

Rešavanje CSS / JS problema

Last updated April 15, 2021

Možete da pokrenete optimizaciju u LiteSpeed Cache dodatku za WordPress i otkrijete da se vaš sajt više ne prikazuje ispravno. To je čest problem, ali verovatno nije sa dodatkom LSCache. Umesto toga, verovatnije je da postoji problem sa jednom CSS ili JavaScript datotekom koja je u sukobu sa postupkom kompresovanja ili kombinovanja.

Isprobajte sledeće korake da biste precizno utvrdili CSS i JavaScript greške:

- Isključite optimizaciju na WordPress kontrolnoj **LiteSpeed Cache > Page Optimization**. Kliknite na karticu **CSS Settings** i podesite sve funkcije CSS optimizacije na **OFF**. Kliknite na karticu **JS Settings** i podesite sve funkcije JS optimizacije na **OFF**.
- Očistite keš na putanji LiteSpeed Cache > Toolbox > Purge i pritisnite dugme **Purge All**.
- Ponovo učitaite stranicu.

Ako stranica i dalje izgleda loše, vaš problem nije u LSCache dodatku, već u vašem sadržaju postoji JavaScript ili CSS datoteka koja nije kompatibilna sa nekim funkcijama za optimizaciju LSCache dodatka.

Kako pronaći uzrok CSS/JS problema

Pre nego što išta promenite uverite se da je zaista problem do optimizacije. Isključite sve funkcije optimizacije, očistite keš memoriju i proverite sajt.

Ako i dalje izgleda loše onda problem nije u funkcijama LSCache-a dodatka za optimizaciju. Ako iznova uključite LSCache, a stranica je ispravna, onda neka druga karakteristika LSCache-a “kvari” vaš sajt. Ako je prikaz sajta neispravan čak i nakon isključivanja LSCache dodatka, onda je problem povezan sa nečim drugim na vašem sajtu.

Ukoliko stranica izgleda dobro, onda ste utvrdili da se problem krije negde u funkcijama za optimizaciju. Sada imate dve mogućnosti:

1. Isključite CSS i JS opcije za optimizaciju i ostavite te opcije optimizacije zauvek isključene. Ovo je jednostavno rešenje, a ako se odlučite na ovo, vaš sajt i dalje može imati velike koristi od keširanja čak i bez omogućene optimizacije.

2. Ponovo uključite sve opcije, pronađite problematične datoteke i izuzmite ih iz optimizacije. Za ovu opciju ispratite sledeće korake opisane u nastavku.

Prvi korak u pronalaženju problematične datoteke je utvrđivanje da li tražite JavaScript ili CSS problem.

Prvo proveravamo JavaScript:

- Ponovo omogućite funkcije optimizacije povezane sa JavaScript, a CSS funkcije ugasite,
- Očistite keš memoriju,
- Osvežite stranicu. Ako je neispravna, onda je jedna od vaših JavaScript datoteka problematična. Ako je u redu, onda su i vaše JavaScript datoteke u redu.

Ako JavaScript nije krivac, onda je to verovatno CSS. Najbolji način da to proverite je:

- Onemogućite funkcije optimizacije povezane sa JS i ponovo omogućite CSS funkcije,
- Očistite keš memoriju,
- Osvežite stranicu,
- Ako je neispravna, onda je jedna od vaših od vaših CSS datoteka problematična.

Sada kada znate da li je CSS ili JS, moraćete da pronađete određenu datoteku koja uzrokuje problem.

Detektovanje datoteke koja uzrokuje problem

Zarad primera, pretpostavimo da imate CSS datoteku koja izaziva probleme. Ako je u slučaju problem sa JavaScriptom, i dalje možete slediti ove iste korake.

- Da bismo pronašli datoteku koja je problematična, prvo moramo dobiti listu svih mogućih CSS (ili JS) datoteka. Kad se to završi, izuzećemo čitav spisak datoteka iz optimizacije, a zatim ćemo ih ponovo uvoditi jednu po jednu dok ne pronađemo krivca.
- Pomoću konzole u pretraživaču otvorite **Network** karticu i kliknite **CSS** da biste videli samo CSS datoteke.
- Ponovo učitajte stranicu. Trebali biste videti listu svih CSS datoteka koje ta stranica koristi, navedene u koloni **Name**.
- Izbacite sve sa ove liste iz optimizacije. (Datoteke možete izuzeti punom adresom ili delimičnom)

Ako je potrebno da vidite punu putanju bilo koje CSS datoteke, kliknite na ime te datoteke i kliknite na Headers. Pronaći ćete punu putanju izabrane datoteke koja je navedena kao URL adresa zahteva.

Testirajte listu datoteka

1. Omogućite sve funkcije optimizacije koje želite da koristite
2. Izuzmite sve CSS datoteke iz optimizacije. Na putanji LiteSpeed **Cache > Page Optimization > Tuning Settings** unesite datoteke sa liste, jednu po liniji, u polje **CSS Excludes**.
3. Očistite keš memoriju i osvežite sajt. Trebalo bi da se pravilno prikazuje.
4. Uklonite jednu od CSS datoteka sa exclude liste, očistite keš memoriju i osvežite stranicu
5. Proverite sajt
 1. Ako i dalje izgleda u redu, datoteka koju ste upravo uklonili nije problematična.
 2. Ako stranica deluje neispravno, onda znate da ste pronašli problematičnu CSS datoteku. Vratite je u Exclude CSS polje.

Na kraju ovog procesa kada vaš sajt izgleda uredno, treba uključiti sve funkcije optimizacije koje želite da koristite i trebalo bi da imate jednu ili više problematičnih CSS ili JS datoteka koje se nalaze u okviru CSS Excludes ili JS Excludes.

Drugi problemi

Brzo punjenje prostora na disku

Neki dodaci mogu generisati CSS ili JavaScript datoteke. Ako je omogućen **CSS Combine** ili **JS Combine**, prisustvo takvih datoteka uzrokuje stvaranje nove kombinovane datoteke za svaku stranicu na sajtu. Ako sajt ima zaseban prikaz za mobilne uređaje, tada će biti napravljene dve kombinovane datoteke za svaku stranicu na sajtu. Ovo nije idealno i ima potencijal da brzo popuni vaš prostor.

Da biste zaustavili ovo ponašanje, morate izuzeti nasumični CSS ili JS iz kombinovanja.

Pronađite i isključite generisanu datoteku na sledeći način:

- Posetite vaš sajt dodajući `/?LSCWP_CTRL=before_optm` nakon naziva domena (npr `example.com/?LSCWP_CTRL=before_optm`), u dva odvojena pretraživača ili prozora u režimu bez arhiviranja (Private ili Incognito).

- Pogledajte **Page Source** u svakom prozoru
- Kopirajte i nalepite kod na sajt za proveru razlika (poput <https://www.diffchecker.com/>) i uporedite
- Trebalo bi da vidite nekoliko redova istaknutih CSS-a ili JavaScript datoteka, zapravo, razliku između učitavanja dve stranice, koja ukazuje gde su nasumično generisane datoteke.
- Pretpostavlja se da će niz imati zajednički deo (na primer, sa abc_12345 i abc_67890, zajednički deo je abc_). Otkrijte šta je zajednički deo za vašu stranicu i unesite zajednički deo slučajnog niza u **CSS Excludes** ili **JS Excludes** po potrebi.
- Purge All CSS/JS

CSS se ne učitava ispravno nakon ažuriranja

Najverovatnije ovo nije problem LSCache dodatka, jer on ne kešira statičke datoteke.

Ako CSS od teme nije ispravno učitavan nakon ažuriranja, proverite keš memoriju pretraživača. Proverite da li imate CDN omogućen ili obrnuti proksi ispred svog servera, poput Cloudflare-a. Možda će biti potrebno očistiti ove mehanizme za keširanje.

Critical CSS se ne učitava

Prvo i najvažnije, uverite se da su sve IP adrese stavljene na belu listu na vašem serveru, u vaše WordPress sigurnosne dodatke (ako ih ima) ili u druge bezbednosne uređaje na vašem sajtu, poput CDN/Application Layer Firewall. IP adrese koje bi trebalo biti na beloj listi možete videti [ovde](#). Ako i nakon toga CSS i dalje ne radi kako treba, proverite sledeće:

- Proverite da li u direktorijumu wp-content/litespeed/ccss/ ima CSS datoteka i da li su generisane CSS datoteke. Datoteke treba generisati unutar tog direktorijuma za svaki od tipova postova na vašem sajtu.
- Ako CSS datoteke ne postoje ili folder CCSS ne postoji, sačekajte da se izvrši cron zadatak. On će generisati CCSS sa naših Cloud servera.
- Ako CSS datoteke postoje i valjane su, izvršite opciju **Purge All – LSCache**. Novi CSS ne bi bio uključen u starije keširane stranice.
- Ako CSS datoteke sadrže ccss-timeout, to znači da je isteklo vreme komunikacije sa vašim serverom kada smo pokušali da generišemo Critical CSS. U ovom slučaju, ručno izbrišite tu datoteku i očistite LSCache kako bi se ista ponovo generisala.
- Ako CSS datoteke sadrže sintaksnu grešku, onda jedna od vaših CSS datoteka sadrži grešku. Potragu za pogrešnom CSS datotekom možete suziti na sledeći način:
 - Isključite **CSS Combine and Minify**

- Kliknite na **Purge All** i **Purge Critical CSS**
- Pokušajte da ponovo generišete CCSS. Sledeća CSS datoteka generisana unutar wp-content/litespeed/ccss/ treba da prikaže stvarnu CSS datoteku koja ima sintaksnu grešku
- Ispravite sintaksnu grešku u navedenoj CSS datoteci i kliknite na **Purge Critical CSS**.
- Ponovo ponovo omogućite CSS Minify i Combine.

“FOUC sa Critical CSS” opcija je i dalje uključena

Omogućavanje funkcije **Generate Critical CSS (CCSS)** trebalo bi da eliminiše **Flash of Unstyled Content (FOUC)**, ali ga ponekad možete i dalje videti, iako je podešavanje uključeno.

Kada je **Load CSS Asynchronously** uključen, CSS vašeg sajta će se učitavati istovremeno kad i HTML kod. Dakle, sav sadržaj koji se učitava pre odgovarajućeg CSS-a biće prikazan bez stila. Cilj funkcije Generate Critical CSS je da zaustavi ovakav ishod. On ubacuje osnovna pravila CSS stila ugrađena u HTML stranice, tako da se ta pravila izvršavaju pre učitavanja samog sadržaja.

Ukoliko primetite nestilizovan sadržaj, čak i kada je Generate Critical CSS uključen ozrok može biti da CCSS nije umetnut inline u HTML stranice.

Kada se ovakva stranica prikaže, proverite izvorni kod. Videćete nešto slično ovome:

```
<style id="litespeed-optm-css-rules"></style>
```

Oznake `<style>` su tu, ali su prazne, što znači da je CCSS omogućen, ali pravilo još uvek nije generisano ili umetnuto na stranicu.

Mogući uzroci:

Potrebno je nekoliko sekundi za generisanje kritičnog CSS-a, a ako ste uključili **Generate Critical CSS in the Background**, LSCache dodaje stranicu u “red” zasnovan na cron-u, tako da se CCSS može generisati kasnije. Kao rezultat, može se dogoditi da se stranica učita pre nego što CCSS postane dostupan, što dovodi do nestilizovanog prikaza.

Imajte na umu da iste vrste stranica dele isti CCSS. Na primer, ako imate post1, post2 i post3, kada pristupite postu1, generisaće se CCSS. Tada će CCSS biti tamo i spreman za upotrebu kada se pristupi post2 i post3.

Rešenja:

Najjednostavnije rešenje je isključiti **Generate Critical CSS in the Background**.

Generiranje CCSS-a u prvom planu zahteva da prvi posetilac sačeka nekoliko sekundi, ali eliminiše problem nestilizovanog sadržaja. Kao alternativu, isprobajte **Purge All – LSCache** nakon generisanja CCSS pravila. Ovo će omogućiti LSCacheu da umetne CCSS na stranice koje su već bile keširane pre nego što je CCSS generisan.

Critical CSS se ne generiše

Ako ste potvrdili da CCSS nije uspeo da se generiše, to je možda zbog sintaksne greške. Postoji nekoliko načina da se takve greške potraže u vašem CSS-u.

Metoda br. 1 : QUIC.cloud kontrolna tabla

Ako ste povezali svoj sajt sa QUIC.cloud servisom, možete da proverite na nadzornoj tabli QUIC.cloud servisa. Kartica CCSS i nudi upozorenja u odeljku **Recent Requests**.

Metoda br. 2 : Vaš sistem datoteka

Ako svoj sajt niste povezali sa QUIC.cloud, možete direktno proveriti sistem datoteka. Dođite do /wp-content/litespeed/ccss pomoću FTP-a ili File Managera na kontrolnoj tabli. Pogledajte unutar CSS datoteka u tom direktorijumu. Jedan od njih treba da sadrži sintaksnu grešku.

Metoda br. 3: HTML izvorni kod

I na kraju, možete da proverite HTML izvorni kod, mada ovaj metod nije toliko pouzdan kao prethodna dva.

Pronađite originalni CSS fajl

Bez obzira na metod koji ste koristili, verovatno ste otkrili nešto poput: /*CssSyntaxError: /wp-content/litespeed/cssjs/5aaa0.css:1:1: Unclosed block*/ Datoteka koja uzrokuje problem je /wp-content/litespeed/cssjs/5aaa0.css , ali ovo nije originalna CSS datoteka. Bilo koja datoteka koju pronađete u direktorijumu /wp-content/litespeed/ je ona koju je generisao dodatak LiteSpeed Cache.

U ovom slučaju, to je optimizovana CSS datoteka koja se generiše kroz funkcije CSS Combine i/ili CSS Minify. Ne možemo da popravimo sintaksnu grešku u generisanoj

datoteci.

Da biste pronašli izvorni CSS i ispravili sintaksnu grešku, moraćete da isključite optimizaciju i izvršite ponovnu proveru.

- Idite na **Page Optimization > CSS** i isključite CSS Combine i CSS Minify. Sačuvajte promene
- Idite na Toolbox > Purge i izvršite **Purge All** i **Purge All – CCSS**
- Posetite sajt tako da se CCSS ponovo generiše
- Koristite bilo koji metod koji ste koristili prvi put da biste ponovo proverili CCSS i potražili sintaksnu grešku.

Ako niste sigurni odakle potiče CSS, naziv datoteke treba da vam da nagoveštaj. Ako je iz dodatka, putanja bi trebalo da izgleda poput /wp-content/plugins/plugin-name/path/file.css. CSS tema treba da izgleda otprilike kao /wp-content/themes/theme-name/path/file.css.

Zaobilazak optimazicije U AJAX-u

Ako imate konflikt i treba da zaobiđete funkcije za optimizaciju u AJAX-u, možete dodati sledeći kod u okviru fajla functions.php vezan za vašu temu: `defined( 'DOING_AJAX' ) && add_filter( 'litespeed_can_optm', '__return_false' );`, ili možete pozvati odgovarajući filter kada koristite AJAX.

DevTools Failed to Load SourceMap

Konzola u Google Chrome pretraživaču može prikazati poruku poput sledeće:

```
DevTools failed to load SourceMap: Could not load content for https://domain.com/wp-content/litespeed/cssjs/xxxx.js.map: HTTP error: status code 404 , net ::  
ERR_HTTP_RESPONSE_CODE_FAILURE
```

Ovo je apsolutno normalno. Alat pokušava da učitava mapu resursa za kompresovane JS ili CSS datoteke, tako da ona može biti umanjena. Naravno, ova datoteka ne postoji, pa vraća grešku 404.