

# Dokumentacija za Custom API Plugin za Wordpress

**Uvod:** Custom API Plugin je sofisticirani dodatak kreiran za Wordpress platformu. Njegova osnovna svrha je da omogući korisnicima da komuniciraju sa sajtom putem API interfejsa, dok u pozadini koristi specifičnu bazu podataka hostovanu na phpMyAdmin pod imenom `custom_api_gateway`.

Baza Podataka - Arhitektura i Upotreba: Baza `custom_api_gateway` sadrži tri ključne tabele:

**wp\_api\_logs:** Ova tabela igra centralnu ulogu u praćenju i dokumentovanju svih API zahteva koji dolaze na sajt. Svaki zahtev je pažljivo logovan, pružajući administratorima sajta mogućnost da u svakom trenutku pregledaju historiju zahteva putem Wordpress admin panela, tačnije putem menija "Custom API/ API logs".

**wp\_api\_settings:** Ova tabela služi kao konfiguracioni centar plugina. Omogućava administratorima da definišu, menjaju i brišu različite varijable zajedno sa njihovim imenima i vrednostima. Ove postavke se koriste za prilagođavanje rada plugina i njegove interakcije sa korisnicima, a sve postavke su lako dostupne na istoimenom meniju u Wordpress admin panelu.

**wp\_api\_users:** Kao što samo ime sugeriše, ova tabela je srž registracionog i verifikacionog procesa korisnika koji žele da koriste API. Implementira mehanizam registracije i autentikacije, garantujući da samo ovlašćeni korisnici mogu pristupiti API resursima.

Proces Registracije i Verifikacije: Kada korisnik poseti stranicu `localhost/wp/registracija`, postupak je sledeći:

- Unosi svoju email adresu.
- Nakon što potvrdi unos, sistema automatski generiše aktivacioni kod koji se potom šalje korisniku putem emaila.
- Uz ovaj kod, korisniku se dostavlja i specifičan verifikacioni link.
- Klikom na ovaj link, korisnik pokreće proces generisanja JWT (Json Web Token).
- Ovaj token se zatim šalje korisniku, koji ga može koristiti kao sredstvo autentikacije pri svakom API zahtevu.

**API Registracija za Wordpress Korisnike:** Za one korisnike koji već imaju nalog na Wordpress sajtu, postoji poseban mehanizam registracije putem API-a. Međutim, neophodan preduslov je posedovanje važećeg JWT.

**Deaktivacija i Čišćenje Podataka:** Jedna od ključnih karakteristika ovog plugina je pažljivo razmatranje deinstalacije. Prilikom deaktivacije plugina, administratorima je ponuđena opcija da odluče da li žele da trajno obrišu logove iz `wp_api_logs` tabele.

**Zaključak:** Custom API Plugin predstavlja robustno i sigurno rešenje za sve one koji žele da prošire funkcionalnosti svog Wordpress sajta putem API-ja. Njegova modularna struktura, uz kombinaciju sa visokim stepenom prilagodljivosti i sigurnosti, čini ga idealnim izborom za savremene web sajtove. Uz redovne ažuriranje i pridržavanje najboljih praksi, ovaj plugin će nesumnjivo unaprediti korisničko iskustvo i operativnu efikasnost svakog sajta.

# index.php

Metainformacije plugina: U samom početku fajla, navedene su metainformacije plugina koje obuhvataju:

- Naziv plugina: Custom API
- Opis: Plugin omogućava kreiranje članaka putem POST zahteva.
- Verzija: 1.0
- Autor: Lav IT Solutions

Provera Bezbednosti: Postoji bezbednosna provera koja osigurava da se PHP fajl ne pokreće direktno. Ako se fajl pokuša pokrenuti izvan Wordpress okruženja (bez `add_action` funkcije), ispisuje se poruka "Not allowed" i prekida se izvršavanje skripte.

Uključivanje Dodatnih Fajlova (Includes): Pet različitih PHP fajlova su uključeni, što znači da funkcije i klase iz tih fajlova postaju dostupne u `index.php` fajlu:

`api-functions.php` - Sadrži osnovne API funkcionalnosti.  
`admin-functions.php` - Funkcije administrativnog interfejsa plugina.  
`database-functions.php` - Funkcije povezane sa operacijama baze podataka.  
`registration-functions.php` - Sadrži funkcionalnosti vezane za registraciju korisnika putem API-ja.  
`verify.php` - Sadrži funkciju za verifikaciju korisnika.

Hooks (Kuke):

Kuke su jedan od ključnih aspekata Wordpress arhitekture. Omogućavaju dodavanje prilagođenih funkcija na specifična mesta ili tokom specifičnih događaja.

- API Hooks:
  - `rest_api_init`: Poziva funkciju `register_api_endpoints` prilikom inicijalizacije REST API-a, verovatno da registruje nove krajnje tačke (endpoints).

- `register_activation_hook`: Kada se plugin aktivira, pokreće se funkcija `logs_api_create_table` koja verovatno kreira tabelu za logovanje API zahteva.
- Admin Hooks:
  - Kuke poput `admin_menu` i `admin_init` koriste se za dodavanje stranica menija, podmenija, postavki i drugih administrativnih funkcionalnosti plugina.
  - `admin_enqueue_scripts` kuke se koriste za dodavanje specifičnih skripti i stilova na admin panel.
  - `wp_ajax_delete_custom_table` omogućava obradu AJAX zahteva koji je verovatno povezan s brisanjem određene tabele.
- Registracioni Hooks:
  - Funkcije kao što su `registracija_jwt`, `add_registration_page`, `register_query_vars` i slične, ukazuju na proces registracije, dodavanje registracione stranice, dodavanje nove stranice u meni i postavljanje šablona za stranicu.

---

Ovaj fajl je osnovni ulaz za Custom API plugin i služi kao centralno mesto za uključivanje svih drugih fajlova i registraciju akcija i filtera koji upravljaju kako plugin funkcioniše.

# admin-funtions.php

Ovaj PHP fajl `admin-functions.php` je deo "Custom API" WordPress plugina koji se nalazi u `custom-api/admin` direktorijumu, a ovaj fajl je specifično zadužen za funkcionalnosti koje su vezane za administratorski deo WordPress-a.

Evo detaljnog pregleda sadržaja ovog fajla:

Zavisnosti: Uključen je fajl koji se bavi prikazom logova

```
require_once plugin_dir_path __FILE__ . 'api-log-list-table.php';
```

## Podešavanja (Settings):

- Definicija glavne stranice menija za podešavanje Custom API-a.
- HTML forma za postavke koje omogućava administratorima da postavke neke osnovne postavke.
- Pohrana podešavanja: Ako je forma poslata, funkcija za pohranu postavki će biti pozvana. Ova funkcija proverava i pohranjuje podešavanja u bazu podataka.

## Podmeni za Logove:

- Kreira se podmeni za prikaz logova API-a.
- Funkcija `render_api_log_page` koristi klasu `API_Log_List_Table` za prikaz logova u tabelarnom formatu.

## Editovanje i Brisanje Logova:

- Funkcija za dodavanje skrivenih podmenija koji se koriste za editovanje ili brisanje logova.
- Funkcija za brisanje loga iz baze podataka.
- Funkcija za editovanje loga i ažuriranje informacija u bazi podataka.

## Deaktivacija Plugina i Brisanje Tabele:

- Kada administrator pokuša da deaktivira plugin, korisnik će videti JavaScript potvrdu.
- Ako administrator potvrdi deaktivaciju, AJAX poziv će obrisati tabelu koja je povezana sa ovim pluginom.

Ovaj fajl je suštinski zadužen za kreiranje administratorskog interfejsa za "Custom API" plugin, kao i za obradu postavki, logova i drugih administrativnih zadataka vezanih za rad sa pluginom.

# api-log-list-table.php

Ovaj fajl `api-log-list-table.php` predstavlja proširenje WordPress-ove klase `WP_List_Table` kako bi se omogućilo kreiranje tabele za prikaz logova API-a na administratorskoj stranici.

Ukratko, evo kako ovaj fajl funkcioniše:

Zavisnosti: Uključivanje osnovne `WP_List_Table` klase ako već nije učitana:

```
if ( ! class_exists( 'WP_List_Table' ) ) {  
    require_once ABSPATH . 'wp-admin/includes/class-wp-list-table.php';  
}
```

Definisanje Klase `API_Log_List_Table`:

- Ova klasa proširuje `WP_List_Table`, što omogućava kreiranje prilagođenih tabela u administratorskom delu.

Konstruktor:

- Postavlja osnovne informacije o tabeli kao što su singularno i množinsko ime stavke.

Definicija Kolona:

- `get_columns()`: Definiše koje će kolone biti prikazane u tabeli.

Priprema stavki:

- `prepare_items()`: Ova metoda koristi globalnu `$wpdb` varijablu kako bi preuzela sve logove iz baze podataka i postavila ih kao stavke za tabelu.

Prikazivanje podataka:

- `column_default()`: Ova metoda kontroliše kako će podaci biti prikazani za svaku kolonu. Na primer, za "actions" kolonu, ona poziva `column_actions()` kako bi generisala dugmadi za editovanje i brisanje.
- `column_actions()`: Generiše linkove za editovanje i brisanje za svaku stavku u tabeli.

U osnovi, ovaj fajl kreira tabelu koja omogućava administratorima da vide sve logove API-a i pruža im mogućnost da edituju ili brišu svaki log direktno iz tabele.

# api-functions.php

Ovaj `api-functions.php` fajl sadrži niz funkcija namenjenih za rad s WordPress REST API-jem. Pregled funkcionalnosti:

Registrowanje endpointa: (`localhost/wp-json/custom-api/`)

- `register_api_endpoints()`: Registruje nove API rute.
  - `custom-api/createpost`: Kreira novi članak.
  - `custom-api/register`: Registruje novog korisnika.
  - `custom-api/webhook`: Upravlja webhook zahtevima.

Kreiranje članka preko API-a:

- `create_clanak_from_post()`: Kreira članak na osnovu prosleđenih podataka. Proverava pristup putem funkcije `check_verification()`. Ako je sve u redu, pravi unos u bazu i vraća ID novog članka.

Logovanje API u bazu:

- `log_api_request()`: Sve API zahteve beleži u bazu podataka, kao što su endpoint, metod, user agent, IP adresa i datum i vreme.

Registrowanje korisnika preko API-a:

- `register_user()`: Ova funkcija omogućava registraciju korisnika putem API-ja. Proverava da li korisničko ime ili e-mail već postoje, ako ne, kreiraju se novi korisnik.

Webhook:

- `handle_webhook_request()`: Upravlja webhook zahtevima. Funkcija proverava da li je određeni webhook prepoznat na osnovu prosleđenog ID-a.

Provera verifikacije:

- `check_verification()`: Funkcija koja proverava validnost JWT tokena kako bi verifikovala korisnika. Ako je token validan i korisnik je aktiviran, funkcija vraća `true`, u suprotnom `false`.

U globalu, ovaj fajl omogućava interakciju s vašim WordPress sajtom putem REST API-ja za kreiranje članaka, registraciju korisnika, i upravljanje webhook zahtevima. On takođe pruža bezbednosne provere putem JWT tokena da bi se osiguralo da su zahtevi ovlašćeni.

# registration-functions.php

Fajl `registration-functions.php` sadrži skup funkcija namenjenih radu sa registracijom korisnika preko JWT tokena. Evo pregleda onoga što svaka funkcija radi:

## Učitavanje zavisnosti:

- Učitavaju se potrebne biblioteke: Firebase JWT za rad sa JWT tokenima i PHPMailer za slanje e-mailova.

## Generisanje JWT tokena:

- `generate_jwt_token($email)`: Ova funkcija generiše JWT token na osnovu prosledjene e-mail adrese. Taj token ističe za 1 sat.

## Generisanje aktivacionog koda:

- `generate_activation_code()`: Generiše se jedinstveni aktivacioni kod koristeći random bytes, koji se konvertuju u heksadecimalni format.

## Slanje e-maila:

- `send_email($email, $subject, $body)`: Ova funkcija koristi PHPMailer da bi poslala e-mail koristeći prosledjene parametre.

## JWT Registracija:

- `registracija_jwt()`: Ova funkcija upravlja registracijom korisnika. Ako korisnik sa datom e-mail adresom već postoji, ispisuje se poruka. U suprotnom, kreiran je aktivacioni kod, podaci se upisuju u bazu, a korisniku se šalje e-mail sa linkom za verifikaciju.

## Dodavanje stranice za registraciju:

- `add_registration_page()`: Dodaje pravilo za preusmeravanje kako bi kreirala custom stranicu za registraciju.
- `register_query_vars($vars)`: Registruje query varijable za preusmeravanje.
- `add_registration_page_to_menu()`: Dodaje "Registracija" stranicu u meni ako već ne postoji.
- `template_include($template)`: Ova funkcija kontroliše prikazivanje određenog šablona (u ovom slučaju stranice za registraciju) kada je određena query varijabla prisutna.

Ovaj fajl omogućava kreiranje i upravljanje registracionim procesom koristeći JWT i pruža funkcionalnost za slanje aktivacionih e-mailova koristeći PHPMailer. Također omogućava kreiranje i upravljanje custom stranicom za registraciju unutar WordPress-a.



# verify.php

Ovaj fajl, `verify.php`, postupa sa verifikacijom korisnika koji su se prethodno registrovali. Funkcionalnost je prilično direktna. Ovo je šta se događa korak po korak:

## Provera Parametara:

Skripta proverava da li su postavljene GET promenljive `code` i `id`. Ako jesu, skripta uzima te vrednosti za dalju obradu.

- `code`: aktivacioni kod koji je prethodno poslat korisniku na e-mail.
- `id`: jedinstveni identifikator korisnika u bazi podataka.

## Dohvatanje korisnika iz baze podataka:

- Skripta dohvata korisnika iz baze podataka prema ID-u.

## Provera korisnika i aktivacionog koda:

- Ako korisnik postoji, skripta proverava da li je već aktiviran.
- Ako korisnik nije aktiviran, proverava se da li se poslati aktivacioni kod podudara sa onim koji je sačuvan u bazi podataka za tog korisnika.
- Ako se kodovi podudaraju, korisnik se smatra verifikovanim.

## Ažuriranje statusa korisnika:

- Ukoliko je korisnik uspešno verifikovan, njegov status "activated" se postavlja na 1.

## Generisanje i slanje JWT tokena:

- Po uspešnoj verifikaciji, korisniku se generiše JWT token koristeći funkciju `generate_jwt_token` koja je definisana u prethodno navedenom fajlu `registration.functions.php`.
- Ovaj token se potom čuva u bazi podataka za tog korisnika.
- JWT token se zatim šalje korisniku na e-mail koristeći funkciju `send_email`.

## Ispis informacija:

- Zavisno o tome da li je korisnik uspešno verifikovan, da li je već bio verifikovan, ili ako postoji problem sa aktivacionim kodom, odgovarajuća poruka se ispisuje korisniku.

U globalu, ova skripta omogućava korisnicima da verifikuju svoje registracije koristeći aktivacione kodove koje su primili na e-mail, a zatim, nakon uspešne verifikacije, primaju JWT token koji mogu koristiti za autentifikaciju na web sajtu ili aplikaciji.

# database-funtions.php

Ovaj fajl, `database-functions.php`, sadrži dve funkcije koje upravljaju kreiranjem i brisanjem tabele u bazi podataka za praćenje logova API-a. Evo pregleda kako funkcije rade:

## `logs_api_create_table:`

Ova funkcija kreira tabelu `api_logs` za praćenje logova API-a ako ona već ne postoji.

- Koristi globalni `$wpdb` objekat za interakciju s WordPress bazom podataka.
- Proverava da li tabela već postoji koristeći SQL upit `SHOW TABLES LIKE`.
- Ako tabela ne postoji, kreira se nova tabela sa sledećim kolonama:
  - `id`: Primarni ključ, automatski se povećava.
  - `api_endpoint`: Endpoint ili ruta API-ja koji je pozvan.
  - `api_method`: HTTP metoda (npr. GET, POST) korišćena prilikom poziva.
  - `user_agent`: Informacija o klijentu koji je poslao zahtev.
  - `ip_address`: IP adresa klijenta.
  - `request_data`: Podaci koji su poslani u zahtevu.
  - `timestamp`: Vremenska oznaka kada je zahtev poslat.
- Funkcija koristi WordPress funkciju `dbDelta` koja je optimalna za kreiranje i ažuriranje tabela jer može automatski prepoznati promene u strukturi i primeniti ih.

## `logs_api_drop_table:`

Ova funkcija briše tabelu `api_logs` ako ona postoji.

- Ponovo, koristi se globalni `$wpdb` objekat.
- Proverava da li tabela postoji kao i u prethodnoj funkciji.
- Ako postoji, koristi SQL upit `DROP TABLE` da obriše tabelu.
- Ako brisanje nije uspelo, informacija o grešci se beleži u logove pomoću `error_log` funkcije. Ukoliko je uspelo, broj redova koji su pogođeni (u ovom slučaju, broj obrisanih tabela) beleži se takođe u logove.

Ove funkcije mogu biti korisne prilikom aktiviranja i deaktiviranja određenog plugina. Na primer, kada se plugin aktivira, možete automatski kreirati potrebne tabele koristeći `logs_api_create_table`, a kada se deaktivira, možete koristiti `logs_api_drop_table` da obrišete tabele koje nisu više potrebne.

## confirm-deactivation.js

Ovaj JavaScript fajl, `confirm-deactivation.js`, povezan je sa brisanjem tabele prilikom deaktivacije plugin-a i koristi jQuery za interakciju s korisnikom. Evo kako funkcioniše:

`document.ready`: Ova funkcija se poziva kada se HTML dokument potpuno učita, tako da će svi elementi HTML-a biti dostupni za manipulaciju.

`$('.delete-custom-plugin').on('click', function(e))`: Ovaj deo koda dodeljuje funkcionalnost kliku na sve elemente sa klasom `delete-custom-plugin`. To je verovatno klasa koja se dodaje linku za deaktivaciju plugin-a.

`e.preventDefault()`: Ova linija sprečava podrazumevano ponašanje linka kada se klikne. Umesto toga, koristi se JavaScript za kontrolu događaja.

`var deactivationUrl = $(this).attr('href')`: Ova linija uzima vrednost atributa `href` elementa na koji je kliknuto, verovatno linka za deaktivaciju plugin-a. Ova URL adresa će biti korišćena kasnije za preusmeravanje korisnika nakon brisanja tabele ili deaktivacije plugin-a.

`var confirmed = confirm("Da li želite da obrišete tabelu?")`: Ova linija prikazuje dijalog za potvrdu sa porukom "Da li želite da obrišete tabelu?" i dva dugmeta, "OK" i "Cancel". Ako korisnik pritisne "OK", vraća se `true`, a ako pritisne "Cancel", vraća se `false`. Vrednost se smešta u promenljivu `confirmed`.

`if (confirmed)`: Ovaj blok koda proverava vrednost `confirmed`. Ako je `true` (korisnik je potvrdio brisanje), šalje AJAX zahtev serveru za brisanje tabele.

- `$.post(custom_vars.ajax_url, {action: "delete_custom_table"}, function(response))`: Ovde se koristi jQuery `$.post` funkcija za slanje POST zahteva serveru.
- `custom_vars.ajax_url` je URL adresa servera koju je prethodno definisao WordPress za rukovanje AJAX zahtevima.
- `{action: "delete_custom_table"}` je objekat koji se šalje serveru i sadrži informaciju o akciji koja se zahteva, u ovom slučaju, brisanje tabele.
- Funkcija `function(response)` je callback funkcija koja će se izvršiti nakon što server obradi zahtev. U ovom slučaju, nakon uspešnog brisanja tabele, korisnika se preusmerava na URL `deactivationUrl` (link za deaktivaciju plugin-a).

`else`: Ako korisnik odabere "Cancel", onda će biti preusmeren na URL `deactivationUrl` bez brisanja tabele.

Ovaj kod omogućava korisnicima da brzo i jednostavno obrišu tabelu prilikom deaktivacije plugin-a, uz dodatnu potvrdu radi sprečavanja slučajnog brisanja.

# registration-page.php

Ovo je HTML kod za registracionu stranicu (`registration-page.php`) koja se koristi za prikazivanje forme za registraciju. Ovde je šta svaki deo ovog HTML koda radi:

`<!DOCTYPE html>`: Ovo je deklaracija tipa dokumenta i označava da se radi o HTML5 dokumentu.

`<html lang="en">`: Ovo označava početak HTML dokumenta i postavlja jezik na engleski.

`<head>`: Ovaj deo HTML-a sadrži metapodatke i informacije o stranici koje se ne prikazuju na samoj stranici, već su za internu upotrebu. Na primer, ovo podešava karakterni skup i naslov stranice.

- `<meta charset="UTF-8">`: Postavlja karakterni skup stranice na UTF-8, koji podržava različite znakove.
- `<meta name="viewport" content="width=device-width, initial-scale=1.0">`: Ovo podešava prikaz na mobilnim uređajima tako da se pravilno skalira i učitava.
- `<title>Registracija</title>`: Postavlja naslov stranice na "Registracija", koji će se pojaviti u tabu pregledača.

`<body>`: Ovo je deo stranice koji sadrži vidljive elemente koji će se prikazati korisnicima.

- `<h1>Registracija</h1>`: Prikazuje naslov "Registracija" na stranici.
- `<form method="post" action="">`: Ovde počinje HTML forma koja omogućava korisnicima da unesu podatke i pošalju ih na server. Forma koristi HTTP POST zahtev (`method="post"`) i trenutno je postavljena da šalje podatke na istu stranicu (`action=""`). Kasnije se na ovoj stranici rukuje registracijom.
  - `<label for="email">Email:</label>`: Prikazuje oznaku za unos email adrese.
  - `<input type="email" name="email" id="email" required>`: Polje za unos email adrese. `type="email"` omogućava pregledaču da proveri da li je unesena email adresa u ispravnom formatu, a `required` označava da je ovo polje obavezno za popunjavanje.
  - `<input type="submit" value="Registruj se">`: Dugme za slanje forme sa natpisom "Registruj se". Kada korisnik klikne na ovo dugme, forma će se poslati na server za registraciju.

Ovaj HTML kod predstavlja osnovni okvir registracione stranice sa poljem za unos email adrese i dugmetom za registraciju. Nakon što korisnik popuni formu i klikne na "Registruj se", server će obraditi registraciju na osnovu unetih podataka.

U zaključku, kombinovana dokumentacija koja obuhvata niz PHP fajlova i HTML stranica za WordPress dodatak nazvan "Custom API" pruža sveobuhvatno rešenje za kreiranje i konfigurisanje API-ja, proces registracije korisnika i upravljanje API zahtevima i logovima. Ovaj dodatak omogućava pristup kroz HTTP zahteve i JWT autentifikaciju korisnika.

Ovde je sažetak ključnih komponenata ovog WordPress dodatka:

PHP Fajlovi: Ovaj dodatak sastoji se od više PHP fajlova koji obavljaju različite funkcionalnosti.

- **index.php**: Glavni fajl dodatka koji definiše osnovne informacije o dodatku i postavlja akcione tačke za API registraciju i administraciju.
- **admin-functions.php**: Fajl koji sadrži funkcionalnosti za administraciju dodatkom, uključujući i postavljanje stranica za podešavanje.
- **api-functions.php**: Definiše API tačke i funkcionalnosti za kreiranje članaka putem POST zahteva i registrovanje korisnika.
- **registration-functions.php**: Sadrži funkcije za generisanje JWT tokena, slanje verifikacionih emailova, i proces registracije.
- **verify.php**: Obrada verifikacionog koda nakon što korisnik klikne na link za verifikaciju.
- **database-functions.php**: Definiše funkcije za kreiranje i brisanje tabela za logovanje API zahteva.

HTML Stranica za Registraciju: "**registration-page.php**" je HTML stranica koja prikazuje formu za unos email adrese korisnika koji želi da se registruje. Nakon unosa email adrese i slanja forme, korisnik će dobiti verifikacioni link na email adresi za potvrdu registracije.

JavaScript za Deaktivaciju: "**confirm-deactivation.js**" sadrži JavaScript kod koji se koristi za potvrdu deaktivacije dodatka i brisanje tabela u bazi podataka pre nego što se dodatak deaktivira.

Ovaj dodatak za WordPress omogućava administraciju i pristup API-ju, proces registracije korisnika, verifikaciju email adresa, i beleženje API zahteva u bazi podataka. Uz dodatnu konfiguraciju i integraciju u postojeću WordPress instalaciju, ovaj dodatak može biti koristan alat za upravljanje API-jem i registracijom korisnika.

