

**Bachelor Project**



**Czech  
Technical  
University  
in Prague**

**F3**

**Faculty of Electrical Engineering  
Department of Computer Science**

## **Creating a plugin for managing partners on the web**

**Margarita Gorbatenko**

**Supervisor: Ing. Tomáš Vondra, Ph.D.  
Field of study: Software Engineering and Technology  
May 2023**



## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Gorbatenko** Jméno: **Margarita** Osobní číslo: **495570**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávající katedra/ústav: **Katedra počítačů**  
Studijní program: **Softwarové inženýrství a technologie**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**Tvorba pluginu pro správu partnerů na webu**

Název bakalářské práce anglicky:

**Creating a plugin for managing partners on the web**

Pokyny pro vypracování:

Czech-American TV (CATVUSA) is a non-profit TV promoting the Czech Republic in the USA. They need help editing their website to support marketing/fundraising campaigns.

1. Familiarize yourself with the Wordpress CMS and creating plugins for it.
2. In cooperation with the TV producer, collect requirements for the administration and displaying of partners on the website.
3. Implement a plugin that will manage all spaces for the promotion of partner organizations.
4. Design and implement a new API for this plugin as well as the older radio station plugin, so that their content can be displayed on mobile and Smart TV applications.
5. Properly test the plugin and API using code validators and perform user testing of the functionality of both parts.
6. Create documentation and a user manual for the created work.

Seznam doporučené literatury:

1. WordPress Plugins [online]. WordPress.org. [Cit. 2021-01-25]. Dostupné z: <https://wordpress.org/plugins/>.
2. Plugin Developer Handbook [online]. WordPress.org. [Cit. 2021-03-02]. Dostupné z: <https://developer.wordpress.org/plugins/>.
3. MESSENLEHNER, B. a COLEMAN, J. Building Web Apps with Word-Press, 2nd edition. Sebastopol, CA: O-Reilly Media, Inc., 2019. 546 s. ISBN 9781491990087.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

**Ing. Tomáš Vondra, Ph.D. katedra počítačových systémů FIT**

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **14.02.2023**

Termín odevzdání bakalářské práce: **26.05.2023**

Platnost zadání bakalářské práce: **22.09.2024**

Ing. Tomáš Vondra, Ph.D.  
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.  
podpis děkana(ky)

## III. PŘEVZETÍ ZADÁNÍ

Studentka bere na vědomí, že je povinna vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studentky



## Acknowledgements

I would like to thank Ing. Tomas Vondra, Ph.D. for his professional advises and John Honner for his time and help. I would also like to thank my family and friends for their support.

## Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended. In accordance with §2373 paragraph 2 of Act No. 89/2012 Coll., Civil Code, as amended, I hereby grant a non-exclusive authorization (license) to use this work of mine, including all computer programs that are part or appendices thereof and all their documentation (hereafter referred to as the "Work"), to all persons who wish to use the Work. Such persons are entitled to use the Work in any manner that does not diminish the value of the Work and for any purpose (including use for profit). This authorization is unlimited in terms of time, territory and quantity.

In Prague, May 26, 2023

## Abstract

Czech-American TV organisation needs help with editing its website to support marketing/fundraising campaigns. This project presents an analysis and solution of the problems found.

**Keywords:** plugin, requirements analysis, PHP, WordPress, Czech-American TV

**Supervisor:** Ing. Tomáš Vondra, Ph.D.

## Abstrakt

Organizace Czech-American TV potřebuje pomoc s úpravou svých webových stránek pro podporu marketingových/fundraisingových kampaní. Tento projekt představuje analýzu a řešení nalezených problémů.

**Klíčová slova:** plugin, analýza požadavků, PHP, Wordpress, Czech-American TV

**Překlad názvu:** Tvorba pluginu pro správu partnerů na webu

# Contents

<b>1 Introduction</b>	<b>1</b>	5.2.2 Partner Banner Administration API.....	34
<b>2 Theory</b>	<b>3</b>	5.2.3 Endpoints.....	35
2.1 HTML, JavaScript, CSS .....	3	<b>6 Testing</b>	<b>39</b>
2.2 PHP .....	3	6.1 Testing code with a validation service .....	39
2.3 Programming and testing tools ..	4	6.2 User Testing .....	39
2.3.1 XAMPP .....	4	6.2.1 Pre-testing questions .....	40
2.3.2 PhpStorm.....	4	6.2.2 Test scenario .....	40
2.4 WordPress .....	4	6.2.3 Feedback .....	41
2.4.1 Post Type .....	5	6.2.4 Result .....	42
2.4.2 Taxonomy .....	5	<b>7 Conclusion</b>	<b>43</b>
2.4.3 Plugins .....	5	7.1 Conclusion.....	43
2.4.4 Themes .....	5	7.1.1 Partner Banner Administration plugin.....	43
2.4.5 Hook .....	6	7.1.2 API for custom plugins .....	44
2.5 API.....	7	<b>Bibliography</b>	<b>47</b>
2.5.1 HTTP Statuses .....	7	<b>Attachment</b>	<b>51</b>
2.5.2 REST API.....	7	1.1 Partner Banner Administration User Manual .....	51
2.5.3 WordPress REST API.....	7	1.1.1 Installation and launch .....	51
<b>3 Analysis</b>	<b>9</b>	1.1.2 Administration Page .....	53
3.1 Goals of the project .....	9	1.2 Extended API User Manual .....	59
3.2 Sponsor .....	9	1.2.1 Installation and launch .....	59
3.3 Current state.....	10	1.2.2 Endpoints.....	60
3.3.1 Website.....	10		
3.3.2 Partners banners.....	11		
3.3.3 Radio manager plugin .....	15		
3.3.4 AndroidTV application .....	18		
3.4 Desired outcome.....	19		
3.4.1 Customer requirements for administration of partners banners	19		
3.4.2 Customer requirements for applications-plugins communication problem .....	20		
<b>4 Suggested solution</b>	<b>21</b>		
4.1 Administration of partners banners.....	21		
4.2 Communication between custom plugins and applications .....	22		
4.2.1 Description and reasoning of the proposed solution .....	22		
<b>5 Implementation</b>	<b>23</b>		
5.1 Partner Banner Administration .	23		
5.1.1 Frontend.....	23		
5.1.2 Backend .....	28		
5.2 API plugin.....	32		
5.2.1 Radio Manager API .....	33		

## Figures

2.1 Most popular content management systems. Source: w3techs.com . . . . .	4
3.1 Czech-American TV logo . . . . .	10
3.2 Homepage . . . . .	10
3.3 Banners with partners . . . . .	11
3.4 Broadcast Partners on Homepage	12
3.5 Custom Sidebars Administration user interface . . . . .	12
3.6 A sidebar that contains partner banners . . . . .	13
3.7 HTML sctructure of a page. A div element with a sidebar for partners banners and a div with the content of the page . . . . .	13
3.8 HTML structure of banner's container . . . . .	14
3.9 Administration is not user-friendly . . . . .	14
3.10 Audio program on the website .	15
3.11 Radio station's playlist settings	16
3.12 getData() function in Radio Manager plugin . . . . .	17
3.13 AndroidTV application . . . . .	18
3.14 Player in TV application . . . . .	18
3.15 Hard coded names of radio stations: "cultural", "classical", "folk"	19
5.1 Administration of Partner's banners. . . . .	24
5.2 Request to the server to get data about all banners to display them in the table. . . . .	24
5.3 The process of creation of the each row in the table. . . . .	24
5.4 Popup with form to add a new banner . . . . .	25
5.5 End of popup with form to add a new banner . . . . .	25
5.6 Categories in form . . . . .	26
5.7 The process of getting list of pages from server. . . . .	26
5.8 The process of creating a list of banner's checked categories. . . . .	26
5.9 Warnings in form. . . . .	27
5.10 Warnings in form. . . . .	27
5.11 Popup with form to edit a banner. . . . .	27
5.12 Confirmation window to delete a banner. . . . .	28
5.13 Banner's preview. . . . .	28
5.14 The hook that adds the main page of the plugin. . . . .	29
5.15 Activation and deactivation hooks. . . . .	29
5.16 Inserting Logo to the WP media library . . . . .	29
5.17 Inserting Post to the WP database. . . . .	30
5.18 Getting attachment URL. . . . .	30
5.19 Saving Post metadata to the WP database. . . . .	31
5.20 The process of creating HTML of the banner . . . . .	32
5.21 Registration of scheduled event.	32
5.22 The process of registration of the REST route. . . . .	33
5.23 The process of getting data from Radio Manager plugin. . . . .	34
5.24 Get all data (Postman). . . . .	35
5.25 Get the data for a TV player (Postman). . . . .	36
5.26 Get playlist data (Postman). . .	36
5.27 Get warning data (Postman). . .	37
5.28 Get settings data (Postman). . .	37
5.29 Get banners' data (Postman). . .	38
6.1 Markup validation result. . . . .	39



## Tables







# Chapter 1

## Introduction

In this chapter I am introducing the reader to the issues that this thesis is dealing with.

Czech-American TV is a non-profit organization that promotes culture and traditions of Czech Republic in the USA. Czech-American TV has a website (catvusa.com [1]), which provides educational programs about the Czech Republic, articles about traditions, language lessons, radio stations.

Czech-American TV has an AndroidTV application with videos, radio stations and podcasts. Application was written by a CTU student Ing. Viktor Sinelnikov [2]. Also, there is another Smart TV application created by CTU student Bc. Matúš Višcor as his Master's thesis (Smart TV application development for a nonprofit TV).

Organisation uses WordPress and plugins written by students to manage its website. The WordPress content management system is well-known for its simplicity, compatibility and adaptability on computers and mobile devices. It is a tool that allows user to create web applications and easily manage websites [3]. Currently, the TV application has problems with the plugins functionality. Also, there are some things to be improved in website's functionality. This project presents an analysis and solution to these problems.

Goals of this thesis are:

- Get to know CMS WordPress and the process of creating plugins
- To analyse current state of the website and application, get the details about requirements from the sponsor (Czech-American TV)
- To propose the administration for managing the partners' promotions on the website and applications
- Implement a plugin that will ensure the management of the promotion of partner organizations
- Design and implement a new API for this plugin as well as for older plugins (Radio Station)
- To test the result with validators and user testing
- To create a documentation and a user manual

The **theoretical part** [2] of the project includes all needed theory about technologies such as WordPress, API and other.

**Analysis part** [3] contains information about sponsor. It also describes current status and desired outcome of web pages and application.

**Solution part** [4] contains proposed solution, based on customer requirements.

**Implementation part** [5] includes description of the implementation and the outcome, based on proposed solution.

**Testing part** [6] describes the process and results of code tests and user testing.

And, in the end, **Conclusion part** [7] summarizes the work done and compares the final result with the initial requirements.

## Chapter 2

### Theory

In this chapter I describe technologies that will be required for deployment of solution.

#### 2.1 HTML, JavaScript, CSS

HTML, JavaScript and CSS are the three main technologies used to create interactive and stylish web pages. They work on the client side, also known as the frontend. They run directly in the user's browser and are responsible for displaying and interacting with the web page.

**HTML** (stands for HyperText Markup Language) is a markup language used to define the structure and content of a web page. It is the most basic building block of the Web [4].

HTML5 is the latest version of the HTML standard. It is a set of web technologies, which are the HTML Living Standard, along with JavaScript APIs to enhance storage, multimedia, and hardware access [5].

**JavaScript** is one of the most popular interpreted programming language that runs on the client side (web browser) and is widely used to create dynamic web pages, web applications, games, mobile applications, etc. [6] Generally, JavaScript is used to add interactivity and functionality based on events and user actions. JavaScript can change the content of the web page and structure of an HTML document using the DOM (Document Object Model). It can change the content dynamically, without the need to reload the page. There are many JavaScript libraries and frameworks such as React, Angular, jQuery, and others that make it easy to develop web applications and provide many ready-made solutions and tools.

**CSS** (Cascading Style Sheets) is a style language that is used to define the appearance of web pages [7]. It defines the style of HTML elements.

#### 2.2 PHP

PHP is a scripting programming language that is widely used for developing web applications and dynamic websites [8]. PHP code is executed on the

server (backend). The server processes requests from clients and generates HTML pages that are sent back to the client side for display [9].

## 2.3 Programming and testing tools

### 2.3.1 XAMPP

XAMPP (Cross-Platform, Apache, MySQL, PHP, and Perl) is a software that provides an environment for developing, running and testing web applications on a local server. It includes Apache web server, MySQL database management system, PHP interpreter and other tools [10].

### 2.3.2 PhpStorm

PhpStorm is an integrated development environment (IDE) by JetBrains [11]. It allows to develop and debug applications in PHP, HTML, CSS, JavaScript, and other web development languages. PhpStorm also has version control systems, such as Git, built-in code analysis, web development frameworks and tools, remote development and other features that allow developers to code, test and debug web applications efficiently.

## 2.4 WordPress

WordPress is free open-source Content Manager System (CMS) written in hypertext preprocessor language (PHP) and MySQL [3]. WordPress is used to create websites, blogs or other applications. According to official statistics, WordPress is used on 43 percent of websites worldwide, surpassing open source CMS such as Joomla or Drupal [12].

© W3Techs.com	usage	change since 1 April 2023	market share	change since 1 April 2023
1. <a href="#">WordPress</a>	43.1%	-0.1%	63.3%	
2. <a href="#">Shopify</a>	3.8%		5.6%	+0.1%
3. <a href="#">Wix</a>	2.5%		3.7%	
4. <a href="#">Squarespace</a>	2.1%		3.1%	
5. <a href="#">Joomla</a>	1.8%		2.7%	

percentages of sites

**Figure 2.1:** Most popular content management systems. Source: w3techs.com

Built-in system of plugins allows to construct projects with wide functionality. The core structure of WordPress website is similar to a set of building blocks. The page content is located in main containers: header, sidebar/menu, content, footer, and other optional elements. In the WP root directory there are three core folders: wp-content, wp-includes, and wp-admin. [13]

### ■ 2.4.1 Post Type

Post type is the way in which WordPress can sort content. [14] It is used to create archives with thematically similar posts. Post type is one of the main areas in WordPress that developers work with. In WordPress blog posts and pages are all post type and developer needs to add new post type to extend functionality. Using the `register_post_type()` function you can register your own post type. WordPress by default offers several post types:

- Posts
- Pages
- Attachments
- Revisions
- Navigation Menus
- Custom CSS
- Changesets

### ■ 2.4.2 Taxonomy

Taxonomy is a way of grouping things, such as post and custom post types, together. [15] For example, default taxonomies are tags and categories. WordPress stores taxonomies in a `term_taxonomy` database table. Developers can register their own taxonomies in the same table along with existing ones.

### ■ 2.4.3 Plugins

There are more than 50000 free WordPress plugins to expand project's functionality [16]. The most popular are, for example, Yoast SEO, Akismet, Imagify, WP Rocket. Plugins allows you to expand the functionality of the website and modify its inner logic. Using plugins, you can add to your project galleries, forums, social media widgets, spam protection, search engine optimization, etc. You can also create complex plugins by yourself with HTML, CSS and JS, as well as PHP. A WordPress plugin should have a required unique header and should be located in `wp-content/plugins` directory. A plugin applies to all WordPress themes, executes on page load when activated and serves one main purpose. [17]

### ■ 2.4.4 Themes

WordPress themes are templates that define the look, structure and style of website pages. Themes could be found in `wp-content/themes` directory. [18] They composed of a set of files, that define a layout of web pages. Such files are written on PHP and HTML and include `header.php`, `page.php`,

**sidebar.php, footer.php**, etc. The template files may include calls to WordPress core functions such as `get_header()`, `get_footer()`, `get_sidebar()`, etc. This allows to display content dynamically. Administration of themes is possible through the WordPress admin panel, where user can activate or deactivate themes.

## ■ Sidebars and widgets

Sidebars and widgets are theme features. **Sidebar** is a feature provided for displaying side information on a web page, they are located in theme's directory, usually in **sidebar.php** file [19]. Sidebars are placed on the side, bottom or top of the page, depending on design settings of the theme. Administrator of the website can customize sidebars, for example, using Custom Sidebars plugin. Sidebar area usually contain widgets. **Widgets** are small blocks of content or functionality, for example, calendar, search, banner, etc. Usually, they could be found in **functions.php** file in theme's directory [20]. WordPress allows users to create custom widgets as well.

## ■ 2.4.5 Hook

Hook is the main mechanism of interaction, communication between the core of WordPress and plugins, themes [21]. To use a hook, you need to write a Callback function. There are two types of hooks:

- **Actions** allow you to add data or change the behavior of WordPress [22]. They are activated at a specific point of WordPress core code flow. Actions step into the code flow to perform their action. After the action is executed, the flow returns to its normal course. Callback functions for actions don't return anything.
- **Filters** allow you to modify data during the execution of a WordPress core flow [23]. A filter Callback function takes a variable, modifies it, and returns it back.

There are three the most important hooks for creating a plugin for WordPress [24]:

- The activation hook (**register\_activation\_hook()**) runs when plugin is activated. It is usually used for setting up some settings.
- The deactivation hook (**register\_deactivation\_hook()**) runs when plugin is deactivating. It may be used, for example, to provide functions to delete temporary data.
- The uninstall hook (**register\_uninstall\_hook()**) runs after plugin is deleted. It may be used to clean all data created by plugin.



## ■ 2.5 API

API (Application Programming Interface) is a description of how one computer program interacts with other programs [25]. It is an interface for communication between applications. API contains a collection of classes, methods, functions, and protocols that allow one application to communicate with another, mutually synchronize user or system data. API simplifies the programming process when creating applications by abstracting the basic implementation and providing only the objects or actions developer needs. In a web environment, the API is used to communicate between a web solution and a remote API. Both parties usually communicate through standard communication protocol HTTP/S with platform-independent formats such as XML, CSV or JSON. These formats send data in a standardized notation and both sides are able to read these notations (parse) and create new (generate).

### ■ 2.5.1 HTTP Statuses

The statuses are sent with the response to the request through the API. The response statuses and their decoding are listed below. HTTP status is a part of the first line of the server response for HTTP requests. A set of statuses is a standard, they are described in the relevant RFC documents. A status contains three symbols. The first one indicates the status class:

- 1xx - Informational;
- 2xx - Success;
- 3xx - Redirection;
- 4xx - Client Error;
- 5xx - Server Error.

### ■ 2.5.2 REST API

REST (REpresentational State Transfer) is an architectural style, a set of concepts and principles [26]. REST implements client and server independently. Services in the REST API interact via HTTP (Hypertext Transfer Protocol).

### ■ 2.5.3 WordPress REST API

The WordPress REST API is the application interface for interacting with a WordPress site. The API provides REST endpoints (URLs) representing built-in WordPress data types, for instance, Taxonomies and Post Types. Using API, an application sends and receives data in JSON (JavaScript Object Notation) format to endpoints to create, modify, or query content on the website.



## Chapter 3

### Analysis

In this chapter I recall main goals of the project, then describe the original state of website and application from a visual and implementation points of view, as well as the desired outcome with customer's requirements.

#### 3.1 Goals of the project

As I mentioned in the Introduction [1], main goals of this thesis are as following:

- To understand the structure of the sponsor's website, applications, plugins, etc.
- To analyze sponsor's requirements.
- To propose and implement a WordPress plugin for managing the partners' promotions on the website and applications. This plugin's user interface will be more user-friendly. Also, it should provide more options than the current one.
- To design and implement a new API for this plugin as well as for older plugins.

The desired result is described in more detail at the end of the chapter.

#### 3.2 Sponsor

Czech-American TV also known as CATV USA is a non-profit organization founded in 2003. The main purpose of this organization is to spread knowledge about Czech culture and traditions in the United States. The founder and director of CATV USA is John Honner, an American producer of Czech descent. CATV USA website ([catvusa.com](http://catvusa.com) [1]) offers Czech language lessons, articles on Czech traditions, recipes of Czech cuisine, ancestry searches for Americans with Czech roots, facts about the Czech Republic, three internet radios with Czech folk, classical music and cultural podcasts. Same content is in CATV USA AndroidTV application. Professionals, students and teachers



Figure 3.1: Czech-American TV logo

from the USA and the Czech Republic participate on the improvement of website and application as volunteers. They maintain web pages, create content for videos and lessons, etc. Students from Czech universities, including CTU, have been working on the development and management of the website, content, etc. for more than ten years. As a result, application for AndroidTV, plugins and WordPress templates have been developed.

### 3.3 Current state

#### 3.3.1 Website

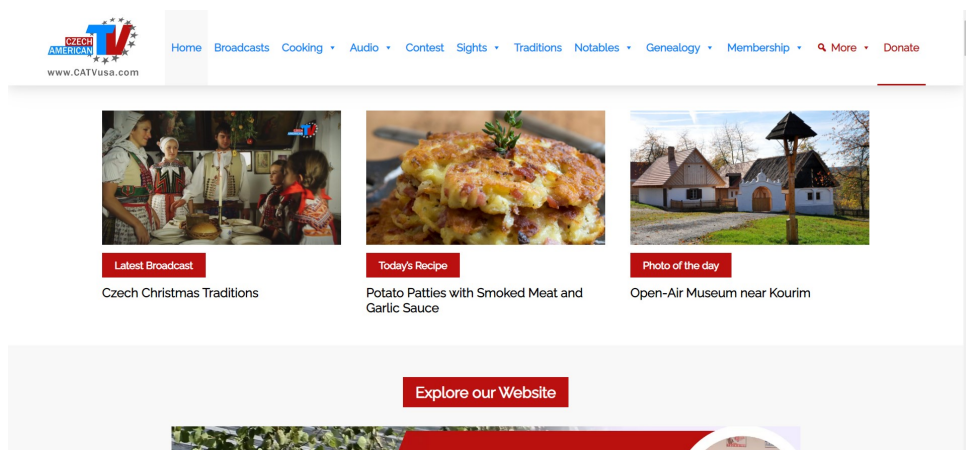


Figure 3.2: Homepage

CATVUSA website is located on [catvusa.com](http://catvusa.com) [1]. It has posts with lessons, traditions, information about Czech cities, as well as three radio stations

and broadcasts. The internal data structure is managed through WordPress admin page. Content (videos and pictures) is gathered into different post types. Each post type has its own taxonomy. Users' membership are managed through the **S2Member Pro** plugin. This plugin allows you to assign a level of access to each user [27]. Video content is managed in the **WPlyr** plugin, created by a student of UWB Jakub Vaverka [28]. Usually, videos are uploaded to the WordPress media storage, from which they go to the web. However, on CATVUSA pages videos are uploaded to the plugins directory. WPlyr plugin integrates uploaded video into the content on the page. Each video post has an intro, a video itself and an outro. This plugin helps to create and maintain that structure. Radio stations are managed by a **Radio Manager** plugin, created by a CTU student Karel Vrabec [29].

### 3.3.2 Partners banners

On many pages of the website there are banners with information about the partners of the organization. CATVUSA is a non-profit organization, therefore it is important to represent and thank sponsors and partner organizations.

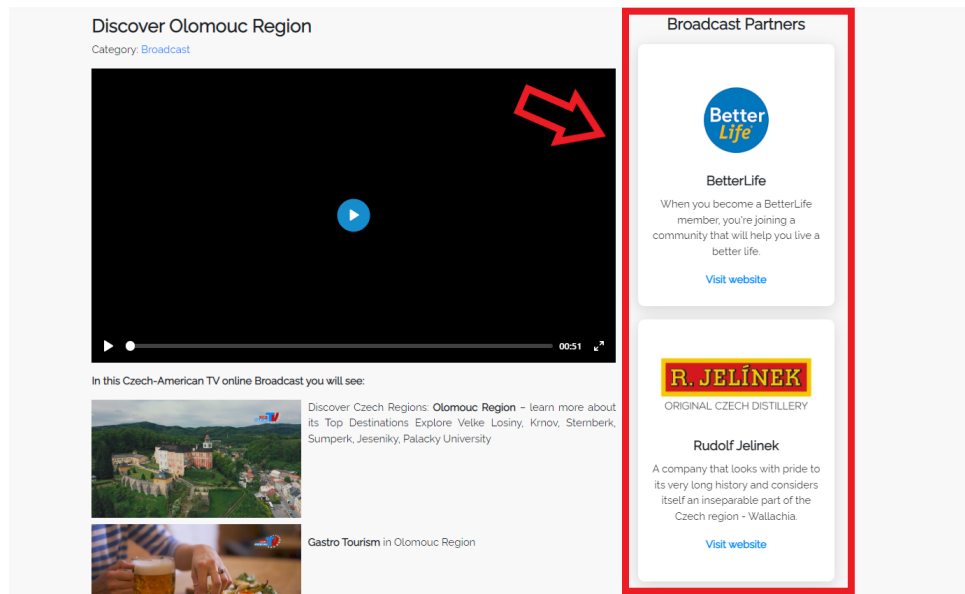


Figure 3.3: Banners with partners

Also, there are partners representation on homepage in the Broadcast Partners section and Partners' Stories section.

Management of these banners is performing by the user in admin panel. Administrator can manage banners' settings through the **Custom Sidebars** plugin [30]. However, this process is very inconvenient. Admin must rewrite a part of a pre-written code to add a new partner banner (the process is described in more detail in the next section 3.3.2). The customer is not a programmer and, despite the fact that this process is described step by

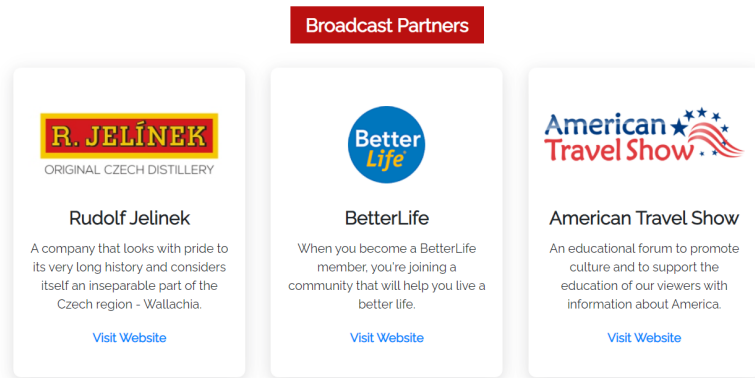


Figure 3.4: Broadcast Partners on Homepage

step in the manual, he complains about the complexity and not user friendly interface.

The **Smart Slider** plugin [31] controls the display and scrolling of banners with partner logos in the carousel on Homepage and Our Partners page. The plugin provides a convenient and intuitive dashboard interface for managing banners.

### ■ Custom Sidebars

Custom sidebars is a WordPress plugin by the WebFactory Ltd [30]. It is a flexible widget area manager that allows user to move, add, edit and delete classic widgets and sidebars on website pages. Administration of the Custom Sidebars for the CATVUSA website is available through the admin menu.

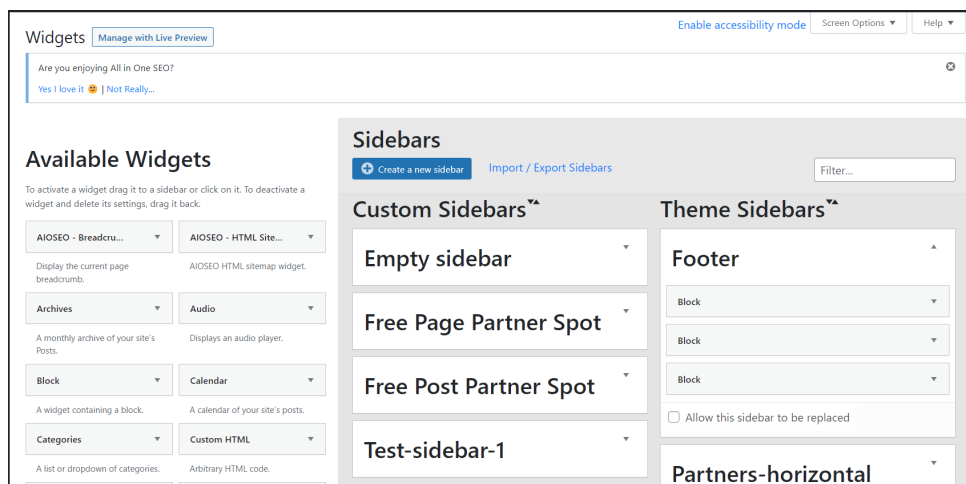


Figure 3.5: Custom Sidebars Administration user interface

As stated in the theme written for the CATVUSA website, each page of the post or page type contains a **div class="broadcast-content"** and a **div class="broadcast-partners-container"**.

The "broadcast-content" div has the content of the page itself, whether it is text, video player, pictures, etc.

The div class="broadcast-partners-container" contains a sidebar with a widget with partner's banner. It includes a div with the title Page partners and a div containing a banner with the partner's name, description, link to the website and logo.

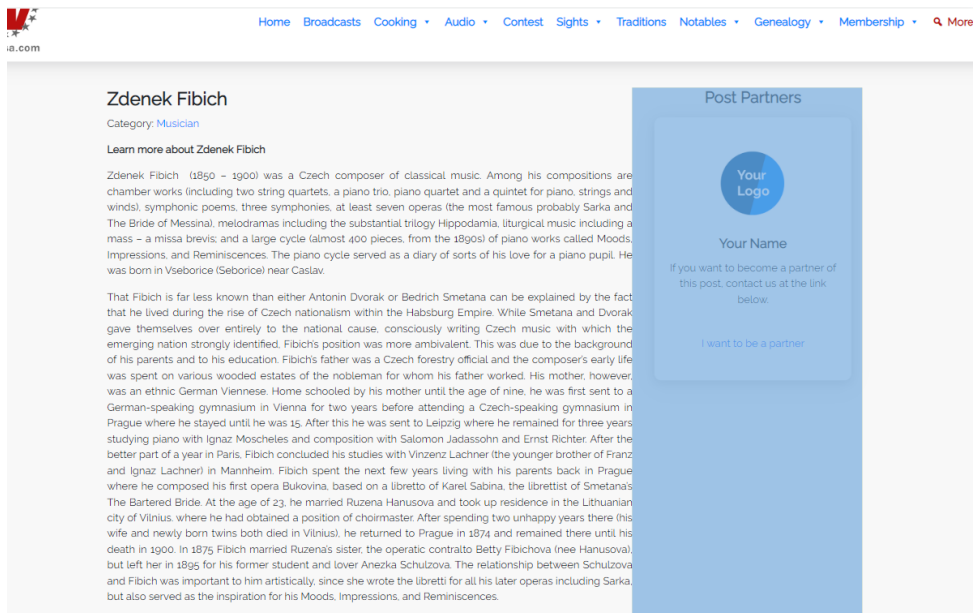


Figure 3.6: A sidebar that contains partner banners

```

<div class="content-container">
  <div class="broadcast-page">
    <div class="row"> flex
      ::before
      <div class="broadcast-content"> ... </div>
      <div class="broadcast-partners-container"> ... </div>
      ::after
    </div>
  </div>
</div>

```

Figure 3.7: HTML structure of a page. A div element with a sidebar for partners banners and a div with the content of the page

This page's structure is declared in the website theme, in the files `wp-content/themes/tv/singular.php` and `wp-content/themes/tv/custom-page-class.php` (fig. 3.8).

```

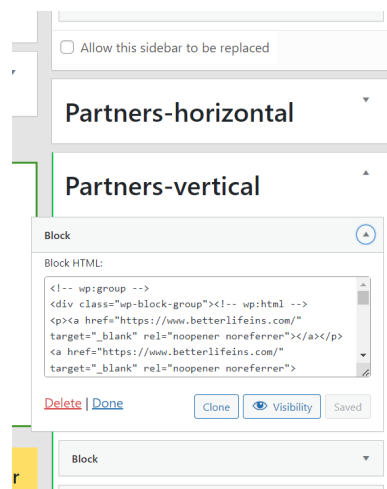
<div class="broadcast-partners-container">
  <?php
  if($post->post_type == 'broadcast') {
    echo "<h4><b>Broadcast Partners</b></h4>";
  } else if(is_page()) {
    echo "<h4><b>Page Partners</b></h4>";
  } else{
    echo "<h4><b>Post Partners</b></h4>";
  }
  dynamic_sidebar( index: 'partners-vertical');
  ?>
</div>

```

**Figure 3.8:** HTML structure of banner's container

To add a new banner to a web page, user goes to the Custom Sidebars administration, copies and pastes into the desired field HTML-code, adding there all the necessary data about the partner (name, link to the site, description and link to a logo picture). Then user sets, on which pages he wants to see this banner.

After, the plugin takes the entered HTML code of and inserts it into a **div class="broadcast-partners-container"** on the desired page/pages. Deletion and editing takes place on the admin page of the Custom Sidebars plugin.



**Figure 3.9:** Administration is not user-friendly



## ■ Problems with Custom Sidebars

The Custom Sidebars plugin, however, does not fully satisfy all the wishes of the customer. The main complaints are that adding new banners is inconvenient for the user who is not familiar with HTML. The customer needs to edit the partner's data in the HTML code himself.

Also, on each page, where the banners are, he would like to see them randomly changing their order among themselves, once in a while. These functions are not available in the Custom Sidebars.

Despite the fact that the Custom Sidebars plugin allows you to select categories and Post Types on which the banner will be displayed, the customer would like to expand this selection so that he can add banners to selected single pages and to all pages of the website.

### ■ 3.3.3 Radio manager plugin

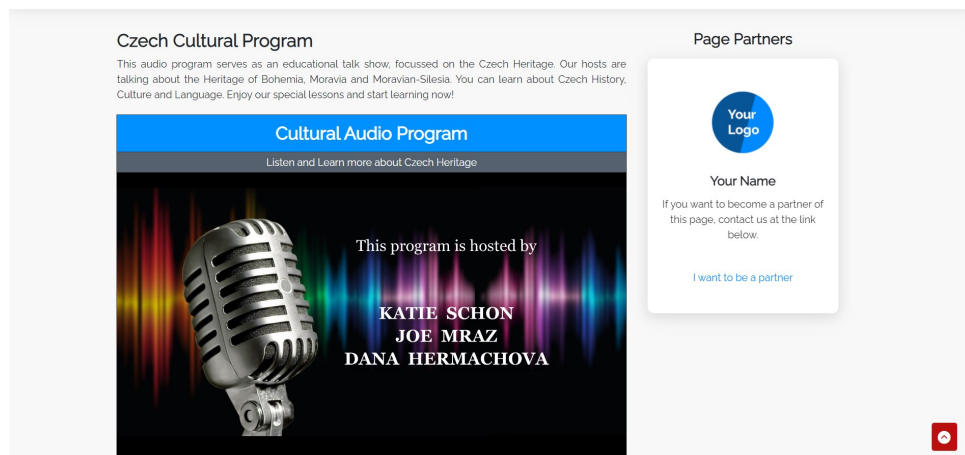


Figure 3.10: Audio program on the website

Radio manager plugin, developed by CTU student Karel Vrabc [29], manages radio stations on CATVUSA web pages. The plugin allows to create and manage playlists, genres, musicians, shuffle playlists etc. On WordPress administration page it has three important sections: Radio Stations, Musicians, Genres.

Every radio station in Radio Stations section is represented as a custom created post type called **rm\_radio\_station**. In the Radio Stations section you can edit, delete or add radio station.

Musicians section is another list where you can add, edit or delete posts with musicians. Each musician is represented as a post named **rm\_musician**. Each post contain a title (musician's name), content (description) and a cover photo. Also, musician can be assign to a specific genre. The listed media are uploaded to the media library and shuffled for playback.

Genres section is a list of posts where you can add, edit or delete taxonomy terms **rm\_genre**. You can assign musicians to different genres, from which

a playlist of the radio station is created.

### ■ Shortcode

Every radio station contains a **shortcode**, which is used for connecting frontend and backend parts. Shortcode has an identification number of the radio contribution station. Admin can copy a shortcode and insert it into any post. For example, this way admin can insert a radio station to the web pages. WordPress function `add_shortcode()` runs after the init hook. It has following parameters: a tag (`rm_radio_station`) and a callback function to run if the shortcode is found. Callback returns HTML code `<div data-rm-output-id="<cislo">></div>`. During parsing, shortcode is replaced by this HTML code with the post ID of the radio station. Thus, after loading the web page, instead of the shortcode, the player of the radio station is displayed.

### ■ Radio station settings

Radio station settings are available on the WordPress admin page. This section contains important playlist settings: you can change image duration and permalink, set music genres, as well as number of musicians and number of recordings for each genre. This way admin sets all the necessary preferences to compose a playlist. This functionality was created by Karel Vrabec, using the Advanced Custom Fields plugin, as a part of his work on the Radio Manager plugin [29].

**Figure 3.11:** Radio station's playlist settings

While loading radio player, this information goes from WordPress web page to the Radio manager plugin. Using `getData($radioStationID)` function

Radio Manager takes the following data and then processes it to compose a playlist:

- radioName
- musicianCaption
- recordCaption
- imgDuration
- logo
- warningData
- postData
- playlistData

```

/**
 * Get the radio station data.
 * @param string $radioStationID - The ID of the radio station page.
 * @return array $data - An array of the radio station data.
 */
public function getData( $radioStationID )
{
    $radioName = get_the_title( $radioStationID );
    $radioLogo = $this->getRadioLogo( $radioStationID );
    $radioSettings = $this->getRadioSettings( $radioStationID );
    $radioWarnings = $this->getRadioWarnings( $radioStationID );
    $radioPlaylist = $this->getRadioPlaylist( $radioStationID );

    $data =
    [
        "radioName"      => $radioName,
        "musicianCaption" => $radioSettings[ "musicianCaption" ],
        "recordCaption"  => $radioSettings[ "recordCaption" ],
        "imgDuration"    => $radioSettings[ "imgDuration" ],
        "logo"           => $radioLogo,
        "warningData"    => $radioWarnings,
        "postData"       => $radioSettings[ "postData" ],
        "playlistData"   => $radioPlaylist,
    ];

    return $data;
} // GET DATA

```

**Figure 3.12:** `getData()` function in Radio Manager plugin

### 3.3.4 AndroidTV application

The AndroidTV application was created a year ago by a CTU student Viktor Sinelnikov [2]. The application was developed to provide the same functionality as the website. There are several plugins that help TV application interact with WordPress core. One of the most important of them is **WP REST API**. This plugin provides REST endpoints to access data from WordPress media storage. The application gets data from WordPress media storage and saves them to the local storage using Room library. In order to send content to the application, admin adds necessary post types to the **android\_tv** taxonomy.

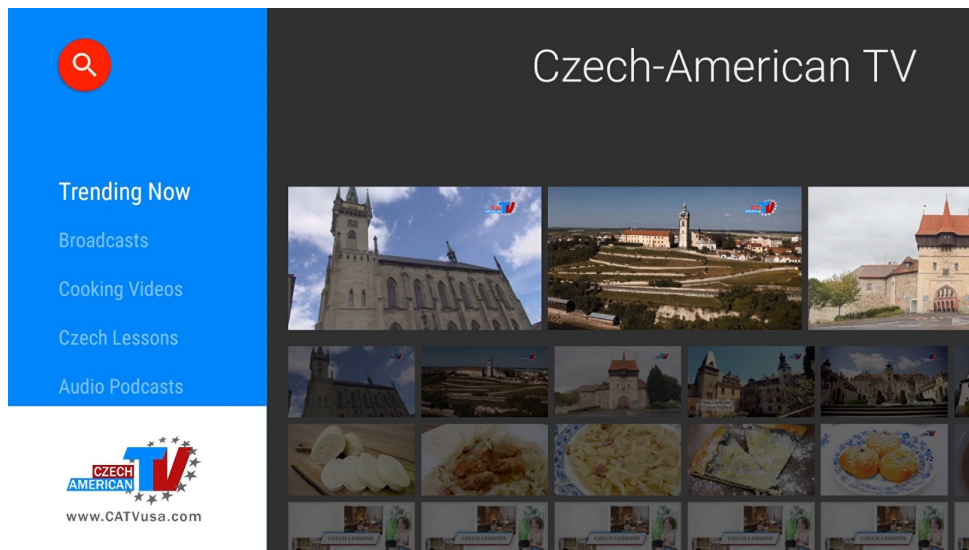


Figure 3.13: AndroidTV application

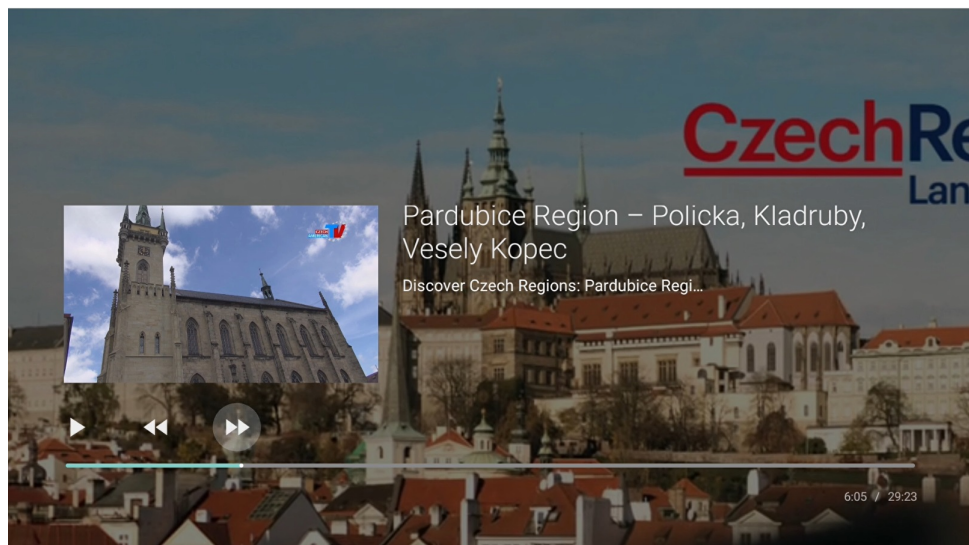


Figure 3.14: Player in TV application

AndroidTV application partly fulfills the requirements. However, there are some problems.

### ■ Problems in TV application

Some parts of the website's functionality are provided by custom plugins, for example, managing radio stations by Radio manager. However, this plugin does not offer the ability to fully interact with its logic, because there is only an API with limited functionality that is not adapted to work with custom plugins. This way, TV application had to use its own business logic in this part of application's functionality. This has resulted in a noticeable difference between the operation of the radio stations on the website and in the TV application. It also led to a limited radio functionality in the application. CATVUSA TV application's logic is not able to adapt dynamically to the changes, that are made on WordPress admin page. If there will be any changes, someone has to go to the source code of the application to make the change also for AndroidTV. For example, names of radio stations are hard coded into the application code. That way, if any radio station will be added to the WordPress admin panel, it will not appear in the TV application.

```
private suspend fun fetchFreshAudioData(): List<CatvusaAudio> {
    val requiredGenres = listOf("cultural", "classical", "folk")

    val musicians = when (val musiciansResult = apiService.getMusicians(requiredGenres)) {
        is Result.Error -> throw musiciansResult.error
        is Result.Success -> musiciansResult.data
    }
}
```

**Figure 3.15:** Hard coded names of radio stations: "cultural", "classical", "folk"

Therefore the application differs from the website in its behavior in some places.

## ■ 3.4 Desired outcome

Customer requirements can be divided into two main parts: 1) requirements for the administration of partners and 2) solution to the the problem of communication between applications and plugins.

### ■ 3.4.1 Customer requirements for administration of partners banners

#### ■ Functional requirements

- Administration page of the solution should allow user to:
  - Add a banner to all pages of the website;
  - Add a banner to all pages of a specific category;

- Add a banner to the specific page;
- Remove a banner from all pages of the website;
- Remove a banner from all pages of a specific category;
- Remove a banner from the specific page;
- Edit a banner;
- Solution should make possible to randomly change the order of displaying banners, for example, once every 24 hours;
- The banner must contain a name, a description, a link to the website of the partner organization and a logo. Logo picture should work as a link to a partner's website as well.

#### ■ **Non-functional requirements**

- User-friendly administration page;
- Compatibility with the latest version of WordPress;
- Correct work in all popular browsers such as Chrome, Firefox, Edge;
- Efficient use of the WordPress database.

#### ■ **3.4.2 Customer requirements for applications-plugins communication problem**

##### ■ **Functional requirements**

- Solution should provide information about radio stations and recordings, including information configured in settings of the Radio Manager plugin;
- Solution should provide information related to the breaks between music in the form of advertisements or announcer messages, which is not currently in the application, but is on the site;
- In application, radio stations should use the same media files (pictures, songs) that are used on the web pages;
- Solution should provide information about partner banners;
- Solution should provide proper documentation.

##### ■ **Non-functional requirements**

- The solution must be designed to work with whatever applications the organization plans to deploy in the future (AndroidTV, Android for smartphones, Samsung, LG etc.);
- The solution must be designed to be easy extendable for the future developed plugins;



## Chapter 4

### Suggested solution

Based on analysis in previous chapter, I propose a solution.

#### 4.1 Administration of partners banners

The decision that meets the customer's requests 3.4.1 is the new WordPress plugin for the partners' banners administration. The plugin will provide banner management in the admin panel and display banners on the site.

After analyzing customer's requirements and the current plugin that displays and manages banners, I realized that there are two ways of resolving the problem: 1) a plugin with user-friendly administration, that will use the Custom Sidebars' functionality on the backend; 2) a new plugin made from scratch. I have decided to create the new plugin from scratch. This decision was made for the following reasons:

- The customer needs not only better user interface, but also more functionality, that Custom Sidebars does not have. For instance:
  - It would be very difficult to add a shuffle of banners to the pages, using Custom Sidebars.
  - Custom Sidebars does not have such functions to add a banner to selected page/pages. In the Custom Sidebars, it is possible to add banner only to selected categories or selected Post Types.
- With a standalone plugin there is no dependency on the Custom Sidebars version. Therefore, if the version of the Custom Sidebars changes, new Administration plugin will not be affected.

Adding, deleting or editing banners on the website will be performed by modifying the content of the posts. I will add a script to the page's code that will insert an HTML element with a banner to the page. With this solution, user will need to simply fill the form to create a new banner or click the button to delete a banner. Thus, it will be more convenient for the user to interact with the banners' administration. Structure, appearance and style for banners will be the same as in the current Custom Sidebars' version.

## ■ 4.2 Communication between custom plugins and applications

### ■ 4.2.1 Description and reasoning of the proposed solution

The most relevant solution that fulfills customer's requirements 3.4.2 is to create a WordPress plugin, that will create a possibility to communicate with other plugins developed for the CATVUSA. New plugin should focus on supplying access to the data.

One of the important plugins is the Radio Manager. Using its example, I will describe solution.

The AndroidTV application [2], as it was written earlier, does not use Radio Manager plugin for compiling playlists. It does not have access to the radio station settings and to the data entered by the admin.

One of the biggest advantages of that solution is the new plugin will provide the data that can not be accessed through the WP REST API, such as plugin-specific settings, information, and the results of some methods. There will be no need to get into the application code to manually change, for example, the names of radio stations or station's settings.

As the customer notes, in the future, new plugins with more complex logic may appear. For these future plugins, it is possible to extend the API and add new endpoints, which will give an access to call their functions.

In the course of analyzing the structure and operation of WordPress sites, I found out that authorization is not necessary.



## Chapter 5

### Implementation

In this chapter I describe the implementation of the solutions that were outlined in the previous chapter (Chapter 4).

Implementation and launch of the project were carried out on a local server using technologies XAMPP, FileZilla and PHPStorm as a development environment.

#### 5.1 Partner Banner Administration

The Partner Banner Administration plugin is implemented using PHP, JavaScript, CSS and HTML languages. The work of the API was tested and checked on a local server.

##### 5.1.1 Frontend

Administration main page consists of a header, a button to add a new banner, and a table with all banners (fig. 5.1).

The table consists of the following columns:

- Banner ID
- Partner's name
- Partner's website
- Partner's logo
- Description
- Actions:
  - Delete
  - Edit
  - Preview

The table is filled dynamically by sending AJAX request to the server (code snippet 5.2), receiving data about banners in JSON format and processing them (code snippet 5.3).

## Partners Banner Administration

Administration of partner's banners. Here you can add/delete/edit/preview banners in the table below.

Banners [+ Add new](#)

ID	Logo image	Partner	Website	Description	Location	Actions
1198840		Partner Name Example	https://examplewebsite.com	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquid ex ea commodo consequat.	All pages	<a href="#">Preview</a> <a href="#">Edit</a> <a href="#">Delete</a>
1198838		Partner Name Example	https://google.com	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquid ex ea commodo consequat.	Categories: Class, Castles Pages: Czech Potato Salad	<a href="#">Preview</a> <a href="#">Edit</a> <a href="#">Delete</a>
					Categories: Traditions, Facts, Cooking Videos Pages: Pardubice Region –	<a href="#">Preview</a>

Figure 5.1: Administration of Partner's banners.

```

fetch('/catvusa_wp/wp-admin/admin-ajax.php?action=get_banners')
  .then(response => response.json())
  .then(data => {
    updateTableWithNewData(data);
  }).catch(error => console.error(error));
}

```

Figure 5.2: Request to the server to get data about all banners to display them in the table.

```

tr.innerHTML = `
<td>${banner.id}</td>
<td style="max-width: 100px; max-height: 100px;"></td>
<td>${banner.name}</td>
<td>${banner.website}</td>
<td>${banner.description}</td>
<td>${allPages} ${categoriesList} ${pagesList}</td>
<td>
  <button class="button viewButton" data-id="${banner.id}">
    <i class="dashicons dashicons-visibility"></i>Preview
  </button>
  <button class="button editButton" data-id="${banner.id}">
    <i class="dashicons dashicons-edit"></i>Edit
  </button>

  <button class="button deleteButton" data-id="${banner.id}">
    <i class="dashicons dashicons-trash"></i>Delete
  </button>
</td>`;
tbody.appendChild(tr);

```

Figure 5.3: The process of creation of the each row in the table.

Admin can add a banner by clicking on the "Add new" button. After, user fills in the information about the banner and selects a location, where he wants to see the banner: "All pages" or "Certain Pages" (Categories and/or Pages) in the Add new banner popup (fig. 5.4, 5.5).

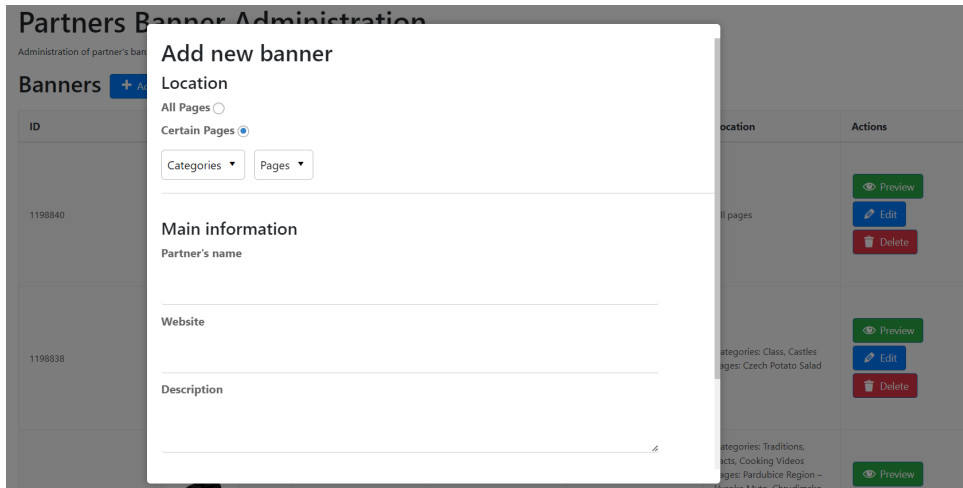


Figure 5.4: Popup with form to add a new banner

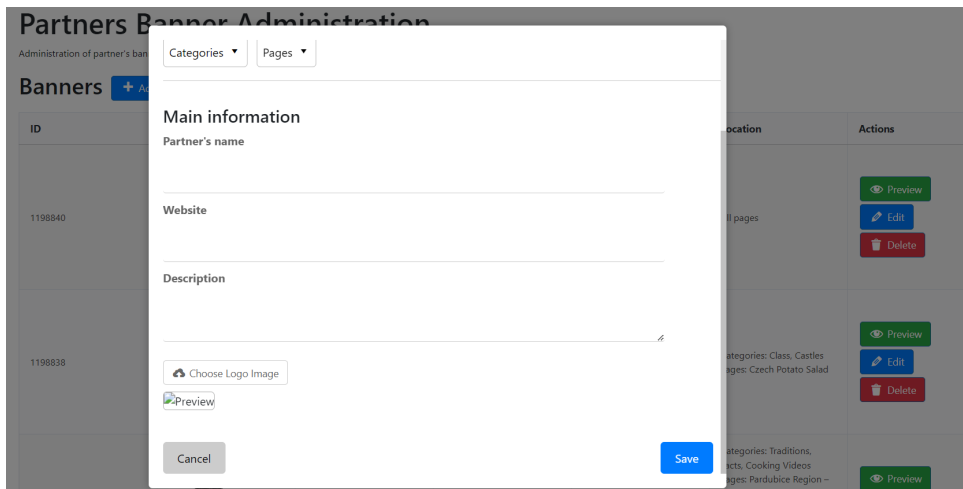


Figure 5.5: End of popup with form to add a new banner.

The list with categories and pages in the form (fig. 5.6) is built dynamically by calling JQuery functions to make AJAX GET requests to the server (code snippet 5.7).

**Figure 5.6:** Categories in form

---

```
$.get('/catvusa_wp/wp-admin/admin-ajax.php', {
  action: 'get_pages'
}, function (data) {
  // Processing data
});
```

---

**Figure 5.7:** The process of getting list of pages from server.

When user chooses the checkbox in one of the dropdown lists, checkbox label (which is a name and ID of the page or category) is pushed to the special array indicating the future location of the banner (code snippet 5.8). This array then will be delivered to the server.

---

```
checkbox.addEventListner('change', function (event) {
  if (event.target.checked) {
    banner.categories.push({
      name: event.target.value,
      id: event.target.id,
    });
  } else {
    const elIndex = banner.categories.findIndex((el) => el.name
      === event.target.value);
    banner.categories.splice(elIndex, 1);
  }
});
```

---

**Figure 5.8:** The process of creating a list of banner's checked categories.

After clicking "Save" button, the form is checked for completeness, and if there is any empty field, the user sees a notification about this (fig. 5.10).

**Location**

All Pages

Certain Pages

Categories ▼ Pages ▼

Please, select location for your banner

Choose Logo Image

Please, upload an image

**Figure 5.10:** Warnings in form.

If the fields are correctly filled, the formData object is created, filled with data from the form and sent to the backend using jQuery AJAX POST request. Then the backend processes the data (described in the next section 5.1.2).

After clicking an Edit button, a popup appears with pre-filled data about the banner (fig. 5.11). User can easily edit this data the way he filled the form when created the banner.

**Edit banner**

**Location**

All Pages

Certain Pages

Categories ▼ Pages ▼

---

**Main information**

**Partner's name**

Partner Name Example

---

**Website**

https://examplewebsite.com

---

**Description**

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquid ex ea commodi consequat.

**Figure 5.11:** Popup with form to edit a banner.

After clicking a Delete button, administrator must confirm his action in the modal window that appears (fig. 5.12). After confirmation, frontend sends AJAX request with the banner's ID to delete this banner, then reloads the page content without deleted element.

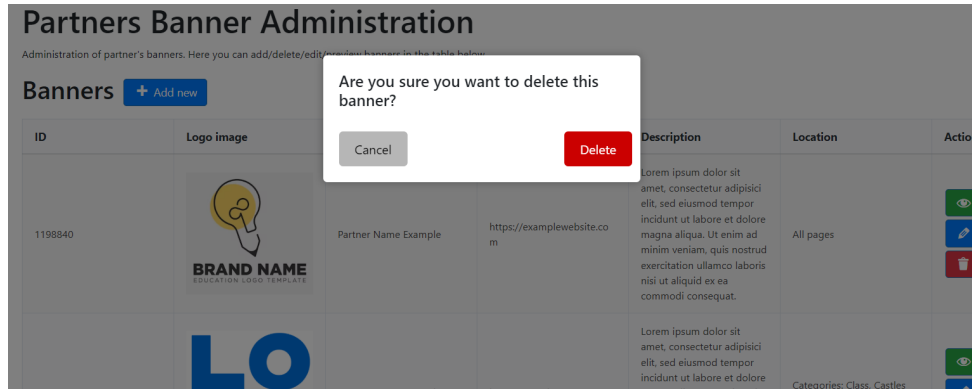


Figure 5.12: Confirmation window to delete a banner.

Preview popup appears by clicking a Preview button (fig. 5.13). The preview is made by inserting inner HTML code into the container of the Preview popup. Inner HTML contains banner's HTML structure and filled data from the chosen banner. It uses same CSS styles, as the actual banners as well.

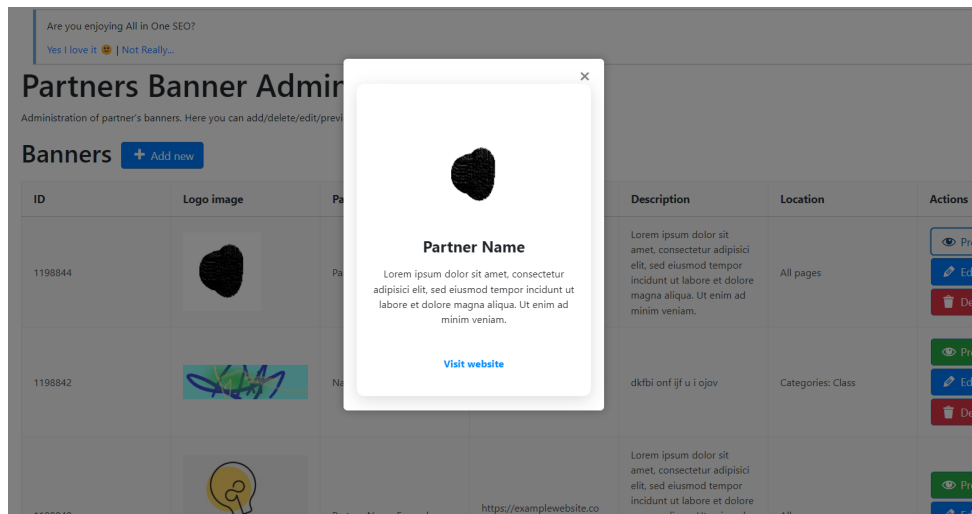


Figure 5.13: Banner's preview.

## 5.1.2 Backend

### Plugin Set Up

The main class of the plugin is the `PartnerBannerAdministration.php`. It has functions for adding and rendering the plugin administration page in the

admin panel. These functions are called with WordPress `add_action` hooks. Rendering of the page is activated with the WordPress hook (code snippet 5.14).

---

```
add_action('admin_menu', \$_callback)
```

---

**Figure 5.14:** The hook that adds the main page of the plugin.

This hook calls function that processes the render of the page, creates HTML structure, and also adds styles and scripts using the `wp_enqueue_script` and `wp_enqueue_style` WordPress functions.

The plugin is initialized using the WordPress hook at the stage of loading plugins, when all the necessary resources and functions of the plugin are already available.

Activation and deactivation hooks (code snippet 5.15) are registered during plugin activation. These hook ensures correct registration of 'banner' Post Type on activation and deleting this Post Type during deactivation.

---

```
register_activation_hook(__FILE__, 'partnerBanner_activation');
register_deactivation_hook(__FILE__, 'partnerBanner_deactivation');
```

---

**Figure 5.15:** Activation and deactivation hooks.

## ■ Saving banners' data to the database

The process of saving data to the database can be split into three steps:

- 1. To add a logo image.

In order to save a logo image, backend adds a new attachment with the image to the WordPress Media Library (code snippet 5.16).

---

```
// Creating attachment
$attachment = array(
    'guid' => $upload_dir['url'] . '/' . $filename,
    'post_mime_type' => $file['type'],
    'post_title' => preg_replace('/\.[^.]+' . '$/', '', $filename),
    'post_content' => '',
    'post_status' => 'inherit'
);

//Saving attachment
$attachment_id = wp_insert_attachment($attachment, $filepath);
```

---

**Figure 5.16:** Inserting Logo to the WP media library

By doing this, we can receive attachment ID, which will be used to save image path to banner metadata.





---

```
update_post_meta($post_id, 'banner_website',
    $_POST['website']);
```

---

**Figure 5.19:** Saving Post metadata to the WP database.

After, attachment ID and URL are saved to the banner meta data in database (code snippet 5.18).

At this point banners' information is successfully added to the database, however, there are no banners on the webpages yet.

### ■ Adding banners to the webpages

As was mentioned in the Analysis chapter 3.3.2, almost every page has a container for banners declared in the CATVUSA Theme. The Custom Sidebar plugin worked with this page's structure. However, in order to display banners on a page with my plugin, it was necessary to change this structure in the Theme and remove the banners' container. The new plugin inserts the banners' container dynamically only to the pages where banners should be.

The main principle of adding a banner to a page has several aspects:

- 1. Receiving a list of pages on which the banner should appear.
- 2. Creating an HTML element of a banner.
- 3. Checking if there is an already created container for banners on the page.
  - If yes, attaching the HTML to a special script and adding the script to the content of the page. Banner will appear in the first free space in the banners' container.
  - If not, adding the container for banners. Then attaching the HTML to a special script and adding the script to the content of the page.

These steps are described in more detail below:

To add banners to their locations, server gets their categories and pages arrays, as well as the Boolean value indicating if the banner should be on all pages. Server goes through these data and for the each page evaluates if it should contain specific banner.

Then, on every page it takes the page's ID and a page's content. Server checks if the page has a container for the banners. If not, then this container is created, filled with data and the the banner's container.

The HTML (code snippet 5.20) is added to the script that appends the custom HTML to the DOM.

If the banners' container already exists, function finds the next available place for a new banner, fills it with new banner and updates the page's content.

---

```
$html = <div class="broadcast-partners-container">
    <h4><b>Page Partners</b></h4>;
$html .= get_partner_html($website,
    wp_get_attachment_url($attachment_id),
    $name,
    $description,
    $attachment_id);
$html .= </div> ;
```

---

**Figure 5.20:** The process of creating HTML of the banner

## ■ Shuffle

One of the customer's requirements was to change the order of displaying banners on each page once in a while. To create such a shuffle, it was necessary to set a scheduled event (code snippet 5.21).

---

```
if ( ! wp_next_scheduled( 'pb_shuffle' ) && ! wp_installing() ) {
    wp_schedule_event( time(), 'daily', 'pb_shuffle' );
}
add_action( 'pb_shuffle', 'pb_shuffle' );
```

---

**Figure 5.21:** Registration of scheduled event.

This function runs if the hook, that is responsible for the shuffle, is not scheduled and if the WordPress is not running through initial installation or system upgrade.

If these two conditions are fulfilled, the function registers an event that will happen every hour, thus calls the Action hook. This hook helps to execute a function that do the shuffle. It takes all the banners from the database, and finds all the pages that have banners. For each such page, the program finds all the banners one by one in the content of the page. Then it removes everything related to banners from the content, generates it again and re-add it in random order.

## ■ 5.2 API plugin

The API plugin is implemented in the PHP language. The work of the API was tested and checked on a local server with the help of the Postman application.

The plugin performs the work of the API, providing rest-endpoints that return JSON arrays with data. For that purpose WordPress Action hook links the core and the plugin. This hook is called when preparing to serve a REST API request: during initialization of the REST API in WordPress, after all REST API files are loaded and before processing REST requests.

Callback of the Action hook is a function that registers a route (code snippet 5.22).

---

```
add_action('rest_api_init', function () use
    ($RadioManagerAPI_instance) {
    register_rest_route('rm_extended_api/v1', '/radio/data/', array(
        'methods' => 'GET',
        'callback' => array($RadioManagerAPI_instance,
            'getRadioManagerData'),
    ));
});
```

---

**Figure 5.22:** The process of registration of the REST route.

Functions called in the REST endpoints' callbacks are contained in the RadioManagerAPI class and PartnerBannerAdministrationAPI class.

### ■ 5.2.1 Radio Manager API

The principle of data calling is the same for all endpoints that are related to the Radio Manager: API plugin's class gets an instance of the Radio Manager manager's class RMShortcodeCreator. Then it calls RMShortcodeCreator's function that uploads and generates data for the Radio Manager plugin.

The main endpoint returns all information about radio stations:

- Radio name
- Musician capture
- Record capture
- Image duration
- Logo (with title, description and link to image)
- Warning data
- Post data (image and text content)
- Playlist data:
  - Genres (with musicians: name, description, images, introduction, records (title and link to MP3))
  - Number of musicians, number of records per musicians, Boolean value if it is allowed to show website posts

It provides an access to the information received by the Radio Manager plugin (code snippet 5.23). API returns data in JSON format.

Also, there are endpoints that return the same data separately. These endpoints were created due to following reasons:

- To speed up loading of the necessary data. Because of the large amount of data returned by the previously described endpoint, the loading process takes longer.
- For comfort of developers who may work with this API in the future.

Full list of the endpoints is described in the next section 5.2.3

---

```

public function getRadioManagerData(){
    require_once WP_PLUGIN_DIR .
        '/radio-manager/includes/RMshortcodeCreator.php';
    $result = [];
    if (class_exists('Inc\\RMshortcodeCreator')) {
        $shortcodeCreator = Inc\\RMshortcodeCreator::getInstance();

        $args = array(
            'post_type' => 'rm_radio_station',
        );

        $query = new WP_Query( $args );

        if ( $query->have_posts() ) {
            while ( $query->have_posts() ) {
                $query->the_post();
                $id = get_the_ID();
                $result[] = $shortcodeCreator->getData( $id );
            }
            wp_reset_postdata();
        } else {
            return new WP_Error('no_posts', __('No posts'),
                array('status' => 404));
        }
    }

    return rest_ensure_response($result);
}

```

---

**Figure 5.23:** The process of getting data from Radio Manager plugin.

## ■ 5.2.2 Partner Banner Administration API

The main principle of obtaining data about banners is the creation and sending of a SQL query to the database. The process is implemented inside the PartnerBannerAdministrationAPI class.

Query results are processed and contain the following information:

- Banner's ID
- Partner's name
- Pages, where banner should be located on the website

- Image's attachment ID
- Image's URL
- Categories, where banner should be located on the website
- Boolean value showing if the banner should be located on all website's pages.

### ■ 5.2.3 Endpoints

In the following endpoints' description the designation 'baseUrl' is stated for the URL of the website.

#### ■ Get all data from Radio Manager

- **Endpoint:** baseUrl/wp-json/rm\_extended\_api/v1/radio/data
- **Response:** Returns following data in JSON format: Radio name, Musician capture, Record capture, Image duration, Logo (with title, description and link to image), Warning data, Post data (image and text content), Playlist data: Genres (with musicians: name, description, images, introduction, records (title and link to MP3)) as well as Number of musicians, number of records per musicians, Boolean value if it is allowed to show website posts.

```
{
  "radioName": "Folk Music Audio Program",
  "musicianCaption": "Music Band: ",
  "recordCaption": "Song: ",
  "imgDuration": 5000,
  "logo": [ ...
],
  "warningData": [ ...
],
  "postData": [ ...
],
  "playlistData": [ ...
]
},
```

**Figure 5.24:** Get all data (Postman).



### ■ Get warning data from Radio Manager

- **Endpoint:** `baseURL/wp-json/rm_extended_api/v1/radio/warning_data`
- **Response:** Returns information about warnings in JSON format, including title of the warning, message and link to the authorization as a member.

```
{
  "first": "46",
  "step": "46",
  "title": "Membership Warning Popup",
  "message": "Do you want to continue listening without interruptions? Then click the OK button and register! your help.",
  "cancel": "Dismiss",
  "confirm": "OK",
  "link": "https://catvusa.com/membership-signup/"
}
```

Figure 5.27: Get warning data (Postman).

### ■ Get settings data from Radio Manager

- **Endpoint:** `baseURL/wp-json/rm_extended_api/v1/radio/settings_data`
- **Response:** Returns image and text content in JSON format. This information is set in the Settings of the Radio Manager plugin.

```
{
  "image": [
    {
      "title": "Learn more about Bohemian and Moravian Regional Cuisine",
      "description": "Click on the image to discover more about this Czech tradition...",
      "src": "https://www.catvusa.com/wp-content/uploads/2018/07/czech.meal_.jpg"
    }
  ],
  "content": "<p><strong>Learn more about Bohemian and Moravian Regional Cuisine</strong></p>\n\n</p>\n\n<p>Typical ingredients used in Czech cuisine are potatoes, flour, legumes, vegetables, and bread. Czech meals are typically served in two or three courses. The first course is soup. It is the most important part of all the meals. Soup is often served as a first course and can be translated as: "Soup is the base". The second course is the main dish. There might be soups as chicken, beef or vegetable broth; cabbage soup with sausage; garlic soup or potato soup."</p>"
}
```

Figure 5.28: Get settings data (Postman).

## ■ Get banners' data from Partner Banner Administration

- **Endpoint:** `baseURL/wp-json/pba_extended_api/v1/banners`
- **Response:** Returns banner's ID, partner's name, pages where banner should be located on the website, categories where banner should be located on the website, Boolean value showing if the banner should be located on all website's pages or not, image's attachment ID, image's URL.

```
{
  "id": "1198840",
  "name": "Partner Name Example",
  "description": "Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed eiusmod tempor inci
    ullamco laboris nisi ut aliquid ex ea commodi consequat.",
  "website": "https://examplewebsite.com",
  "pages": [],
  "img_at_id": "1198839",
  "img": "http://localhost/catvusa_wp/wp-content/uploads/2023/05/logoexample.png",
  "categories": [],
  "all_pages": "true"
},
```

**Figure 5.29:** Get banners' data (Postman).



## Chapter 6

### Testing

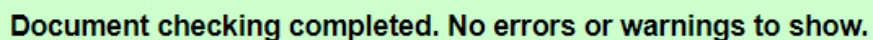
This chapter describes the process and result of code testing and user testing.

#### 6.1 Testing code with a validation service

Code testing is an important step in software development. It allows developers to identify and fix bugs, improve the code and interaction between user and the website.

Validators are tools that verify code conforms to certain standards and rules, helping to identify potential problems and bugs. One of the most popular validator for websites is the Markup Validation Service by W3C [32].

Results of testing my code on the Markup Validation Service website is following: "Document checking completed. No errors or warnings to show"



Document checking completed. No errors or warnings to show.

Figure 6.1: Markup validation result.

#### 6.2 User Testing

User testing, or Usability testing, is a process in which real users test a website to evaluate its usability and effectiveness: users interact with a website, then give a feedback. User testing is an important part of the project development, because it provides valuable insights from real users. It may help developers to find bugs or make more intuitive and user-friendly interface [33]. Testing includes three parts:

- Pre-testing questions.
- Testing itself with the test scenario.
- Post-testing questions.

### 6.2.1 Pre-testing questions

Pre-testing questions help to understand background information of the users engaging in the testing session [34].

Before testing, all users were asked the following questions:

- What is your occupation?
- What is your age?
- How would you rate your level of confidence in using web applications? (Using scale 1 to 10)

### 6.2.2 Test scenario

Every user is going through the test scenario during user testing.

1. Log in to the admin WordPress page using email and password.
2. Find the Partner Banner plugin in the left menu and open the it.
3. Create a new banner with the following data:
  - Partner's name: Test Company.
  - Website: <https://www.company.com>.
  - Description: Wealthy company sponsoring projects with great potential.
  - Logo: choose an image on your computer.
  - Banner's location: 'Museums' category and 'Historie skoly' page.
4. Find your banner in the table and check how it looks like on the website without going to the webpages (using Preview function).
5. Change a name in your banner to 'The Good Company', change the partner's logo.
6. Check if the name and logo were changed.
7. Change the location of your banner, make it appear on all website's pages.
8. Check if location was changed.
9. Delete your banner.

After testing, all users were asked the following questions:

- How would you describe your experience with the Partner Banner Administration?
- What did you like the most?
- What did you like the least?
- Is there anything you would like to change?

### ■ 6.2.3 Feedback

#### ■ User 1

Pre-testing questions:

- **Occupation:** Copywriter;
- **Age:** 23;
- **Level of confidence in using web applications:** 7/10.

Post-testing questions:

- **Experience with the Partner Banner Administration:** Positive.
- **The most liked things:** Everything is clear and intuitive, except for the small things.
- **The least liked things:** It is not immediately clear what does 'Location' header mean (in the forms). The plugin section in the WordPress left menu is far away, it is not clear where to find it. The results of the selection in Location (in the forms) are not written nearby.
- **Recommended improvement:** To add a divider (in the forms) between the main header and 'Location'; to add an information about required size of the logo image; to display chosen categories/pages in the forms.

#### ■ User 2:

Pre-testing questions:

- **Occupation:** Programmer
- **Age:** 25
- **Level of confidence in using web applications:** 10/10

Post-testing questions:

- **Experience with the Partner Banner Administration:** Positive.
- **The most liked things:** User interface; simple and clear work with the plugin.
- **The least liked things:** After selecting a category or page in the Location section, the selected items are not displayed in the form. In the table, categories and pages are placed in one column.
- **Recommended improvement:** To fix written above.

### ■ User 3:

Pre-testing questions:

- **Occupation:** Doctor
- **Age:** 43
- **Level of confidence in using web applications:** 5/10

Post-testing questions:

- **Experience with the Partner Banner Administration:** Mostly positive.
- **The most liked things:** Intuitive design, it is clear where to find every option (such as Add, Edit or Delete).
- **The least liked things:** It is not immediately clear what the location of the banner means (but it became clearer after adding the banner, and its visual display on the site). Page search in the Add new banner/Edit banner forms is not noticeable.
- **Recommended improvement:** There are a lot of pages in the page selection in the Add new banner/Edit banner forms, it would be better to compose them or make a more expressive search field.

### ■ 6.2.4 Result

The results of user testing make it clear that despite the design details that can be improved, the implemented plugin is understandable and convenient for users of different ages and different degrees of experience with web applications.

# Chapter 7

## Conclusion

### 7.1 Conclusion

In the process of working on this thesis, I figured out how the WordPress system works [2], as well as how CATVUSA website and applications work [3].

I collected and analyzed customer's requirements [3.4]. Based on them, I proposed a solution [4] in the form of a 1) new plugin for the administration of partners' banners on website pages, and 2) API for custom plugins.

The proposed solution was implemented [5] and tested [6].

The requirements and their implementation in my solution are analyzed below.

#### 7.1.1 Partner Banner Administration plugin

##### Functional requirements

- **Requirement:** Administration page should allow user to add a banner to all pages of the website, to all pages of a specific category, to the specific page.

**Result:** Administration page allows user to add a banner to all pages, to all pages of a specific category, to the specific page of the website [5.1.1, 5.1.2]. In addition, after adding a banner, the user can see what his banner looks like without leaving the admin panel using Preview option [5.1.1].

- **Requirement:** Administration page should allow user to remove a banner from all pages of the website, from all pages of a specific category, from the specific page.

**Result:** Administration page allows user to delete a banner, which removes it from its location on the website [5.1.1].

- **Requirement:** Administration page should allow user to edit a banner.

**Result:** Administration page allows user to edit a banner [5.1.1].



- **Requirement:** In application, radio stations should use the same media files (pictures, songs) that are used on the web pages.

**Result:** Solution provides access to the media files (pictures, songs) that are used on the web pages [1.2.2].

- **Requirement:** Solution should provide information about partner banners.

**Result:** Solution provides information about banners [5.2.3].

- **Requirement:** Solution should provide proper documentation.

**Result:** Solution provides proper documentation [1.2].

#### ■ Non-functional requirements

- **Requirement:** The solution must be designed to work with whatever applications the organization plans to deploy in the future (AndroidTV, Android for smartphones, Samsung, LG etc.).

**Result:** The API provides data in JSON format, therefore any system or application can process this data.

- **Requirement:** The solution must be designed to be easy extendable for the future developed plugins.

**Result:** The solution is extendable for the future developed plugins. For instance, there are two classes implemented in the current API plugin, each for one custom plugin (Radio Manager plugin and Partner Banner Administration) [5.2].







## Bibliography

1. *Czech American TV* [online]. [visited on 2023-01-19]. Available from: <https://catvusa.com/>.
2. SINELNIKOV, Viktor. *Smart TV application development for a non-profit TV* [online]. Praha, 2021 [visited on 2023-01-19]. Available from: <https://dspace.cvut.cz/handle/10467/99112?locale-attribute=en>. Master thesis. Faculty of Electrical Engineering CTU in Prague.
3. *WordPress* [online]. [visited on 2023-01-19]. Available from: <https://cs.wordpress.org/>.
4. *HTML* [online]. [visited on 2023-05-19]. Available from: <https://developer.mozilla.org/en-US/docs/Web/HTML>.
5. *HTML5* [online]. [visited on 2023-05-19]. Available from: <https://developer.mozilla.org/en-US/docs/Glossary/HTML5>.
6. *JavaScript* [online]. [visited on 2023-05-19]. Available from: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>.
7. *CSS* [online]. [visited on 2023-05-19]. Available from: <https://www.w3.org/Style/CSS/Overview.en.html>.
8. *Php* [online]. [visited on 2023-05-19]. Available from: <https://www.php.net/>.
9. *Php* [online]. [visited on 2023-05-19]. Available from: [https://www.tutorialspoint.com/php/php\\_introduction.htm](https://www.tutorialspoint.com/php/php_introduction.htm).
10. *XAMPP* [online]. [visited on 2023-05-22]. Available from: <https://www.apachefriends.org/about.html>.
11. *PhpStorm* [online]. [visited on 2023-05-22]. Available from: <https://www.jetbrains.com/phpstorm/>.
12. *Most popular content management systems* [online]. [visited on 2023-05-17]. Available from: <https://w3techs.com/>.
13. *WordPress architecture* [online]. [visited on 2023-05-17]. Available from: <https://www.hostinger.com/tutorials/wordpress-architecture>.
14. *Post Types* [online]. [visited on 2023-05-17]. Available from: <https://wordpress.org/documentation/article/what-is-post-type/>.



33. *User-testing* [online]. [visited on 2023-05-19]. Available from: <https://mindsea.com/user-testing/>.
34. *Usability-testing-questions* [online]. [visited on 2023-05-19]. Available from: <https://careerfoundry.com/en/blog/ux-design/how-to-write-usability-testing-questions/>.



## Attachment

### 1.1 Partner Banner Administration User Manual

#### 1.1.1 Installation and launch

- 1. Download the source code of the catvusa.com website;
- 2. Open the wp-content/themes/tv/singular.php file;
- 3. Remove the following part of the code:

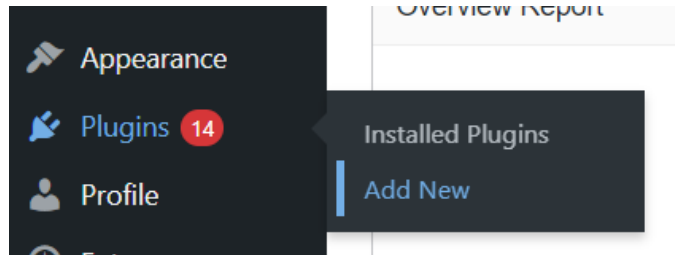
```
<div class="broadcast-partners-container">
    <?php
    if($post->post_type == 'broadcast') {
        echo "<h4><b>Broadcast Partners</b></h4>";
    } else if(is_page()) {
        echo "<h4><b>Page Partners</b></h4>";
    } else{
        echo "<h4><b>Post Partners</b></h4>";
    }
    dynamic_sidebar( index: 'partners-vertical');
    ?>
</div>
```

- 4. Open the wp-content/themes/tv/custom-page-class.php file;

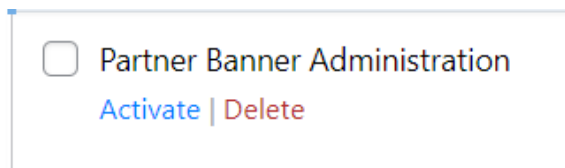
- 5. Remove the following part of the code:

```
<div class="broadcast-partners-container">
  <?php
  if($post_type->name == 'broadcast') {
    echo "<h4><b>Broadcast Partners</b></h4>";
  } else if(is_page()) {
    echo "<h4><b>Page Partners</b></h4>";
  } else{
    echo "<h4><b>Post Partners</b></h4>";
  }
  ?>
  <?php dynamic_sidebar( index: 'partners-vertical'); ?>
</div>
```

- 6. Add the Partner Banner Administration plugin through the admin panel in the Plugins section;

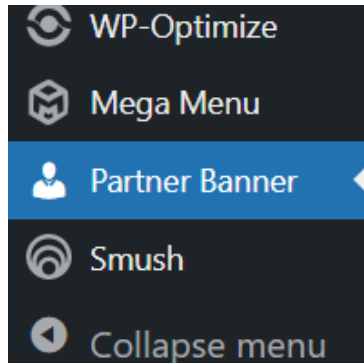


- 7. Click Activate button near the plugin to activate the plugin;



## 1.1.2 Administration Page

You can get to the Partner Banner Administration page through the left menu section in the admin panel



Partner Banner Administration page has a table, where you can find information about all banners.

### Partners Banner Administration

Administration of partner's banners. Here you can add/delete/edit/preview banners in the table below.

Banners [+ Add new](#)

ID	Logo image	Partner	Website	Description	Location	Actions
1198840		Partner Name Example	https://examplewebsite.com	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquid ex ea commodo consequat.	All pages	<a href="#">Preview</a> <a href="#">Edit</a> <a href="#">Delete</a>
1198838		Partner Name Example	https://google.com	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquid ex ea commodo consequat.	Categories: Class, Castles Pages: Czech Potato Salad	<a href="#">Preview</a> <a href="#">Edit</a> <a href="#">Delete</a>
					Categories: Traditions, Facts, Cooking Videos Pages: Pardubice Region -	<a href="#">Preview</a>

## How to add a new banner

- 1. Click the Add new button;

### Partners Banner Administration

Administration of partner's banners. Here you can add/delete/edit/preview banners in the table below.

**Banners** [+ Add new](#)

ID	Logo image	Partner	Website	Description	Location	Actions
1198840		Partner Name Example	https://examplewebsite.com	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquid ex ea commodo consequat.	All pages	<a href="#">Preview</a> <a href="#">Edit</a> <a href="#">Delete</a>
1198838		Partner Name Example	https://google.com	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquid ex ea commodo consequat.	Categories: Class, Castles Pages: Czech Potato Salad	<a href="#">Preview</a> <a href="#">Edit</a> <a href="#">Delete</a>
					Categories: Traditions, Facts, Cooking Videos Pages: Pardubice Region	<a href="#">Preview</a>

- 2. Choose banner's future location: if it should be on all webpages, click 'All Pages'; if it should be on specific page(s), click 'Certain Pages';

**Partners Banner Administration**

Administration of partner's banners

**Banners** [+ Add new](#)

**Add new banner**

**Location**

All Pages

Certain Pages

Categories  Pages

**Main information**

Partner's name

Website

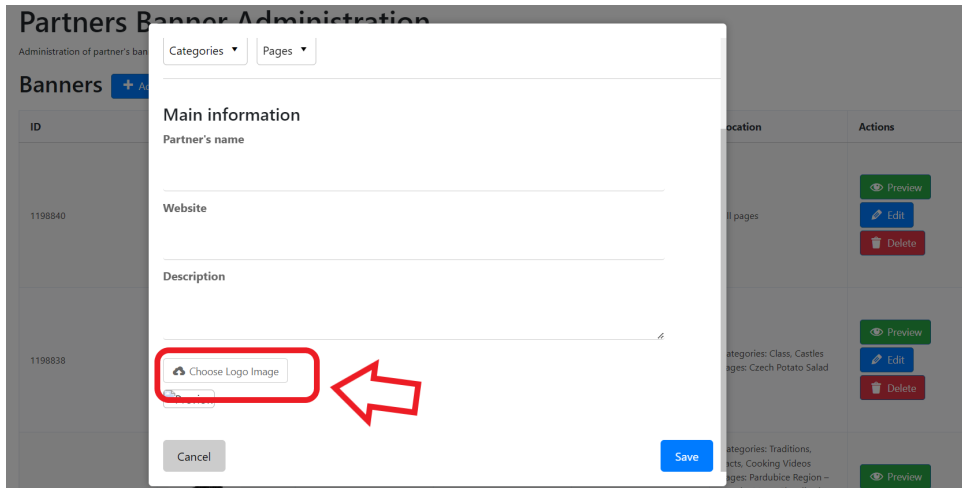
Description

ID	Logo image	Partner	Website	Description	Location	Actions
1198840		Partner Name Example	https://examplewebsite.com	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquid ex ea commodo consequat.	All pages	<a href="#">Preview</a> <a href="#">Edit</a> <a href="#">Delete</a>
1198838		Partner Name Example	https://google.com	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquid ex ea commodo consequat.	Categories: Class, Castles Pages: Czech Potato Salad	<a href="#">Preview</a> <a href="#">Edit</a> <a href="#">Delete</a>
					Categories: Traditions, Facts, Cooking Videos Pages: Pardubice Region	<a href="#">Preview</a>

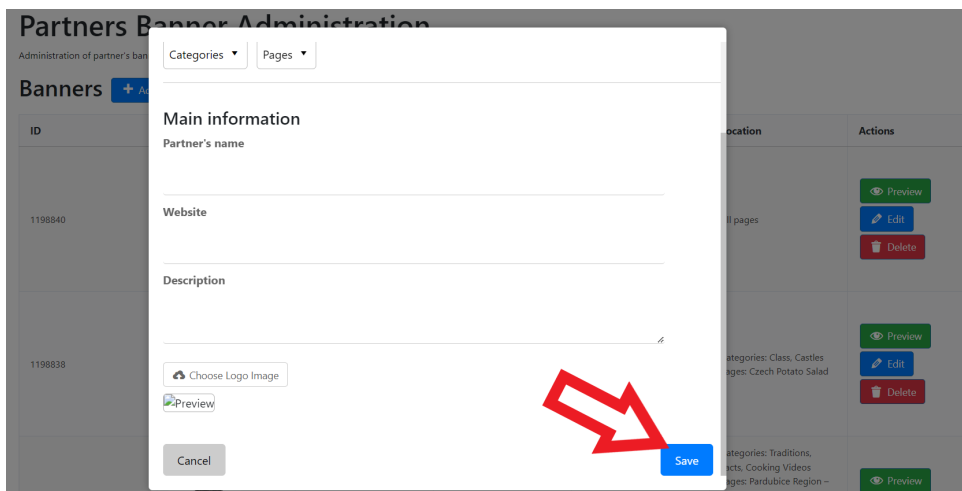
- 3. In case you have chosen 'Certain Pages', choose categories of the website or specific pages, where the banner will be;



- 4. Fill the name of the partner, description of the partner, website of the partner;
- 5. Add logo image (recommended maximum dimension 260 x 101);



- 6. Click on Save button to save the banner. New banner will appear in the table on the Partner Banner Administration page;



- 7. If you do not want to save the banner, click the Cancel button.



## How to delete a banner

- 1. Click the Delete button near the banner;

### Partners Banner Administration

Administration of partner's banners. Here you can add/delete/edit/preview banners in the table below.

Banners [+ Add new](#)

ID	Logo image	Partner	Website	Description	Location	Actions
1198840		Partner Name Example	https://examplewebsite.com	Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquid ex ea commodo consequat.	All pages	<a href="#">Preview</a> <a href="#">Edit</a> <a href="#">Delete</a>
1198838		Partner Name Example	https://google.com	Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquid ex ea commodo consequat.	Categories: Class, Castles Pages: Czech Potato Salad	<a href="#">Preview</a> <a href="#">Edit</a> <a href="#">Delete</a>
					Categories: Traditions, Facts, Cooking Videos Pages: Pardubice Region	<a href="#">Preview</a>



- 2. Confirm that you want to delete the banner;

### Partners Banner Administration

Administration of partner's banners. Here you can add/delete/edit/preview banners in the table below.

Banners [+ Add new](#)

Are you sure you want to delete this banner?  
[Cancel](#) [Delete](#)

ID	Logo image	Partner	Website	Description	Location	Actions
1198840		Partner Name Example	https://examplewebsite.com	Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquid ex ea commodo consequat.	All pages	<a href="#">Preview</a> <a href="#">Edit</a> <a href="#">Delete</a>
					Categories: Class, Castles	<a href="#">Preview</a>

- 3. Banner is successfully deleted.

## How to edit a banner

- 1. Click the Edit button near the banner;

### Partners Banner Administration

Administration of partner's banners. Here you can add/delete/edit/preview banners in the table below.

Banners [+ Add new](#)

ID	Logo image	Partner	Website	Description	Location	Actions
1198840		Partner Name Example	https://examplewebsite.com	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquid ex ea commodi consequat.	All pages	<a href="#">Preview</a> <a href="#">Edit</a> <a href="#">Delete</a>
1198838		Partner Name Example	https://google.com	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquid ex ea commodi consequat.	Categories: Class, Castles Pages: Czech Potato Salad	<a href="#">Preview</a> <a href="#">Edit</a> <a href="#">Delete</a>
					Categories: Traditions, Facts, Cooking Videos Pages: Pardubice Region	<a href="#">Preview</a>

- 2. Edit the fields you want to edit;

## Edit banner

### Location

All Pages

Certain Pages

Categories  Pages

---

### Main information

**Partner's name**

Partner Name Example

---

**Website**

https://examplewebsite.com

---

**Description**

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquid ex ea commodi consequat.

- 3. Click on Save button to save changes;
- 4. If you do not want to save changes, click the Cancel button.

## How to see a preview

1. Click the Preview button near the banner;

### Partners Banner Administration

Administration of partner's banners. Here you can add/delete/edit/preview banners in the table below.

Banners [+ Add new](#)

ID	Logo image	Partner	Website	Description	Location	Actions
1198840		Partner Name Example	https://examplewebsite.com	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquid ex ea commodi consequat.	All pages	<a href="#">Preview</a> <a href="#">Edit</a> <a href="#">Delete</a>
1198838		Partner Name Example	https://google.com	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquid ex ea commodi consequat.	Categories: Class, Castles Pages: Czech Potato Salad	<a href="#">Preview</a> <a href="#">Edit</a> <a href="#">Delete</a>
					Categories: Traditions, Facts, Cooking Videos Pages: Pardubice Region	<a href="#">Preview</a>

2. Now you can see a preview in the opened popup;

Are you enjoying All in One SEO?  
[Yes I love it](#) | [Not Really...](#)

### Partners Banner Administration

Administration of partner's banners. Here you can add/delete/edit/preview banners in the table below.

Banners [+ Add new](#)

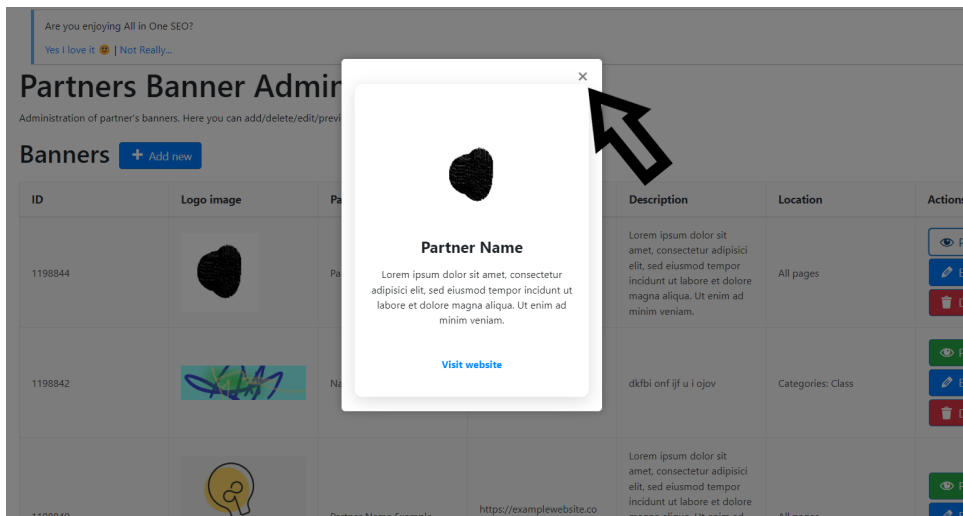
ID	Logo image	Partner	Website	Description	Location	Actions
1198844		Partner Name		Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam.	All pages	<a href="#">Preview</a> <a href="#">Edit</a> <a href="#">Delete</a>
1198842		Partner Name		dkfbi onf ijf u i oiov	Categories: Class	<a href="#">Preview</a> <a href="#">Edit</a> <a href="#">Delete</a>
1198840		Partner Name Example	https://examplewebsite.com	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquid ex ea commodi consequat.	All pages	<a href="#">Preview</a> <a href="#">Edit</a>

**Partner Name**

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam.

[Visit website](#)

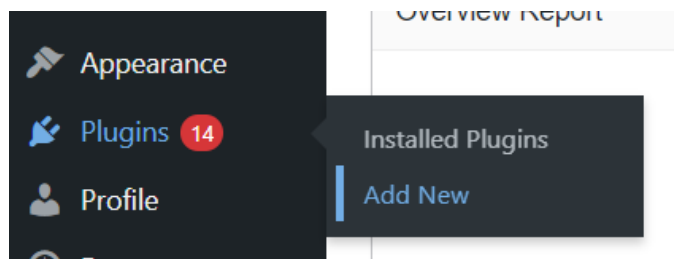
- 3. Click the X to close the preview.



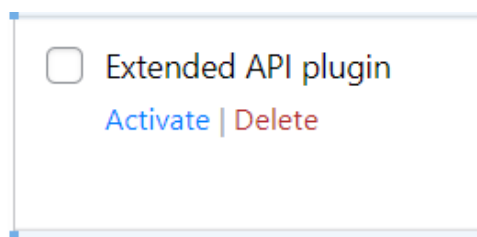
## 1.2 Extended API User Manual

### 1.2.1 Installation and launch

- 1. Install Extended API plugin through the admin panel in the Plugins section;



- 2. Click Activate button near the plugin to activate the plugin;



## 1.2.2 Endpoints

In the following endpoints' description the designation 'baseUrl' is stated for the URL of the website.

### Get all data from Radio Manager

- **Endpoint:** baseUrl/wp-json/rm\_extended\_api/v1/radio/data
- **Response:** Returns following data in JSON format: Radio name, Musician capture, Record capture, Image duration, Logo (with title, description and link to image), Warning data, Post data (image and text content), Playlist data: Genres (with musicians: name, description, images, introduction, records (title and link to MP3)) as well as Number of musicians, number of records per musicians, Boolean value if it is allowed to show website posts.

```
{
  "radioName": "Folk Music Audio Program",
  "musicianCaption": "Music Band: ",
  "recordCaption": "Song: ",
  "imgDuration": 5000,
  "logo": [ ...
],
  "warningData": [ ...
],
  "postData": [ ...
],
  "playlistData": [ ...
]
},
```

## ■ Get the data for a TV player from Radio Manager

- **Endpoint:** baseURL/wp-json/rm\_extended\_api/v1/radio/player\_data
- **Response:** Returns following data in JSON format: Radio name, Musician capture, Record capture, Image duration, Logo (with title, description and link to image).

```
"Folk Music Audio Program",
[
  {
    {
      "title": "For Your Listening Pleasure",
      "description": "This presentation is provided by Czech-American TV",
      "src": "https://catvusa.com/wp-content/uploads/2021/04/folk-music-audio-program-mraz-kotek.jpg"
    }
  },
  "Music Band: ",
  "Song: ",
  5000,

```

## ■ Get playlist data from Radio Manager

- **Endpoint:** baseURL/wp-json/rm\_extended\_api/v1/radio/playlist\_data
- **Response:** Returns following data in JSON format: genres (with musicians: name, description, images, introduction, records (title and link to MP3)), number of musicians, number of records per musicians, Boolean value if it is allowed to show website posts.

```
{
  "genres": [
    {
      "slug": "cultural",
      "musicians": [
        {
          "id": 1192139,
          "name": "Joe Mraz",
          "description": "<p><strong>Joe Mraz</strong></p>\n<p><img class=\"aligncenter size-full wp-image-119213 host-joe-mraz.jpg\" alt=\"\" width=\"1920\" height=\"1080\" /></p>\n<p>Joe Mraz was born in Chicago attended the Chicago Academy of Fine Art and the American Academy of Art in Chicago. After graduati now and lives in Longview, Texas. Where he teaches watercolor painting and conducts workshops for a comes from his grandparents who emigrated from Pilsen (Plzen) and Prague (Praha) during the late 18",
          "images": [
            {
              "title": "For Your Listening Pleasure",
              "description": "Click on the image to discover more about this host...",
              "src": "https://catvusa.com/wp-content/uploads/2021/04/cultural-audio-program-joe-mraz.jpg"
            }
          ],
          "introductions": [],
          "records": [
            {
              "title": "St. Nicholas",
              "src": "https://catvusa.com/wp-content/uploads/2021/04/joe-mraz-st-nicholas.mp3",
              "type": "audio"
            }
          ]
        }
      ]
    }
  ]
}
```





## ■ Get banners' data from Partner Banner Administration

- **Endpoint:** `baseURL/wp-json/pba_extended_api/v1/banners`
- **Response:** Returns banner's ID, partner's name, pages where banner should be located on the website, categories where banner should be located on the website, Boolean value showing if the banner should be located on all website's pages or not, image's attachment ID, image's URL.

```
{
  "id": "1198840",
  "name": "Partner Name Example",
  "description": "Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed eiusmod tempor inci
    ullamco laboris nisi ut aliquid ex ea commodi consequat.",
  "website": "https://examplewebsite.com",
  "pages": [],
  "img_at_id": "1198839",
  "img": "http://localhost/catvusa_wp/wp-content/uploads/2023/05/logoexample.png",
  "categories": [],
  "all_pages": "true"
},
```