Technical design Quizzer

Carlos Kelkboom

July 13, 2014

Abstract

This document is written as a compendium to the Quizzer solution. Most of the technical design decisions and requirements are explained.

Contents

1 Introduction

The quizzer application is an open source project which contains all the necessary application parts to create an extensible testing suit and framework. Separate from this design goal the Quizzer application can also serve as a template for building smart web applications. These concepts are not restricted to building browser based applications.

2 AngularJS

At the root of our application and architecture lies AngularJS[1]. This framework for building web applications and web sites focusses around a MVC type approach to binding your data to your views.

There are many JavaScript MVC frameworks out there and there are more build every day. The reason I chose AngularJS is that when I started writing single page applications I looked for the most flexible system. A system which did not superimpose rules on how to build your application. I find that every time a framework tries to impose a structure on your application you will always run into limitations of this structure. I have been working with AngularJS for over a year and have not found any such restrictions. Ive used AngularJS with an ASP.NET MVC backend with a ruby and rails backend but my favourite remains NodeJS[2].

This is some other text

[1] https://angularjs.org/

[2] NodeJS will be explained in a following chapter

3 Forms

Forms are a big part of the application. Your forms need to be consistent and stylable. The way the Quizzer application solves this problem is by creating AngularJS directives per form field. This way you can guarantee that each control of the same type will be rendered in exactly the same way.

This section will explain the implementation and usage of each of these controls.

3.1 Text

The textbox is the main component and the first well look at within this document.

```
// comment

var foo = function(x, y) {
    return x + y;
}
```

```
<!-- usage -->
<ck-text-m></ck-text-m>
```

4 Conclusion

Write your conclusion here.

Query