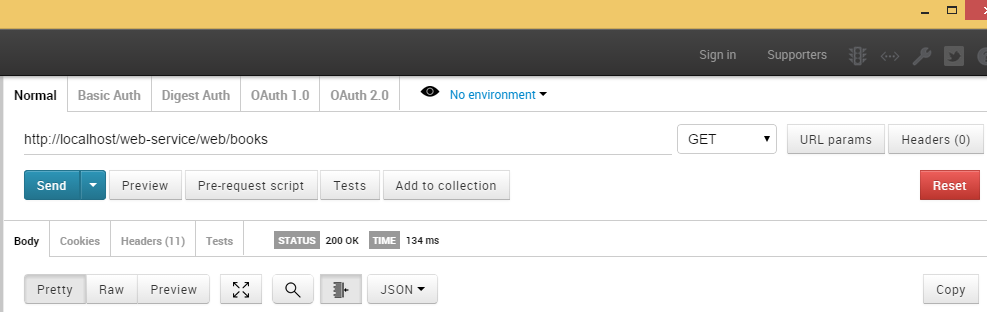
With the above minimal amount of effort, you have already finished your task of creating the RESTful APIs for accessing the user data. The APIs you have created include:

* GET /books: list all books page by page;
* HEAD /books: show the overview information of book listing;
* POST /books: create a new book;
* GET /books/123: return the details of the user 123;
* HEAD /books/123: show the overview information of book 123;
* PATCH /books/123 and PUT /books/123: update the book 123;
* DELETE /books/123: delete the book 123;
* OPTIONS /books: show the supported verbs regarding endpoint /books;
* OPTIONS /books/123: show the supported verbs regarding endpoint /books/123.

Info: Yii will automatically pluralize controller names for use in endpoints. You can configure this using the [[yii\rest\UrlRule::$pluralize]]-property.



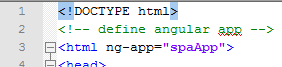
You can testing with Postman (Chrome Extensions)



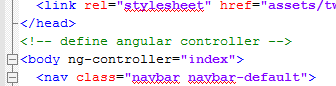
## Create Web Client with AngularJs + Twitter Bootstrap

### Modify index.html atau main layout

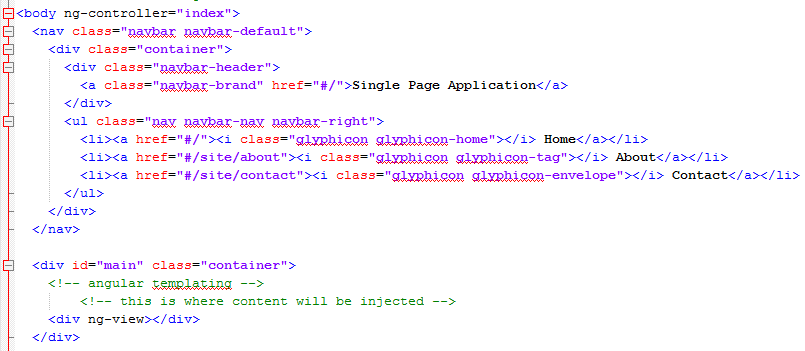
Attaching ng-app attribute to html tag to define angular app,



Attaching ng-controller attribute to body tag to define default angular controller,

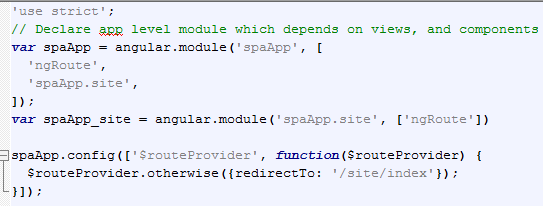


Add this content inside body tag, its are Navbar for menu, main for page container, and footer



Important thing is main container.. <div ng-view></div>, in this div will placed the dynamic content from other file or page view.

### Create app.js in web root



Default route is /site/index, this route will handled by sub module “spaApp.site”

### Create file site.js in folder controller

This file is controller to handle site views, maybe in Yii is like as SiteController.php

This file handle routing of angular app for sub module site. Every route have may templateUrl and controller.

TemplateUrl is other file html as partial content

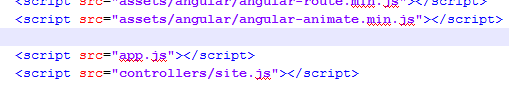
Controller is name of controller that do some process, for example set variable in template.

$scope.message, message is variabel in file templateUrl, let say views/site/index.html, point to {{message}}



If You Yiiers, Sub module in angular maybe is like as Controller in Yii, and .controller in angular is like as function of controllers in Yii.

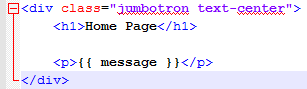
### Include app.js in index.html

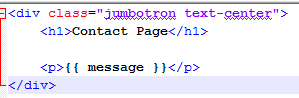


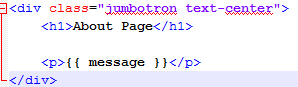
Okey, after it then create views

### Create views/site/index.html, views/site/contact.html, views/site/about.html

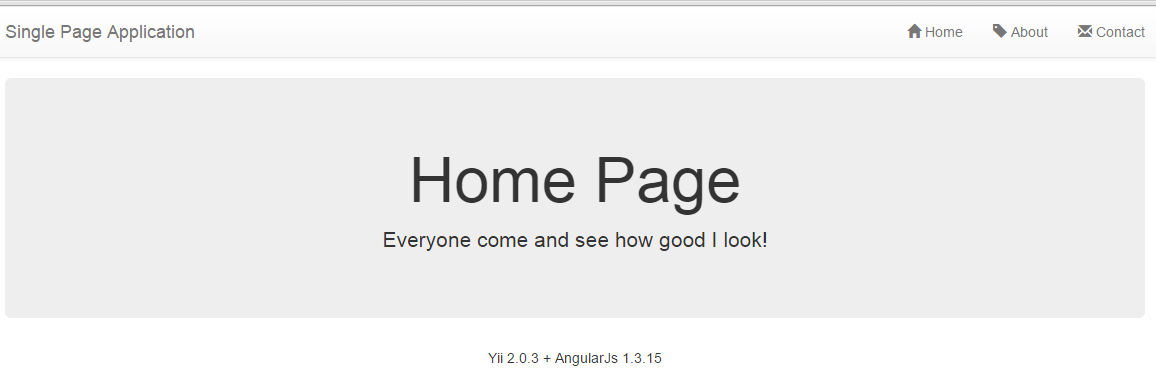
This is only example





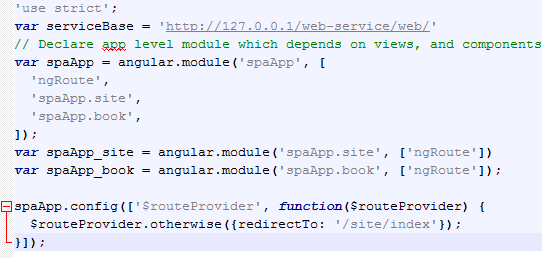


### Testing Your Angular App



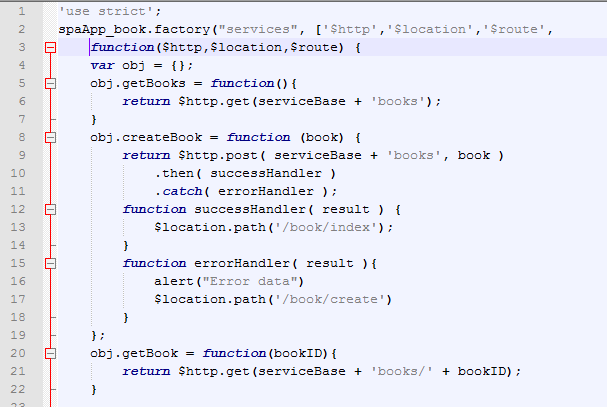
### Modify app.js

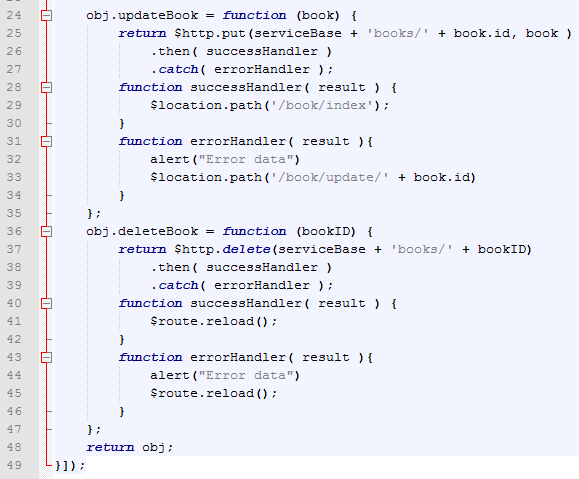
Add js global varabel serviceBase that refer to Your Yii 2 web service, then add sub module spaApp.book



### Create file book.js in folder models

book.js will handle CRUD data from Rest Yii 2. I force You understand it as models in Yii 2 ☺





To understand it, so simple.. focus to function of objects: obj,getBooks, obj.createBook, etc. And in every function, there are process to shoot to Rest Yii.. for example

obj.getBooks = function(){

return $http.get(serviceBase + 'books');

}

For get list of the book use method GET, see in <https://github.com/yiisoft/yii2/blob/master/docs/guide/rest-quick-start.md>,

obj.createBook = function (book) {

return $http.post( serviceBase + 'books', book )

create book use method POST

obj.updateBook = function (book) {

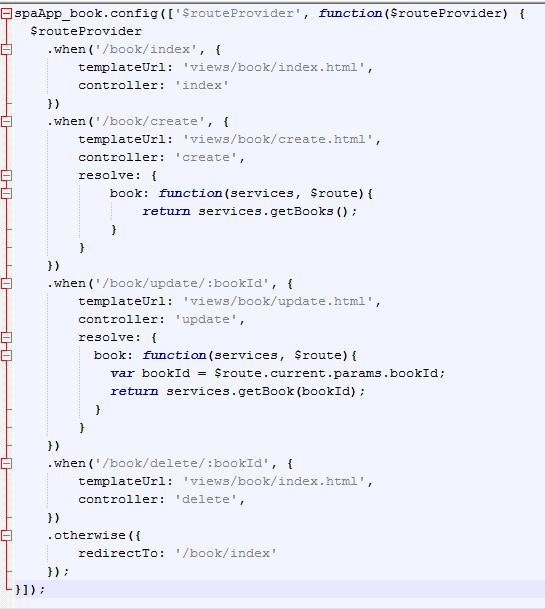
return $http.put(serviceBase + 'books/' + book.id, book )

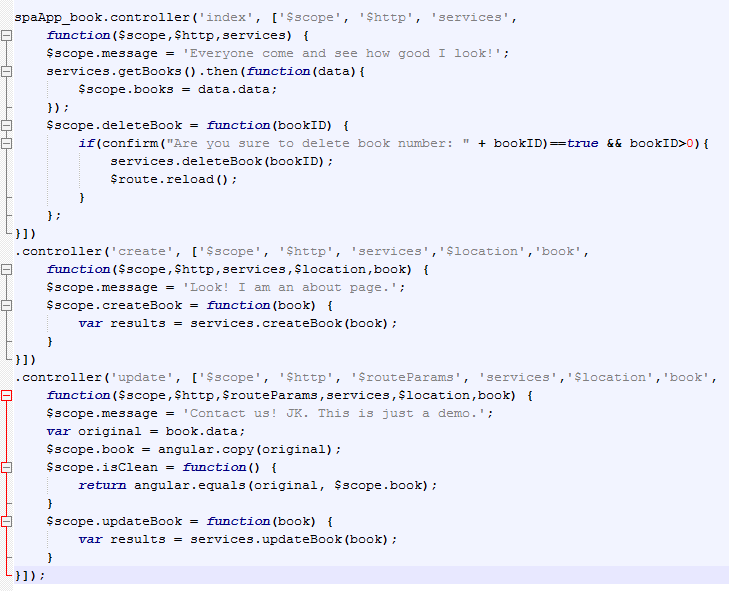
update book use PUT method..

Okey, sos simple, You understand?

### Create file book.js in folder controller

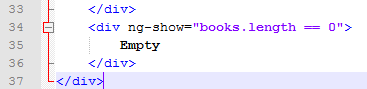
This file is controller to handle book views, maybe in Yii is like as BookController.php



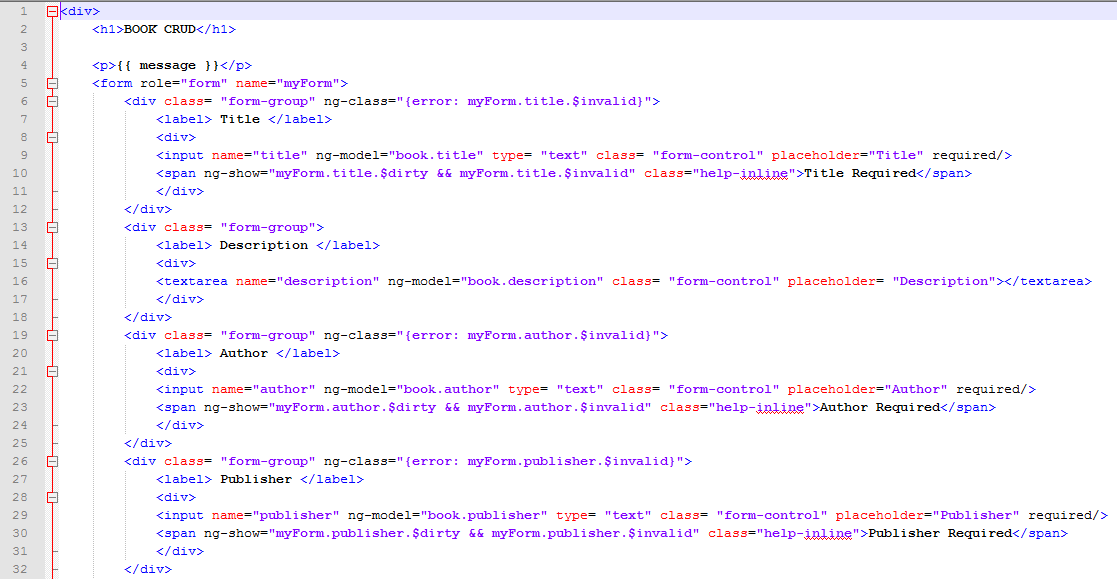


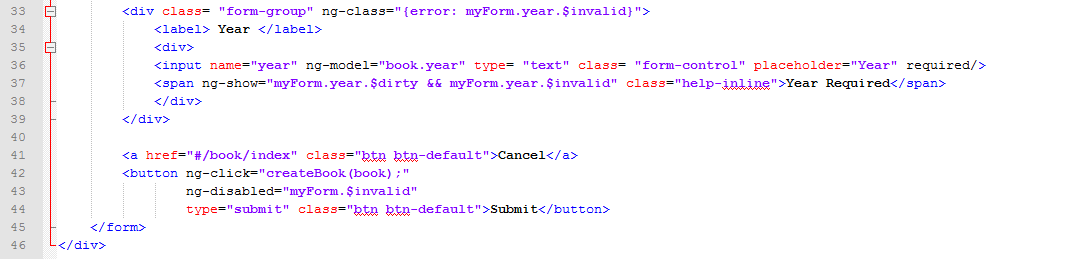
### Create index.html in folder views/book/



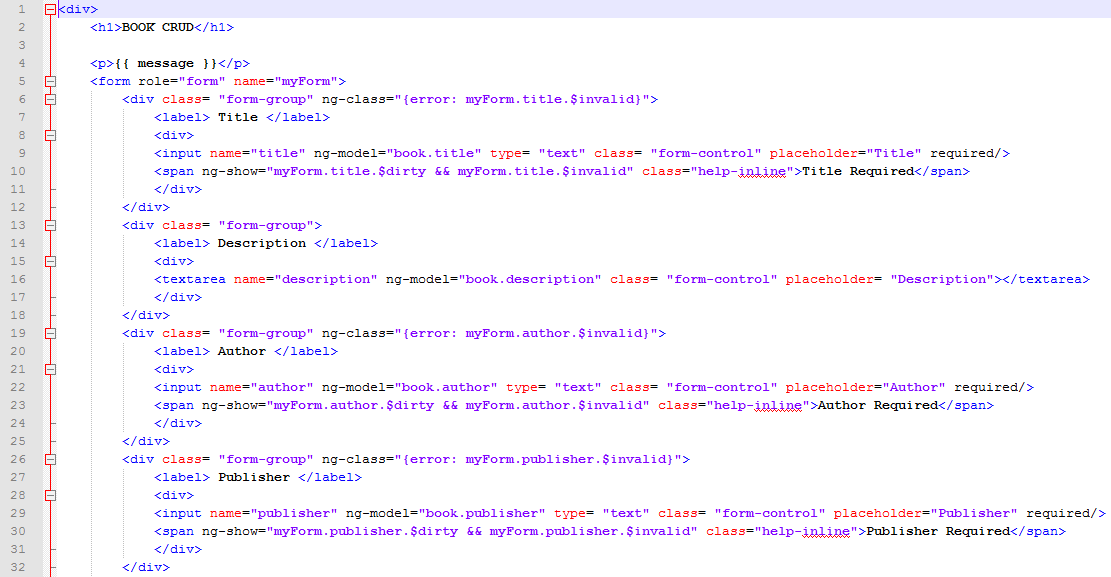


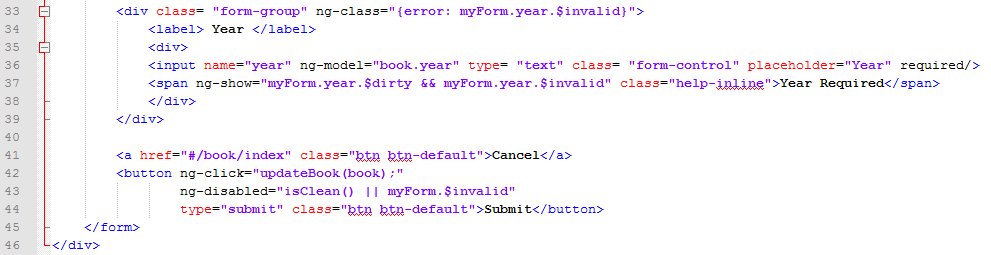
### Create create.html in folder views/book/



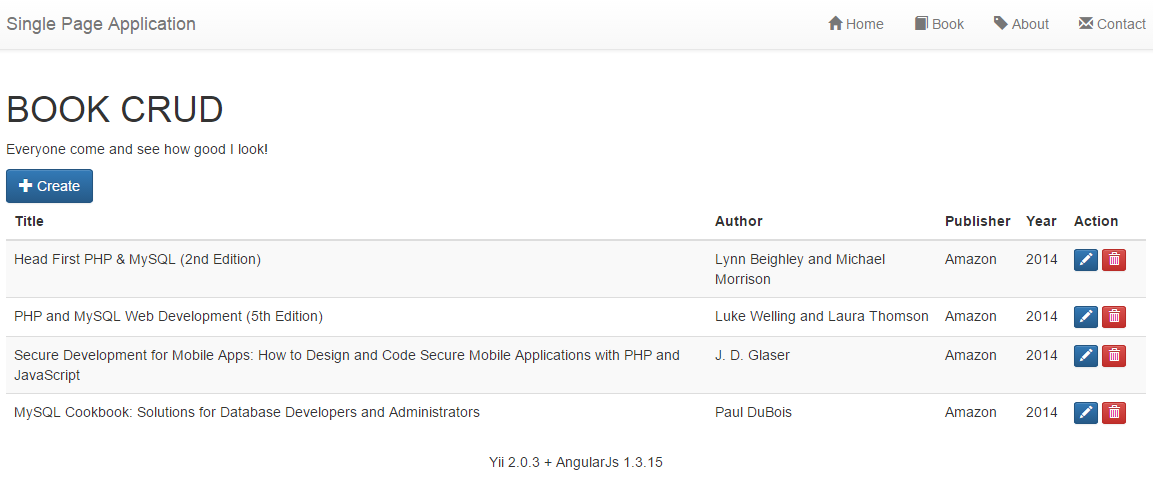


### Create update.html in folder views/book/





### Testing now



# How about You?

Can You create CRUD for some tables?

# Contact Us

+6281559915720

[hafidmukhlasin@gmail.com](mailto:hafidmukhlasin@gmail.com)

http://www.hafidmukhlasin.com