

# CAKEPHP & EXTJS - RESPONSIVE WEB TECHNOLOGIES

Davor Lozić, Alen Šimec

Tehničko veleučilište u Zagrebu

## Sažetak

Ovaj rad prikazuje današnje, moderne tehnologije za responzivni web. Prikazuje način na koji ove različite tehnologije komuniciraju na webu i potrebne uvjete za siguran i brz rad ovih tehnologija. Članak detaljno opisuje rad serverskih alata kao što je CakePHP i klijentkih alata kao što je ExtJS. Pojmovi kao što su MVC, JSON, AJAX komunikacija, HTTP zahtjevi i odgovori su također objašnjeni. Rad detaljno opisuje i proces od otvaranja stranice na klijentskoj strani do stvaranja odgovora na strani servera.

**Ključne riječi:** *Internet, responzivni web, MVC tehnologija, server-klijent komunikacija*

## Abstract

This paper presents descriptions of today's modern technologies for responsive web applications. It presents the way those different technologies communicate on the web and what are the dependencies for a secure and fast work of these technologies. In applied research, concepts for server-side tools like CakePHP and client-side tools like ExtJS are explained in depth. Terms like MVC, JSON, AJAX communication, HTTP request and response are also explained. This paper also shows the complete proces from opening the page on client's side and creating the response on the server's side.

**Key-Words:** *Internet, responsive web, MVC technology, server-client communication*

## 1 INTRODUCTION

In today's world of the Internet one can literally do everything. One can surf the web, use social networks, read the news or just buy anything one wants. The question I would like to answer is what technology is behind this entire grid of information. How is it even possible to have the entire web application

somewhere in the world and use it in your home?

First of all, there are two things. One's own computer, connected to the Internet, of course. On the "other side of the line", there is a server – computer specialized for many things and one of them is giving answers to one's questions, specialized questions. That 'specialization' is called protocol and the most famous of all is TCP-IP and the whole Internet depends on it.

## 2 MVC TECHNOLOGIES

MVC (model-view-controller) is currently one of the best programming techniques. In a simple web application, inside of a file one can have lots of different technologies: HTML, CSS, JavaScript, PHP etc. but in MVC programming model, one divides those technologies and creates some conventions.

### 2.1 MVC LAYERS

Model layer – it's the M in MVC and it's the biggest layer. Whenever working with data (base) one works with Model. Connection to database, retrieving, saving and changing data, it's all part of this layer.

View layer – also called a "presentation layer". Here you include HTML, CSS, JavaScript, headers and footers, anything one wants to show to user.

Controller layer – it's the "connecting" layer. Everything that doesn't belong in Model or View, belongs here. Main purpose of this layer is connecting the data from model to the presentation layer – View.

### 2.2 MVC REQUEST

How does a simple request look like in MVC? [2]

1. Browser makes a request to:  
www.application.com/users/read.

2. Controller gets the request and sees that one needs *user* controller and the action *read* inside of it.
3. Inside the *read* action called the Model.
4. User's Model gets the data one needs and sends it back to the Controller.
5. Controller gets the data and prepares it for the View.
6. View gets the data from Controller and sends the HTML to the user's browser.

Almost every request goes through these steps. Sometimes there is no data needed so the steps 3 and 4 are not necessary.

## 2.3 CAKEPHP

CakePHP is one of the most famous PHP frameworks. PHP is a server-side language and CakePHP is the MVC framework for PHP. Of course, every framework has some of the advantages and disadvantages.

Advantages [5]:

- Simple configuration
- Open Source, OOP
- convention over configuration
- bake scripts
- friendly URLs
- rich query API

Disadvantages:

- documentation needs some work
- learning curve
- leak of understanding the Cake background

CakePHP really makes coding easier. Earlier, if you've created five web applications, you've wrote instructions for log in and user registration five times. CakePHP has implemented those actions into Components and all you have to do is create the configuration array inside the component and the framework will do the work for you.

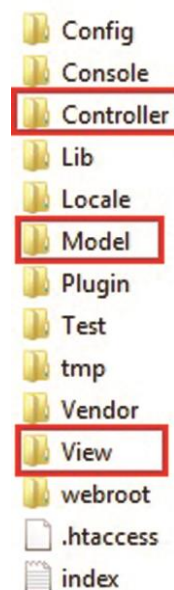
### 2.3.1 CAKEPHP CONCEPTS

CakePHP has several concepts and rules. DRY, convention over configuration, special folder structure, and scaffolding are some of them.

DRY – don't repeat yourself – never duplicate the code. When wanting to share code, always use Components which can be included in any Controller or put it inside of ApplicationController, parent of all Controllers. Duplicated code is hard to maintain, change and creates unnecessary work.

Convention over configuration – the real CakePHP “magic” happens when one uses Cake's convention rules. Table names inside the database are always plural, Models are singular, Controllers are plural concatenated with “Controller” word and Views are plural and inside the folder named by Controller. When using this convention, CakePHP connects Model, Controller and the View and does the whole request. If this convention is not followed, details inside those modules need to be specified.

Special folder structure – all files inside CakePHP have a place where they need to be.



Picture 1 special folder structure

As presented, CakePHP has a really clear structure and you always know where to put source code.

Scaffolding – is a technique supported by most MVC frameworks in which the programmer may write some specifications that describe how the application database may be used and in this case CakePHP, internally generates the code with which one can check his/her application “on-the-fly”.

```

{
  "type": "object",
  "properties": {
    "a": {"type": "string"},
    "b" : {"type": "integer"},
    "c" : {"type": "boolean"}
  }
}

```

Picture 2 CakePHP Request [5]

## 2.4 EXTJS

ExtJS is Sencha's solution for a MVC JavaScript library. It's great for creating large and rich web applications. [3] With ExtJS there are a lot of features: textfields, textareas, listboxes and comboboxes, grids, trees, sliders etc. It's not about the UI, it's about the functionality behind this huge framework such as Ajax calls, DOM scripting etc.

Advantages [4]:

- working with data (pagination, filtration, sorting)
- simple grids
- CSS out-of-the-box
- consistent look in the whole application
- Sencha tools for packaging
- great documentation

Disadvantages:

- learning curve
- size of the library
- CSS – very easy to get lost
- hard debugging
- not free for commercial software

ExtJS framework makes client-side programming easier, especially web design. Concept of programming in this framework looks like a big JavaScript object full of configuration. One assigns the configuration object and the framework renders the rest. Just like CakePHP, ExtJS has DRY concepts, conventions and special folder structure. [7] Great benefit of using this framework is also portability; there is no need to write one version of code for Internet Explorer and the other version for every other browser. MVC model in this framework is implemented really sophisticated. The example is Store in ExtJS. Store is a collection of Models, collection of data. Then, connected to the grid, View, to the

store and whatever changes inside the Store it is automatically visible inside the Grid.

## 3 SERVER-CLIENT AJAX COMMUNICATION

AJAX stands for Asynchronous JavaScript and XML. It's a development technique for getting the data from the server without refreshing the whole page. Looking at Facebook chat system, eBay “on mouse hover” pictures etc. it is apparent that these are “desktop-like” application.

So, one creates a request, s data and waits for the server's response. The data being sent is JSON (JavaScript Object Notation) or XML (eXtensible Markup Language). Both technologies are just standardized text with their own beneficial. [6]

XML is used when sending some configuration or simple data and JSON when sending much more data. In this case, reading and writing is done inside the grid, thus using JSON.

```

{
  "type": "object",
  "properties": {
    "a": {"type": "string"},
    "b" : {"type": "integer"},
    "c" : {"type": "boolean"}
  }
}

```

Picture 3 JSON example

## 4 CONCLUSION

Although these technologies are relatively new, nowadays, many applications use MVC and AJAX. Users want “desktop-like” applications and they get that with those technologies which are, at least, free for open-source projects. Also, there is a large community support and one can get help and response on many forums in a few hours, sometimes minutes. Another advantage, just looking at the source code of these open source projects, one can learn a lot and that's what all is about for many great developers.

## 5 REFERENCES:

- 1] David Golding; *Beginning CakePHP: From Novice to Professional*, Apress, 2008.
- 2] Mariano Iglesias; *CakePHP 1.3 Application Development*; Packt Publishing Ltd.; 2011.
- 3] Crystel Villa; Armando Gonzalez; *Learning Ext JS 4.*; Packt Publishing Ltd.; 2013.
- 4] Ahsanul Bari; *CakePHP Application Development: Step-by-step introduction to rapid web development using the open-source MVC CakePHP framework*; Packt Publishing Ltd.; 2008.
- 5] Bartosz Porebski; Karol Przystalski; Leszek Nowak; *Building PHP Applications with Symfony, CakePHP, and Zend Framework*; Wiley Publishing; 2011.
- 6] Jesus Garcia; *Ext JS in Action*; Manning Publication; 2011.
- 7] Nicholas C. Zakas; *Professional JavaScript for Web Developers*; John Wiley & Sons, Inc.; 2012.

**Davor Lozić** (dlozic@tvz.hr), student at the Polytechnic of Zagreb, currently working as a lead software developer at Informatička podrška d.o.o. creating web-based applications with ExtJS and CakePHP. Interested in web-application security, internal work of operating systems and deeper knowledge of DBMS's, transactions and query optimizations



**Dr. sc. Alen Šimec** (alen@tvz.hr), was born January 19th 1979. in Zagreb and graduated from professional undergraduate studies at Polytechnic of Zagreb in 2002 where he earned the professional title of Information Engineer. He also graduated from professional undergraduate studies at College of Security in 2006 where he earned the professional title of Security Engineer. In 2008 he graduated from the Professional Study Polytechnics Program, Department of Computer Science. Also in 2008, he received his Ph. D. from the University of Zagreb, Department of Information and Communication Sciences.