

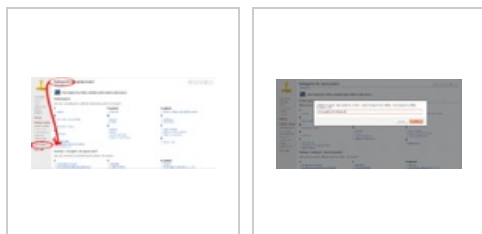
MediaWiki:CategoryIntersection.js/dokumentace

Skript MediaWiki:CategoryIntersection.js je pomůcka umožňující zobrazit seznam stránek, které jsou zároveň obsaženy v několika zadaných kategoriích. Nebere ohled na jmenné prostory, takže zobrazuje i takové stránky, u nichž je v jedné kategorii obsažena diskuzní stránka a v jiné zakategorizován samotný článek.

Návod

Tuto pomůcku mohou využít redaktori. (Uživatelům, kteří nemají redaktorská práva se pomůcka nenačte.) Postup je následující:

1. Načtete stránku v jmenném prostoru „*Kategorie:*“ (např. Kategorie:Ke zpracování), od níž chcete hledat průnik s jinou kategorií.
2. V levém menu v sekci „*Redakční nástroje*“ klikněte na odkaz „*Průnik kategorií*“.
3. Zadejte název kategorie, s níž chcete hledat průnik. Název zadávejte bez jmenného prostoru (tj. např. pouze „*Chirurgie*“, „*Vnitřní lékařství*“,...). Je možno zadat i více kategorií najednou, pak se zadávají bez mezer oddělené svislíkem (např. „*Chirurgie|Vnitřní lékařství*“).
4. Počkejte, až se výpočet provede. Čas zpracování se odvíjí od rychlosti Vašeho připojení a velikosti zadaných kategorií. Pokud zpracováváte kategorie s velkým počtem článků, může celý proces trvat celkem dlouho.
5. Po zpracování se Vám otevře nové okno s odkazy na nalezené články.
6. Užijte si výsledek!



Na stránce kategorie klikněte na odkaz „Průnik kategorií“ v levém menu.

Zadejte kategorie, se kterými se má hledat průnik.

Technická dokumentace

Všechny funkce jsou hojně vybaveny komentáři, které vysvětlují jejich funkci i jejich jednotlivé kroky.

Celý běh skriptu probíhá asi takto:

1. Skript se načítá pomocí funkce `importScript()` na stránce `MediaWiki:UpravyMenu.js`, pokud je uživatel ve skupině *Redaktor* a načtená stránka je ve jmenném prostoru `14 („Kategorie:“)`.
2. Přidá se do levého menu odkaz, který po kliknutí spouští funkci `categoryIntersection()`.
3. Vytvoří se zároveň objekt `ciObject`, do kterého se ukládají všechny odpovědi API.
4. Po kliknutí na odkaz se spustí funkce `categoryIntersection()`.
 - a) Ta pomocí funkce `ciOverlayStatusOn()` vytvoří překryvný element stránky, do kterého se bude zapisovat, co se zrovna děje, aby uživatel byl v obraze a věděl, že má čekat.
 - b) Také pro každou kategorii volá funkci `ciAsyncGetCategoryList()`, která je jádrem komunikace s API.
5. Funkce `ciAsyncGetCategoryList()` podává žádosti API o data a spouští se rekurzivně, dokud nejsou všechna data získána. Data se ukládají do objektu `ciObject` stejně jako informace o tom, jsou-li již kompletní. V takovém případě se volá funkce `ciFinal()`. Uživatel je o činnosti informován pomocí funkce `ciOverlayStatusUpdate()`.
6. Funkce `ciFinal()` kontroluje, zdali jsou kompletní všechny žádosti pro všechny kategorie. Je-li tomu tak, volá funkci `ciCountIntersection()`, `ciOverlayStatusOff()` a `ciOutput()`.
 - a) `ciCountIntersection()` je jádrem zpracování získaných dat. Vytvoří si pomocnou proměnnou jako pole polí a následně pro každou položku v jedné kategorii kontroluje, zdali je obsažena i v těch ostatních. Tím se získá průnik kategorií.
 - b) `ciOverlayStatusOff()` skryje překryvný element, do něhož se vypisovalo, co se děje, protože již není potřeba a jen by překážel.
 - c) `ciOutput()` vytvoří nové okno, do kterého se vypíší výsledky, a to jak v podobě odkazů, tak v podobě wikikódu.

Komunikace s API se v JavaScriptu potýká s problémem, že se žádosti o informace posílají asynchronně a API odpovídá s limitovaným počtem výsledků. (Pro uživatele bez bot flagu je maximální limit výsledků v jedné odpovědi pro `action=query&list=categorymembers` 500.) Problém je řešen rekurzivním voláním funkce `ciAsyncGetCategoryList()`, která se volá do té doby, než jsou všechny výsledky pro danou kategorii vyčerpány.

Výsledky se ukládají do globální proměnné `ciObject`, kam se také ukládá informace o tom, zdali získané výsledky pro danou kategorii jsou již kompletní. Pokud ano, zkontroluje se, zda jsou kompletní pro všechny zadané kategorie a v takovém případě se tedy přistoupí ke zpracování výsledků. Pokud daná kategorie není ještě kompletní, vyšle se žádost API o další data. Pokud je kategorie kompletní, ale ostatní ještě ne, nedělá funkce `ciAsyncGetCategoryList()` nic, tj. vlastně čeká, až pro jiné kategorie přijdou odpovědi API, které pak samy spustí zpracování výsledků.