POI Add-on

Verrijk woning pagina's met points of interest

- Introductie
- Admin
- Intergraties
- Afstanden berekenen
- Types

Introductie

Met deze addon kun je verschillende POI's ophalen op basis van locatiegegevens. Een POI bevat (o.a.) de volgende data punten: - Naam - Locatie (latitude/longitude) - Openingstijden* - Foto's - Categorieen - Adresgegevens

* Niet alle POI's bevatten deze informatie.

Op de achtergrond wordt gebruik gemaakt van de Google Places API. Een API key is daarom ook benodigd en kan ingevuld worden in het instellingenscherm van de plugin.

Er zijn twee manieren om POI informatie op te halen: handmatig en d.m.v. een AJAX hook. De laatste optie geniet de voorkeur, omdat deze automatisch caching toepast.

Admin

Deze sectie van de documentatie is bedoeld voor het gebruik van de plugin vanuit de WordPress omgeving.

Dashboard

Na het installeren en activeren van de plugin kan je onder het menu item "Makelaar" in het sub-menu "POI Addon" vinden. Wanneer je hier naar toe navigeert eindig je op het dashboard van de plugin. Via het dashboard kan je de licentie, Google Places API en de cache beheren.

Licentie

Om de functionaliteiten van de plugin te activeren moet een licentie code toegevoegd worden. Op aanschaf van de plugin krijg je deze toegestuurd. Als alternatief kan je ook op

https://app.tussendoor.nl/ inloggen. Binnen deze omgeving kan je al je diensten bij ons beheren. Je logt hierop in met hetzelfde email adres waar de plugin mee is aangeschaft. Mocht je hier nog niet eerder op hebben ingelogd dan kan je van het wachtwoord vergeten traject gebruik maken om een wachtwoord in te stellen.

API

De plugin maakt gebruik van de diensten van de Google Places API. Zonder werkende API zal deze Addon niet functioneren. Mocht je problemen ervaren met het ophalen van POI's, dan is een incorrect ingestelde API key het meest voorkomende probleem.

Cache

Wanneer je gebruik maakt van een AJAX integratie (lees meer op integraties) dan worden aanvragen gecached. Dit betekend dat een aanvraag maar één keer plaats hoeft te vinden voor elk object. Via het 'Geavanceerd' tabblad van de plugin kan je deze cache ook handmatig legen.

Intergraties

Deze sectie van de documentatie is bedoeld voor het uitleggen van de verschillende manieren van integreren.

AJAX integratie

De plugin biedt een AJAX hook aan, waarmee POI informatie opgehaald kan worden. Deze hook verwacht drie paramters: - Types - Latitude - Longitude

De vierde, optionele parameter request_type geeft aan of alle types los van elkaar opgehaald moeten worden (separate) of in één query (compound). Hou er rekening mee dat request_type: separate er voor zorgt dat elk type meetelt in het aantal requests richting Google.

Een voorbeeld AJAX request via jQuery voor het ophalen van informatie ziet er als volgt uit:

```
jQuery.post(ajaxurl, {
    action: 'locatorCache',
    type: ['restaurant', 'supermarket', 'gym'],
    request_type: 'compound',
    latitude: '53.194885',
    longitude: '5.800822',
}, function(response) {
    if (response.status === false) {
        return console.warn("Unable to resolve places: the API did not respond or an invalid request was made.");
    }

    var poi;
    for (var i = response.data.length - 1; i >= 0; i--) {
        poi = response.data[i];
        // Latitude van de POI: poi.geometry.location.lat;
        // Longitude van de POI: poi.geometry.location.lng;
        // Naam van de POI: poi.name;
    }
});
```

Let op: de variabele `ajaxurl` is alleen gedefinieerd in het back-end van WordPress. Op de front-end zul je deze variabele zelf beschikbaar moeten aanmaken. Je kan hiervoor gebruiken maken van de `wp_localize_script` van WordPress. Meer informatie hierover vind je op

https://developer.wordpress.org/reference/functions/wp_localize_script/.

Informatie uit deze hook wordt automatisch gecached. De response (`response.data`) bevat een key `cache_hit` om aan te geven of de informatie uit de cache gehaald kon worden of dat de Google API

geraadpleegd moest worden.

Handmatige integratie

Deze manier van integreren wordt uitgevoerd in PHP op de server. Deze functionaliteit past standaard geen caching toe, waardoor elke request opnieuw via Google verloopt. De parameters zijn nagenoeg gelijk met de AJAX hook, namelijk: - Types - Latitude - Longitude

```
use Tussendoor\Locator\CacheController;

try {
    $cacheController = new CacheController();
    $pois = $cacheController->getPlacesFromApi(
        ['bicycle_store', 'book_store', 'clothing_store'],
        '53.194885',
        '5.800822'
    );
} catch (Exception $e) {
    // Er trad een fout op tijdens de API request
}

if ($pois === false) {
    // Er is een fout opgetreden, bijv. de API key is niet ingevuld
}

foreach ($pois as $poi) {
    $poiLocation = $poi['geometry']['location'];
    // $poiLocation['lat']
    // $poiLocation['lng']
    $poiName = $poi['name'];
    // ... etc
}
```

Afstanden berekenen

Het is mogelijk om de afstand uit te rekenen van de POI ten opzichte van de woning. Voor deze berekening is de `geometry` library nodig, welke beschikbaar is via de Google Maps API. Let op dat deze standaard niet ingeladen wordt. Raadpleeg de documentatie van Google Maps hoe je deze inlaadt: https://developers.google.com/maps/documentation/javascript/libraries.

```
function calculateDistance(poi, baseLocation)
{
    if (typeof google.maps.geometry === 'undefined') {
        console.warn("The distance cannot be calculated, the Geometry library in Google Maps is not loaded.");

        return false;
    }

    // Calculate the distance of the place to the main object.
    var fullDistance = google.maps.geometry.spherical.computeDistanceBetween(
        new google.maps.LatLng(poi.lat, poi.lng),
        new google.maps.LatLng(baseLocation.lat, baseLocation.lng)
    );
}

console.log(calculateDistance(
    {lat: '53.194284', lng: '5.799072'},
    {lat:'53.194885', lng:'5.800822'}
));
```

Types

Alle POI's zijn ingedeeld in 'types'. Kort door de bocht zijn dit de type POI's die opgehaald kunnen worden. Op moment van schrijven zijn de volgende types beschikbaar: - restaurant - supermarket - school - park - library - gym - bus_station - train_station - light_rail_station - bicycle_store - book_store - clothing_store - convenience_store - department_store - ... etc.

Voor een volledig overzicht, zie de documentatie van Google:

https://developers.google.com/places/web-service/supported_types.

De plugin bundelt een aantal van deze types onder één categorie, zodat het zoeken vereenvoudigd wordt. De volgende categorieen zijn beschikbaar: - restaurants - supermarkets - stores - schools - park - library - gym - busstations - trainstations - lightrailstations

De categorieen (en de onderliggende types) zijn als volgt op te halen:

```
use Tussendoor\Locator\App;

$categories = App::get('services');
// Bijvoorbeeld: alle 'stores' types:
$storeTypes = $categories['stores']['name'];
// ['bicycle_store', 'book_store', 'clothing_store',
// 'convenience_store', 'department_store', 'electronics_store',
// 'florist', ...etc. ]
```