CAKEPHP & EXTJS - RESPONSIVE WEB TECHNOLOGIES

Davor Lozić, Alen Šimec

Tehničko veleučilište u Zagrebu

Abstract

This paper presents descriptions of today’s modern technologies for a responsive web applications. It presents the way how those different technologies communicates on the web and what are the dependencies for a secure and fast work of this 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

Today in the world of the Internet, you can literally do everything. You can surf the web, be on the social networks, read the news or just buy anything you want. The question which I would try to answer is, what is the technology behind all of this 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, you have two things. You have your own home computer, connected to the Internet, of course. At the “other side of the line”, there is a server – computer specialized for many things and one of them is giving answers to your 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 technique. In a simple web application, inside of a file you can have a lots of different technologies: HTML, CSS, JavaScript, PHP etc. But in MVC programming model, you divide those technologies and create some conventions.

2.1 MVC LAYERS

Model layer – it's the M in MVC and it's the biggest layer. Whenever you have to work with data(base) you work with the 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 your HTML, CSS, JavaScript, headers, footers, anything you want to show to user.

Controller layer – it's the “connecting” layer. Everything which doesn't belong in Model or View, belongs here. His main purpose is connecting the data from model to the presentation layer – View.

2.2 MVC REQUEST

How does a simple request look like in MVC?

1. Browser makes a request to:
   www.application.com/users/read.
2. Controller gets the request and sees that you need users controller and the action read inside of it.
3. Inside the read action you are calling the Model.
4. User's Model gets the data you need 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 you don't need any data 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 your code. When you want to share code, always use Components which you can include in any Controller or put it inside of AppController, parent of all Controllers. Duplicated code is hard to maintain, change and creates unnecessary work.

Convention over configuration – you see the real CakePHP “magic” when you use 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 you are using this convention, CakePHP connects Model, Controller and the View and does the whole request for you. If you don’t follow this convention, you need to specify details inside those modules.

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

Picture 1 special folder structure

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

Scaffolding – is a technique supported by most MVC frameworks in which the programmer may write some specifications that describes how the application database may be used and, in this case CakePHP, internally generates the code for you and you can check your application “on-the-fly”.
2.4 EXTJS

ExtJS is Sencha's solution for a MVC JavaScript library. It's great for creating huge and rich web applications. With ExtJS you have 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, specially web design. Concept of programming in this framework looks like a big JavaScript object full of configuration. You just put the configuration object and the framework renders everything for you. Just like CakePHP, ExtJS has DRY concepts, conventions and special folder structure. Great benefit of using this framework is also a portability, you don’t have to write one version of code for Internet Explorer and the second 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, you connect the grid, View, to the store and whatever you change inside the Store it will automatically be visible inside the Grid.

3 SERVER-CLIENT AJAX COMMUNICATION

AJAX stands for Asynchronous JavaScript and XML. It's a development technique where you get the data from the server without refreshing the whole page. If you look Facebook's chat system, eBay's “on mouse hover” pictures etc. you will see that you have a “desktop-like” application.

So, you create a request, send data and wait 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.

You use XML when you send some configuration or simple data and JSON when you send much more data. In this case, we are reading and writing to a grid so here we are using JSON.

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

4 CONCLUSION

Although these technologies are relatively new, today, 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. Today, community support is really big and you 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, you can learn a lot and that’s what’s all about for many great developers.
5 REFERENCES:


2] Mariano Iglesias; CakePHP 1.3 Application Development; Packt Publishing Ltd.; 2011.


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.
