

# Chapter I - Introduction



## Little bit of history

If you have already gone through the tutorial set that I have published at [tutewall.com](http://tutewall.com) (<http://tutewall.com/laravel-backbone-js-coffeescript-tutorial-part-01/>), you may already know most part of this book. This tutorial series has been so successful that well-known forums and blogs have referred to it. I initially wrote it in order to educate myself on how to use Backbone.js with CoffeeScript. While I was writing applications I realized that it is a good topic for tutorial series so that most who are new to Backbone.js or reluctant to use CoffeeScript may find it useful if I give proper illustration of why these two languages going well together. Moreover how it can make our development work much easier and fun to do. In order to illustrate the usage of Backbone.js I wanted to have proper back-end service. I could have easily used .NET web or REST service for that as my background mainly from Microsoft technologies. But I knew that most web developers are from either Mac or Linux background and they may like to use neutral platform as their back end. I then decided to use PHP. I found very good and easy to understand framework - Laravel (which was fairly new at that time). After going through several pages of Laravel documentation I found that its design is very much similar to ASP.NET MVC and it may help my learning curve of PHP shorter. That's how all these three awesome technologies got together in order to create this tutorial series. So I hope you will enjoy it as I did.

## What is new in this book

Then the obvious question you may ask is what's new or different in this book compared to what is published on the blog. When I was first writing this series in 2012 Laravel has its version 3 if I remember correctly, and since then Laravel has been improved a lot. So I'm going to use latest version of Laravel (which is 4.2 at the time of this writing). Twitter bootstrap has been changed as well so I will be using latest version of bootstrap in the examples. Previously I used Apache web server to host Laravel, recently [NGINX](http://nginx.org/) (<http://nginx.org/>) has gained lot of attention and it is very fast web server compared to Apache2. I used NGINX to host all of my wordpress blogs and Laravel sites without any issue. So I will be using NGINX in this tutorial. Moreover I'm going to re-do the entire coding from scratch so that I can make sure everything working together well. In addition you will get everything in one little booklet that you can refer whenever you needed. Buying this book for a very small price will help me to write more about programming in future as well :).

## What is covered in this book

This will be a small reference book and it will show you how to build simple one page web site using Backbone.js using CoffeeScript with few REST service APIs at the back-end. I will not be covering every

single theory behind all of these in detail as you may be able to find that information on the web. What most new developers require is complete step by step guide on how to use all these technologies in real world application instead of loosely coupled theory from every part, which makes it so confusing to apply it in practical world. So I will teach you everything from, how to to set up your development environment to fully functioning web application using these three technologies.

## These are the topics we will be discussing.

1. Setting up your development environment.  
*I will be using Ubuntu and I assume you already have Ubuntu installed on a PC or VPS or VirtualBox. I will show you how to install NGINX, MariaDB, PHP on Ubuntu*
  2. Configure your project environment.  
*I will explain how to configure virtual directory using NGINX.*
  3. Download and install Laravel. *How to get the latest version of Laravel and install, then how to configure NGINX properly to use Laravel installation.*
  4. Implement simple data model and generate the database.  
*How to use Laravel migrations to create simple database tables and how to use seed functionality to fill sample data.*
  5. Implement REST services.  
*How to implement simple REST service to serve the data model you created and how to set up routes properly.*
  6. Download and install twitter bootstrap for UI.  
*I always like to see nice UI even during the development stage which gives good feeling about what you do, you will also learn some UI tricks every time by doing so.*
  7. Implement blade templates.  
*Blade template system has been changed since the last version I used in previous tutorial series. There are lot of changes and updates to discuss here. I will also explain how I use templates, partial views in proper way that in large real world project it makes your life much easier and produce cleaner code.*
  8. Creating folder structure for CoffeeScripts and JavaScripts output files, then include them in blade templates.
  9. Building Backbone base modes, collections, view.  
*These are set of re-usable classes which can be reused and group common functionality.*
  10. Building Backbone models, collections, views.  
*These are the classes to handle CRUD operations for the domain model we implemented at server side.*
  11. Building HTML templates.  
*We use HTML templates to render different parts of the UI. We break down each single part of the entire UI into smaller manageable views so that we can control them individually. Moreover we can easily re-use these child views.*
  12. Finally Integrate everything to build complete fully functioning UI to handle back end data.
-