

API Documentation | Cookiedatabase.org | Dev License

Content

1. Function to Call API
2. Get service types
 - a. Add language parameter
3. Get purposes
4. Get cookie information
 - a. Adding license
 - b. stdClass Object
5. Sync services

1.Function to call API

```
/**
 * Example function for the cookiedatabase.org API
 */
define('MY_PREFIX_COOKIEDATABASE_URL',
'https://cookiedatabase.org/wp-json/cookiedatabase/');
```

2.Get service types

```
/**
 * Get list of service types
 * This data should be cached, as it's not often changed.
 *
 * @param string $language //en, nl, de
 *
 * @return array
 */

function get_servicetypes_options( $language ) {
    $endpoint = trailingslashit( MY_PREFIX_COOKIEDATABASE_URL )
                . 'v1/servicetypes/' . $language;

    $response = wp_remote_get( $endpoint );
    $status   = wp_remote_retrieve_response_code( $response );

    $body     = wp_remote_retrieve_body( $response );
    if ( $status == 200 ) {
        $body         = json_decode( $body );
        $serviceTypes = $body->data;
    }

    return $serviceTypes;
}
```

3.Get purposes

```
/**
 * get list of purposes for cookies
 * This data should be cached, as it's not often changed.
 *
 * @param string $language //en, nl, de etc.
 *
 * @return array
 */

function get_cookiepurpose_options( $language ) {

    $endpoint = trailingslashit( MY_PREFIX_COOKIEDATABASE_URL )
                . 'v1/cookiepurposes/' . $language;
    $response = wp_remote_get( $endpoint );
    $status   = wp_remote_retrieve_response_code( $response );
    $body     = wp_remote_retrieve_body( $response );
    if ( $status == 200 ) {
        $body           = json_decode( $body );
        $cookiePurposes = $body->data;
    }

    return $cookiePurposes;
}
```

4. Get Cookie information

```
/**
 * get cookie info from cookiedatabase.org
 *
 * Cookies are searches fuzzy: wp_12335 will return wp_*
 */

function get_cookie_info() {
    $data = array(
        'en' => array
            (
                'Facebook' => array(
                    '_js_datr',
                    'actppresence',
                    '_fbc'
                ),
                'Complianz' => array
                    (
                        'cmlpz_choice',
                        'cmlpz_id',
                        'cmlpz_stats',
                    ),
            ),
    ),
}
```

```

/*
* By adding a list of third-party cookies you can make sure it's only
matched on third party cookies, not on first party cookies.
* When the cookie list is returned, you'll get a list of services, with
information if it's third party or not.
*/

```

```

        'thirdpartyCookies' => Array
        (
            '_js_datr',
            'actppresence',
            '_fbc'
        )
    );

$data['cdb_license'] = '123456789';

$json
    = json_encode( $data );

```

```

$endpoint
    = trailingslashit(MY_PREFIX_COOKIEDATABASE_URL ) .
    'v1/cookies/';

```

```

$ch = curl_init();
curl_setopt( $ch, CURLOPT_URL, $endpoint );
curl_setopt( $ch, CURLOPT_CUSTOMREQUEST, "POST" );
curl_setopt( $ch, CURLOPT_POST, 1 );
curl_setopt( $ch, CURLOPT_POSTFIELDS, $json );
curl_setopt( $ch, CURLOPT_RETURNTRANSFER, 1 );
curl_setopt( $ch, CURLOPT_HTTPHEADER, array(
    'Content-Type: application/json',
    'Content-Length: ' . strlen( $json )
)
);
$result = curl_exec( $ch );
$error = ( $result == 0
    && strpos( $result,

```

```

'<title>502 Bad Gateway</title>' ) === false ) ? false

```

```

        : true;
    if ( $error ) {
        $msg = "Could not connect to cookiedatabase.org";
    }

    curl_close( $ch );

    $result = json_decode( $result );
//cookie creation also searches fuzzy, so we can now change the cookie name
to an asterisk value
    //on updates it will still match.
    if ( isset( $result->data->error ) ) {
        $error_msg = $result->data->error;
    } else {
        $result = $result->data;
    }
    /*
stdClass Object
    (
        [en] => stdClass Object
        (
            [_js_datr] => stdClass Object
            (

```

```

                [name] => _js_datr
                [type] =>
    [typeID] =>
    [retention] => 2 years
                [unique] =>
    [retentionID] => 38
    [serviceID] => 301
    [service] => Facebook
                [collectedPersonalData] =>
    [collectedPersonalDataID] =>
    [cookieFunction] => store user preferences
                [cookieFunctionID] => 158
    [purpose] => Preferences
                [purposeID] => 1257
    [message] =>
    [isPersonalData] =>
    [isMembersOnly] =>
    [ignore] => 1
    [complete] =>
    [language] => en
                [backgroundInfo] =>
    [slug] => _js_datr
                [submittedByAPI] =>
    )
    [complianz_consent_status] => stdClass Object
        (
                [name] => complianz_consent_status
                [type] =>
    [typeID] =>
    [retention] => 1 year
                [unique] =>
    [retentionID] => 36
    [serviceID] => 66
    [service] => Complianz
                [collectedPersonalData] => none
                [collectedPersonalDataID] => 165
    [cookieFunction] => store cookie consent preferences
                [cookieFunctionID] => 148
    [purpose] => Functional
                [purposeID] => 61
    [message] =>
    [isPersonalData] =>
    [isMembersOnly] =>
    [ignore] =>
    [complete] => 1
    [language] => en
                [backgroundInfo] =>
    [slug] => complianz_consent_status
                [submittedByAPI] => 1
        )
    )
    [headers] => stdClass Object
        (
                [Location] => https://cookiedatabase.org
        )
    [status] => 200
    )
}
*/

```

5.Sync Services

```
/**
    * Sync all services
    */

function get_services() {
    $data = array
        (
            'en' => array
                (
                    'Google Analytics',
                    'Facebook',
                    'WordPress',
                    'Complianz',
                ),
            'cdb_license' => '123456'
        );
    $json = json_encode( $data );
    $endpoint = trailingslashit( MY_PREFIX_COOKIEDATABASE_URL )
        . 'v1/services/';

    $ch = curl_init();

    curl_setopt( $ch, CURLOPT_URL, $endpoint );
    curl_setopt( $ch, CURLOPT_CUSTOMREQUEST, "POST" );
    curl_setopt( $ch, CURLOPT_POST, 1 );
    curl_setopt( $ch, CURLOPT_POSTFIELDS, $json );
    curl_setopt( $ch, CURLOPT_RETURNTRANSFER, 1 );
    curl_setopt( $ch, CURLOPT_HTTPHEADER, array(
        'Content-Type: application/json',
        'Content-Length: ' . strlen( $json )
    )
    );

    $result = curl_exec( $ch );

    $error = ( $result == 0
        && strpos( $result,
            '<title>502 Bad Gateway</title>' ) === false ) ? false
        : true;
    if ( $error ) {
        $msg = "Could not connect to cookiedatabase.org";
    }

    curl_close( $ch );

    $result = json_decode( $result );
    //cookie creation also searches fuzzy, so we can now change the cookie name
    //to an asterisk value
    //on updates it will still match.
    if ( isset( $result->error ) ) {
        $msg = $result->error;
    } else {
        $result = $result->data;
    }
}

/*
```

```

* The resulting data returns a list of services with information
                                                                    'ID' => id of
the service on CDB.org
    'name' => name of the service
    'serviceType' => service type
    'serviceTypeID' => service type id
    'sharesData' => if the service shares data
with a third party
    'thirdParty' => if it is a third party
service (shares data, and places cookie on the third party domain)
    'secondParty' => if it is a second party
(shares data, but places cookie on your website's domain)
    'privacyStatementURL' =>
https://policies.google.com/privacy
    'language' => en
    'slug' => google-analytics
Each third party service is returned with a
list of cookies connected with it.
    *
    stdClass Object
    (
        'en' => stdClass Object
    (
        'Google Analytics' => stdClass Object
    (
        'service' => stdClass Object
            (
                'ID' => 294
                'name' => Google Analytics
                'serviceType' => website statistics
                'serviceTypeID' => 162
                'sharesData' => 1
                'thirdParty' =>
                'secondParty' => 1
                'privacyStatementURL' =>
https://policies.google.com/privacy
                'language' => en
                'slug' => google-analytics
            )
        )
    )
    'Facebook' => stdClass Object
    (
        'service' => stdClass Object
            (
                'ID' => 301
                'name' => Facebook
                'serviceType' => display of recent social posts
and/or social share buttons
                'serviceTypeID' => 384
                'sharesData' => 1
                'thirdParty' => 1
                'secondParty' =>
                'privacyStatementURL' =>
https://www.facebook.com/policy/cookies
                'language' => en
                'slug' => facebook
            )
        )
    'cookies' => Array
    (
        '_js_datr',

```

```

        'actppresence',
        '_fbc',
        'fbm*',
    )
)
'Complianz' => stdClass Object
(
    'service' => stdClass Object
    (
        'ID' => 66,
        'name' => 'Complianz',
        'serviceType' => 'cookie consent management',
        'serviceTypeID' => 67
        'sharesData' => 0
        'thirdParty' =>
        'secondParty' =>
        'privacyStatementURL' =>
'https://complianz.io/privacy-statement/',
        'language' => 'en',
        'slug' => 'complianz',
    )
)
)
'headers' => stdClass Object
(
    'Location' => 'https://cookiedatabase.org',
)
'status' => 200
}
*/

```