jamk.fi

# Developing a Web Service

## Databases, Security and Access Control

Markus Moilanen

SUUNNITELMA: https://moilamar.github.io/oppari/plan.xlsx

Bachelor's thesis
Month Year (e.g. September 2015) TODO
Technology, communication and transport
Information and Communication Technology
Degree Programme in Media Engineering

Jyväskylän ammattikorkeakoulu
JAMK University of Applied Sciences

jamk.fi

**Description**

| Author(s) | Type of publication | Date |
|---|---|---|
| Moilanen, Markus | Bachelor's thesis | Month Year TODO |
| | | Language of publication: English |
| | Number of pages TODO | Permission for web publication: x |

Title of publication
**Developing a Web Service**
Databases, Security and Access Control

Degree programme
Media Engineering

Supervisor(s)
Rantala, Ari
Manninen, Pasi

Assigned by
Protacon Solutions Oy

Abstract

Background TODO

The goal of the project assigned by Protacon Solution Oy was to develop work safety familiarization software which includes a front-end web application and a back-end web service. The thesis focuses on some of the most important things to consider when developing a web service.

The web service was built using the PHP framework Symfony. TODO: Doctrine, MySQL, etc.

Results TODO

Conclusions TODO

Keywords/tags (subjects)
Web Service, Database, Security, Access Control, Software, PHP, Symfony, Back-end, REST

Miscellaneous

jamk.fi

**Description**

| | | |
|---|---|---|
| Tekijä(t)<br>Moilanen, Markus | Julkaisun laji<br>Opinnäytetyö, AMK | Päivämäärä<br>Month Year TODO |
| | | Language of publication:<br>Englanti |
| | Sivumäärä<br>TODO | Verkkojulkaisulupa myönnetty: x |
| Työn nimi<br>**Developing a Web Service**<br>Databases, Security and Access Control | | |
| Tutkinto-ohjelma<br>Mediatekniikka | | |
| Työn ohjaaja(t)<br>Rantala, Ari<br>Manninen, Pasi | | |
| Toimeksiantaja(t)<br>Protacon Solutions Oy | | |


Tiivistelmä

When completing this form, start from this field, on the row under the instructions, so that the font size remains 11.

The basic structure of the abstract is as follows:

- background
- task and objectives
- implementation method
- results
- conclusions.

In other words, the abstract summarises the work that has been done – not the content of the report. If there is room, the content of the report may be briefly mentioned.

The entire space reserved for the abstract must be used.

The abstract should be written in the past tense and with a passive voice. The text must not refer to the thesis, i.e., the words 'this thesis' must not be used.

TODO



Avainsanat (avainsanat)
Web-palvelu, tietokanta, turvallisuus, oikeuksien hallinta, ohjelmisto, PHP, Symfony, Back-end, REST

Muut tiedot

## Contents

**Teminology** ..... **5**

**1 Introduction** ..... **7**

1.1 Background ..... 7

1.2 Protacon Solution Oy ..... 7

1.3 Objective of the Thesis ..... 8

1.4 Project Premise ..... 8

**2 Web Services** ..... **9**

2.1 What Are Web Services? ..... 9

2.2 Development Frameworks ..... 9

2.2.1 Symfony ..... 9

2.2.2 Laravel ..... 9

2.2.3 NodeJS Frameworks ..... 9

2.2.4 .NET Framework ..... 9

2.2.5 Java Frameworks ..... 9

2.3 Design Models ..... 9

2.3.1 SOAP ..... 9

2.3.2 REST ..... 9

2.3.3 RPC ..... 9

2.4 Authentication ..... 9

2.4.1 Basic Authentication ..... 9

2.4.2 OAuth ..... 9

2.4.3 API Keys ..... 9

2.5 Best Practices ..... 9

2.6 Databases ..... 9

2.6.1 Relational Databases ..... 9

2.6.2 Non-relational Databases ..... 10

**3 Requirements and Development Choices ..... 10**

3.1 Requirement Specification ..... 10

3.1.1 Entity Relations ..... 10

3.1.2 User Roles and Rights ..... 10

3.1.3 Access Right Requirements ..... 10

3.1.4 Other Requirements ..... 10

3.2 Technologies ..... 10

3.2.1 Back-end Technologies ..... 10

3.2.2 Front-end Technologies ..... 10

3.3 Development Environment and Tools ..... 10

**4 Database Design ..... 10**

4.1 Schema ..... 10

4.1.1 Entities ..... 10

4.1.2 Entity Relations ..... 10

4.2 Accessing Data ..... 10

4.2.1 Symfony Forms ..... 10

4.2.2 Doctrine ORM ..... 10

4.2.3 Lifecycle Hooks ..... 11

**5 Security ..... 11**

5.1 Authentication ..... 11

5.1.1 JSON Web Tokens ..... 11

5.1.2 Logging In ..... 11

5.1.3 File Downloads Using API Keys ..... 11

5.2 Firewalls and Guards ..... 11

5.3 Passwords ..... 11

5.3.1 Encryption ..... 11

5.3.2 Password Generation ..... 11

**6 User Roles and Access Control ..... 11**

6.1 Importance of Access Control ..... 11

6.2 Access Control List ..... 11

6.3 User Roles and Groups ..... 11

6.4 Access Control Entries ..... 11

6.4.1 Entity Design ..... 11

6.4.2 Creating Entries ..... 11

6.4.3 Propagating Rights ..... 11

6.4.4 Inheritance ..... 12

6.4.5 Removing Rights ..... 12

6.5 Post-Production Changes ..... 12

**7 Other Issues ..... 12**

7.1 Performance ..... 12

7.1.1 Debouncing ..... 12

7.1.2 Reducing Database Queries ..... 12

7.2 Client-Specific Features ..... 12

7.3 Post-Production development ..... 12

7.4 Scalability ..... 12

**8 Conclusions ..... 12**

8.1 End Product ..... 12

8.2 Flaws ..... 12

8.3 Improvement Possibilities ..... 12

**References** ..... **13**

**Appendices** ..... **14**

Appendix 1. Ammattikorkeakoulutuksen aloittaneiden läpäisy ..... 14

**Figures**

No table of figures entries found.

**Tables**

**No table of figures entries found.**

# Teminology

**API (Application Programming Interface)**

An interface that allows different programs or machines to communicate with each other and send data, for instance an endpoint on a server.

**Git**

TODO

**HTTP**

TODO

**MVC**

TODO

**Framework**

TODO

**Web server**

TODO

# 1 Introduction

## 1.1 Background

The world of a software developer is one of ever-changing variables. Even when the project requirements are as clear as the developers' knowledge is vast, there are times when one has to stop to think. While working on a software project, one may think that they could have chosen more appropriate technologies for it, or they simply find a particular solution to be insufficient. Frankly, a lot of it comes down to lack of experience in a specific area and can be solved by proper planning. That's when the understanding of different technologies and techniques comes to play.

When developing a web service, there are several important things to consider. First of all, there are multiple types of databases for different purposes as well as techniques to use them. TODO On top of that, there are various ways to handle security and user rights including authenticators, firewalls, user roles and access control. TODO

## 1.2 Project Premise

The goal of the project was to a develop work safety familiarization software which includes a front-end web application and a back-end web service. In a nutshell, the application should allow client companies to create "courses" which are essentially quizzes used to test their employees' knowledge of various topics. The system has to support large organizations that have subsidiaries and subcontractors with a substantial amount of employees. Not only that, but there have to be different user roles and rights for each role, with a single user being allowed to be a part of multiple companies with multiple different roles. A more detailed description of the requirements is specified in chapter TODO.

TODO

## 1.3 Protacon Solutions Ltd

The project was assigned by Protacon Solutions Ltd, a subsidiary of Protacon Group. Protacon Solutions is a software development company based in Jyväskylä, Finland with multiple offices around the country. It focuses on digitalization and software development including creating web applications and services as well as mobile applications. The project in question, accronymized TyPe, was one of Protacon Solutions' many original systems.

The project was developed by a small team of software developers employed by Protacon Solutions at their office in Jyväskylä. The development started in early 2018 and was still in progress at the end of the year.

## 1.4 Objective of the Thesis

Naturally, the main goal of the project was to create a functioning system that fills the requirements set by Protacon Solutions Ltd and its clients. TODO It was difficult at times due to the complex nature of the requirement specifications as well as lack of resources in general.

In addition, the thesis aims to shed light on some of the design choices made by the developers and to spread knowledge about the associated techniques and technologies. TODO

The thesis also served as a method of self-learning. Learning about different development options is highly useful since there is not enough time to study them deeply in normal day-to-day work. Not only that, but there are countless critical aspects of server-side programming that are often ignored or hastily presented in school curriculums resulting in low proficiency and experience when it comes to using those tools in a workplace environment.

# 2 Web Services

## 2.1 What Are Web Services?

## 2.2 Development Frameworks

### 2.2.1 Symfony

### 2.2.2 Laravel

### 2.2.3 NodeJS Frameworks

### 2.2.4 .NET Framework

### 2.2.5 Java Frameworks

## 2.3 Design Models

### 2.3.1 SOAP

### 2.3.2 REST

### 2.3.3 RPC

## 2.4 Authentication

### 2.4.1 Basic Authentication

### 2.4.2 OAuth

### 2.4.3 API Keys

## 2.5 Best Practices

## 2.6 Databases

### 2.6.1 Relational Databases

2.6.2 Non-relational Databases

# 3 Requirements and Development Choices

## 3.1 Requirement Specification

### 3.1.1 Entity Relations

### 3.1.2 User Roles and Rights

### 3.1.3 Access Right Requirements

### 3.1.4 Other Requirements

## 3.2 Technologies

### 3.2.1 Back-end Technologies

### 3.2.2 Front-end Technologies

## 3.3 Development Environment and Tools

# 4 Database Design

## 4.1 Schema

### 4.1.1 Entities

### 4.1.2 Entity Relations

## 4.2 Accessing Data

### 4.2.1 Symfony Forms

### 4.2.2 Doctrine ORM

4.2.3 Lifecycle Hooks

# 5 Security

## 5.1 Authentication

### 5.1.1 JSON Web Tokens

### 5.1.2 Logging In

### 5.1.3 File Downloads Using API Keys

## 5.2 Firewalls and Guards

## 5.3 Passwords

### 5.3.1 Encryption

### 5.3.2 Password Generation

# 6 User Roles and Access Control

## 6.1 Importance of Access Control

## 6.2 Access Control List

## 6.3 User Roles and Groups

## 6.4 Access Control Entries

### 6.4.1 Entity Design

### 6.4.2 Creating Entries

### 6.4.3 Propagating Rights

6.4.4 Inheritance

6.4.5 Removing Rights

6.5 Post-Production Changes

**7 Other Issues**

7.1 Performance

7.1.1 Debouncing

7.1.2 Reducing Database Queries

7.2 Client-Specific Features

7.3 Post-Production development

7.4 Scalability

**8 Conclusions**

8.1 End Product

8.2 Flaws

8.3 Improvement Possibilities

# References

Add the references used in your thesis in alphabetical order here, all in one list. Use markings in accordance with the reporting instructions.

Alasuutari, P. 1999. *Laadullinen tutkimus [Qualitative research]*. 3rd ed. Tampere: Vastapaino.

Hakala, J. T. 2004*. Opinnäyteopas ammattikorkeakouluille [Thesis guide for universities of applied sciences]*. Helsinki: Gaudeamus.

Hendriks, R. 2013. *Perceptions of facility management in Europe: A survey of Finland, Germany and the UK* (Bachelor's thesis). JAMK University of Applied Sciences, School of Business and Services Management, Degree Programme in Facility Management.

Hirsjärvi, S., Remes, P., & Sajavaara, P. 2009. *Tutki ja kirjoita [Research and write]*. 15th ed. Helsinki: Tammi.

X

# Appendices

Appendix 1. Ammattikorkeakoulutuksen aloittaneiden läpäisy

Here you can place, for example, a table that does not fit in naturally with the text.

Be sure to refer to all appendices in the text.