Masaryk University
Faculty of Informatics

A Mobile Application for the Administration of the Kentico System

Bachelor's Thesis

Linda Hansliková

Brno, Fall 2016

Declaration

Hereby I declare that this paper is my original authorial work, which I have worked out by my own. All sources, references and literature used or excerpted during elaboration of this work are properly cited and listed in complete reference to the due source.

Linda Hansliková

Advisor: Bruno Rossi, Ph.D

Acknowledgement

My thanks go to Brunno Rossi and Marek Fešar my advisors for allowing me to proceed with this topic and their advice and patience. And also to my friends Matej Pavla, Jozef Vilkolák and Roman Mačor for their support and their help with proofreading this text.

Abstract

In a time where time is more precious than money it is crucial for peole to accomplish a task as quick as possible. When creating various web-sites, the Kentico Enterprise Marketing Solution (KEMS) is a helpful tool to safe time and therefore money. KEMS is a content management system (CMS) which allows clients to create and manage their web-sites using a single user interface (UI). This thesis is about adding an extension to the said system which allows administrators to administrate their site from their smartphones. The functionality implemented should reflect the basic needs of an administrator of the KEMS. The extension consists of two parts: the custom web application programming interface (API) and the mobile application (app). The custom web API was used to call the Kentico API (KAPI) and retrieve data and the mobile app was used as a gateway for the user and the custom web API.

Keywords

Mobile, Mobile Application, Kentico, Javascript, JQuery, WebAPI, Apache Cordova

Contents

List of Figures

1 Introduction

1.1 Introduction

In this thesis we created an extension to the KEMS called KenticoApp. It allows clients to administrate their site from their smartphones. KenticoApp consists of two parts: the custom web API which stores and retrieves data from and to the database and the mobile app which allows the user to communicate with the system.

The custom web API was created using the .Net framework. It uses KAPI calls and is called by the mobile app. For user authentication we decided to use access tokens (AT). ATs are leveraged to secure the communication between a user and the system. After signing in the user is given a random generated AT by the system and the system stores it in it's database. Before every API call the system checks the users AT against the database. For the call to be executed the AT has to exist in the database with the corresponding user ID and must not be expired. If this is not the case the user is redirected to the welcome page, where he has to sign in. To store them in the database we utilized the Entity Framework and implemented data access layer (DAL). For the purpose of this thesis we decided to represent the ATs as an entity using the Entity Framework. The entity contains the user identification (ID) a unique pseudo-random code and an expiration date and time. The code is of the type string and is generated with the pseudo-random generator Random(). Right after generating the code, it is tested against the database if a AT with the same code exists. If yes another code is generated and tested. If there is not the token is assigned the user ID and the date and time 10 minutes from the assignement.

For the implementation of the mobile app we leveraged the Apache Cordva framework (ACF). The reason being that it is simple to use and supports seven platforms. As opposed to the Xamarin framework (XF) supporting three. Even though XF should be faster than ACF the difference between execution times of today's devices is negligible. The development was divided into two stages. For the appereance we decided to use JQuery Mobile. It is an HTML5-based user interface system which allows users to create aesthetically pleasing mobile el-

ements by utilizing the languages Cascading Style Sheets (CSS) and HyperText Markup Language (HTML). As for the functionality we used the JQuery library which has a small learning curve and offers a fast way to add or delete elements or their behaviour.

Chapter one introduces KEMS, web API and hybrid mobile applications. In the second chapter we describe the application architecture and the implementation of the extension of the KEMS in more detail. Finally, we valorise the achieved result and suggest other potential extensions or solutions.

2 Analysis

2.1 Section 1

Example citation[1] [1] *Example italic* Example reference to other section 2.1 Example of escapes $ %

Example paragraph

2.2 Section2

Section 2...

Eample enumerate:

1. item 1
2. item 2
3. item 3

Example desciption:

Term1 description term 1..

Term2 Tdescription term 2..

```
Example code sample
```

This text references image 2.1

2.3 Kentico CMS

2.4 Web Application Interface

2.5 Hybrid Mobile application

1. Example footnote

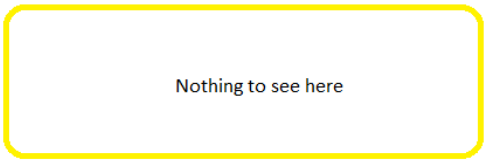

Figure 2.1: This is an empty example image

3 Implementation

3.1 Application Overview

TODO: picture of the architecture of the project and its description

3.2 Extending Kentico

3.2.1 Custom Kentico Module

3.2.2 Kentico 9.0 API

3.3 Web API Application

3.3.1 Microsoft Web API 1.0

TODO:API Controller, Filters, Recieving and sending response (Json), REST: HttpError Codes, stateless, token, not restful

3.4 Cordova Mobile Application

3.4.1 Apache Cordova

TODO: Cordova vs. native vs. Xamarin, PhoneGap, Cordova wrapper

3.4.2 JQuery Mobile

3.4.3 Ajax

TODO: ajax communication with web API

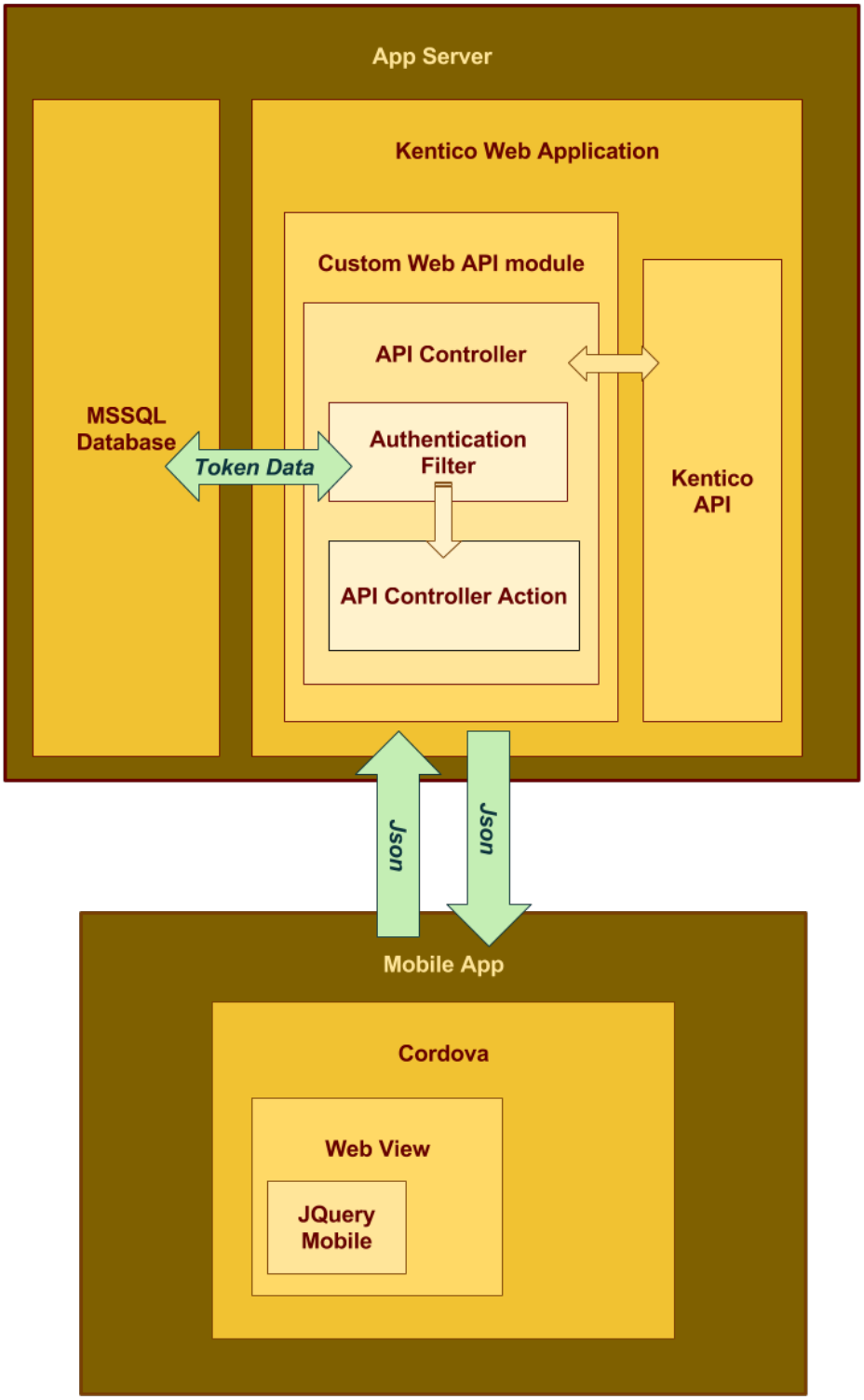

Figure 3.1: Architecture overview

4 Conclusion

4.1 Evaluation

TODO: Functionality

4.2 Future Work

TODO: Ability to choose between availiable sites on Kentico server, Access control, Security Token, Forgotten Password, polished UI

Bibliography

[1] A Great Man. About great men. *Greatness*, 15(9):100–101, 2345.