Osclass software

CAR ATTRIBUTES PRO PLUGIN

Welcome to documentation for Car attributes PRO plugin. Primary function of plugin is to provide features to build solid car attributes in your classifieds.

Please note that it is required to have basic knowledge of Osclass and understand it’s functionality. Please find bellow basic FAQ.

Where is Osclass documentation?
https://docs.osclasspoint.com/

How to translate plugin?
https://docs.osclasspoint.com/translating-and-editing-language-files-po-and-mo

I need help, what to do?
https://forums.osclasspoint.com/general-plugins-discussion/

Overview

Plugin is build on several building blocks:

- Vehicle type
- Section
- Make
- Model
- Property
- Property values

Please follow this order when setting & configuring plugin.

Car attributes PRO plugin comes with default data for vehicle types, makes, models, properties and property values, mainly for vehicle types “personal cars”, “motorcycles”. This content is only for demonstration purpose, as configuration of plugin might be very hard without demo data.

There is no guarantee or support on sample data. If it does not match to your needs, please create own.

Installation

For installation without default/basic data, please first edit index.php of plugin and set constant CAP_INSTALL_BASIC_DATA on line 18 to 0:

```
define('CAP_INSTALL_BASIC_DATA', 0);
```

Installation will now not include basic data.

> **Warning:** There is known bug in basic data – property values IDs must be unique. In basic data, for integer values, this is achieved incorrectly by adding value key to ID. Ie: for value 2800 (kg), if it already exists, new value is added as 2800-123, where 123 is key of value. This leads to problems on property edit page and you might not be able to save property without fixing this issue – as ID for integer values must be number only.
>
> **Solution:** use +- value around one you wanted, or define just values for “all vehicle types”. I.e. use 2799 or 2801 as unique identifier.

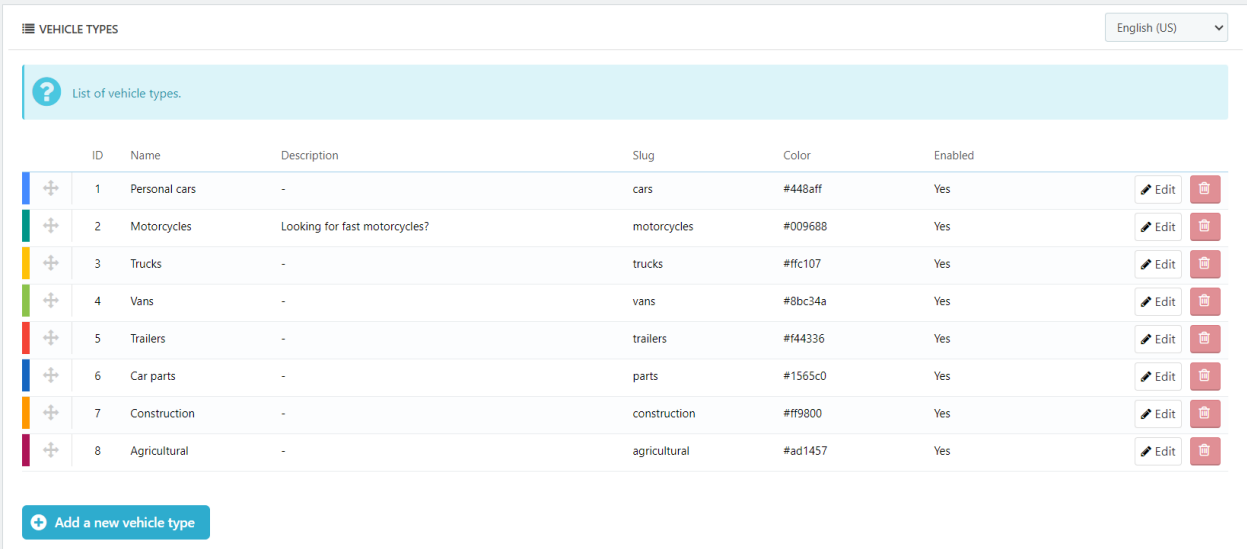
Basic terms

Identifier / slug / unique ID – is not meant to be primary key. Identifier is string used in URLs as replacement for primary keys. This usually brings better SEO and is more user friendly. As example, having vehicle type “Cars” with identifier “cars” and primary key 1, using identifier URL will look like …/cap-type,cars/… instead of …/cap-type,1/…

Identifier cannot be translated, slug can be translated. Length is usually 3-50 characters. These fields cannot start with number and can letters!

Vehicle type

Vehicle type is highest level and Makes, Models and Properties are assigned to specific Vehicle type.



Vehicle type must be linked with one or more Osclass categories. Based on category selection, vehicle types are shown on publish page. On the other side, assigned categories and their children are shown in search widget (if category included).

Configuration specific info:

Description – is more as a SEO text, shown inside of search widget.

Slug – only for vehicle types, slug can be translated. Slug is used in URL instead of primary key.

Icon – you can place SVG (html), image link/url or font awesome class (ie. fa fa-ball). FA class depends on font awesome used in your theme. Plugin does not come with font awesome.

Color – used in vehicle types navigation menu

Categories – select one or more. In ideal case each category belongs to only 1 vehicle type, but plugin does not put any restriction on this.

Vehicle type is also property, but of special type and it's values are no defined on property edit page.

Category restrictions

Category restrictions are purely related to vehicle type settings. In search hook, based on category, vehicle types are found, first one is selected (as default) and properties for this vehicle type are selected.

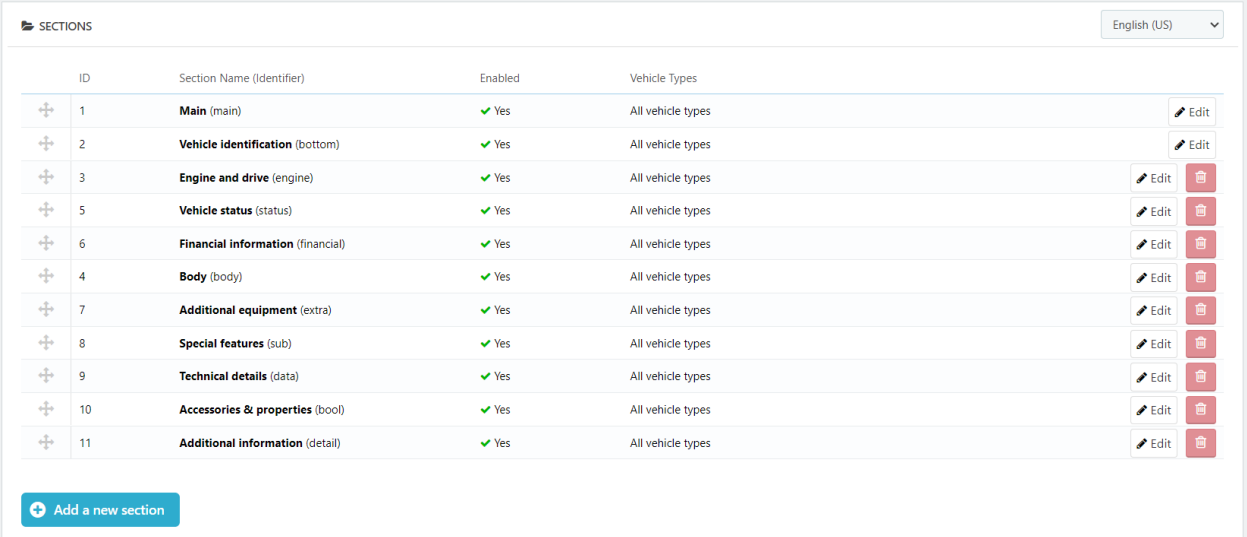
Section

Sections categorize/group properties in widget, item page and publish page. As example you can create "Engine" section that will group all engine related properties.

Section can be visible in one or all locations (item, publish, widget) and can be also restricted by vehicle type. Means it's possible to create different sections for different vehicle types.

When configuring section, consider enabling "Enable Empty Sections" option in configuration page, otherwise your sections will not be visible until you assign attributes to it.

Warning: If you assign property to specific section, this property will be visible just in those vehicle types enabled in section & property at same time!



“Main” section is primary and cannot be removed or disabled!

Configuration specific info:

Name – section name.

Identifier – currently not really used in plugin, but required.

Description – shown in widget section button as short section summary.

Help text (search) – used as a header of section in widget. If not defined, name is used. Set to “-“ to disable header of this section.

Help text (publish) – used as a header of section on publish page. If not defined, name is used. Set to “-“ to disable header of this section.

Help text (item) – used as a header of section on item detail page. If not defined, name is used. Set to “-“ to disable header of this section.

Icon – SVG, font awesome class or image URL. Used just in widget.

Vehicle types – enable or disable in specified vehicle types.

Section type – select in what locations is section visible – publish page, widget, item page.

Properties Style on Item page – select style for properties shown in this section. There are many different styles those can be used.

Widget Style – select how properties are visible on widget

- Box – properties are placed directly to widget
- Button – properties are hidden under button and visible after clicking on button in modal

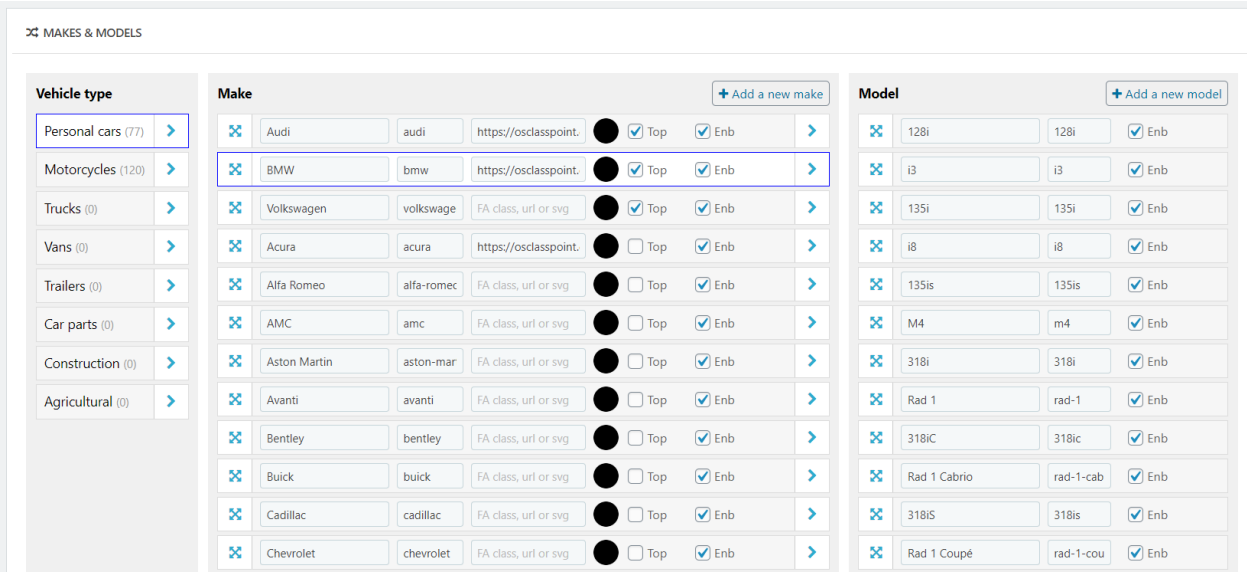
Enable – quickly enable/disable section in front. Keep in mind when section is disabled, assigned properties are disabled as well.

Order – position in front.

Make

Each make can belong to one and only one vehicle type. Based on vehicle type selection, makes and models are reloaded.

There is no limit how many makes you can define, but we do not recommend to add more than 200 makes per vehicle type.



Configuration specific info:

Name – cannot be translated.

Slug – used in URL, must be unique, cannot be translated.

Icon – same as for vehicle types, use SVG, font awesome or image link.

Color – currently not utilized in plugin.

Top – specify if you want to show this make in “top makes” list – at top of select box.

Model

Each make can have models assigned. Each model belongs to one and only one make. Models are reloaded based on makes selection.

There is no limit how many models you can define, but we do not recommend to add more than 500 models per make.

Configuration specific info:

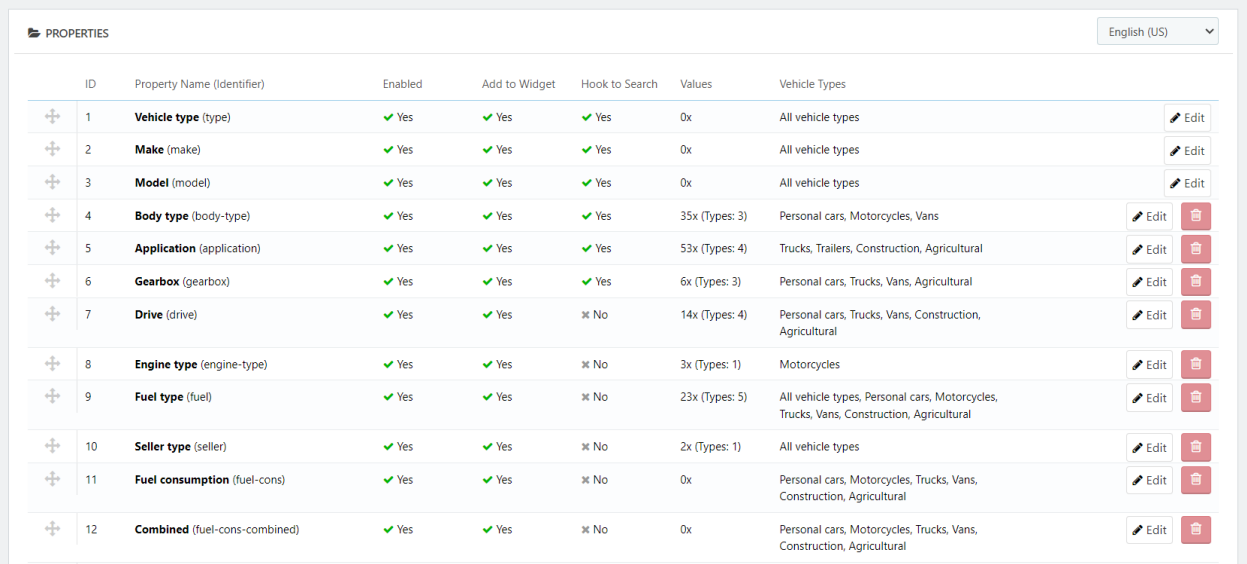
Name – cannot be translated.

Slug – used in URL, must be unique, cannot be translated.

Property

Property is considered as any attribute of vehicle (listing). It might be color, engine size, if vehicle is new, or it is on leasing.

Settings of properties relates and you will not be able to select some property parameters for specific types, or selection will be restricted.



> **Warning**: There is no limit on number of properties defined by plugin, however there is hard limit set by maximum number of columns in one table. When you reach it, you will not be able to create any more properties until you remove some. Hard limit by MySQL is 4096 columns, but real limit will be much lower. You should not have more than 500 properties at all!

Configuration specific info:

Identifier – used in URL instead of primary key. Must be unique. Based on identifier, database column is created (using data type as well). You can only use a-z, 0-9 and dash in identifier. Cannot be translated.

Identifier cannot be changed for reserved properties (vehicle type, make, model, category, location, pattern). These also cannot be removed.

> **Warning:** If you change identifier of property, it's going to drop old column and add new column for item attribute values. It means it will drop all existing values for this property defined on items!

Name – used as label and placeholder (if not defined). Can be translated.

Description – use as hint/tip next to label name. Use this field to better explain what property means, in case it's not clear. It's not recommended to define description to every single field, even those that are self-explanatory (like Color). Can be translated.

Help Text (Search) – text used in placeholder of input/textarea or first select option on search & widget. Can be translated.

Help Text (Publish) – text used in placeholder of input/textarea or first select option on publish page. Can be translated.

Icon – as usual, SVG, font awesome class or image link/url.If not specified, default icon (asterisk) may be used on item page.

Property type – how the values of property are selected/entered by user (primarily on publish page):

- Select from predefined: select box, check box, radio,…
- User define value: input, textarea
- Group other properties: Property is just wrapper for other properties

  Property groups: Use this type to wrap/group other properties together. As example fuel consumption in city, outside city and combined would be grouped together under "fuel consumption" property. Group parent will behave just as a wrapper & header and no value can be defined for it.

Default style – based on property type, select default layout for property. It can be select box, checkbox, slider, input, textarea… List of values is restricted based on previous selection.

Data type - how the values are stored.

- Select box - integers (ie year, number of doors, gears,…) where user can select value from list or define own (in search) or you may use just values list, where user can only select from predefined values (like color selection).
- Input/textarea – Integer for whole numbers (ie airbags count), decimal (ie fuel consumption, always DECIMAL(10,3)) or varchar for string values (ie VIN or serial number)

Multi-select in Search/Widget – use to allow users in search to select more than one value. Note that on publish page, all properties are single-select only.

Range-select in Search/Widget – use to generate from & to select boxes/inputs for this property in search. Make sense for properties like mileage where it does not make sense to search by specific number, but by range.

Opposite name – for switch type, where you have property with name “Rent” just as Boolean and you want opposite value of switch to be “Sale”.

Measure unit – km for mileage, kg for weight properties etc. Do not use more than 3 characters, will not be nice on mobiles.

Required on publish – check if this field should be required field on publish page.

Vehicle types – select to which vehicle types is this property related. It will be visible just under selected vehicle types.

Search engine – for user defined values, you can use equal or like engines.

Enable – you may quickly disable property from front without it’s removal.

Item detail section – select in which section property value will be shown on item page. It often means also layout used to show property.

Add to widget – check if you want to include this property in search widget.

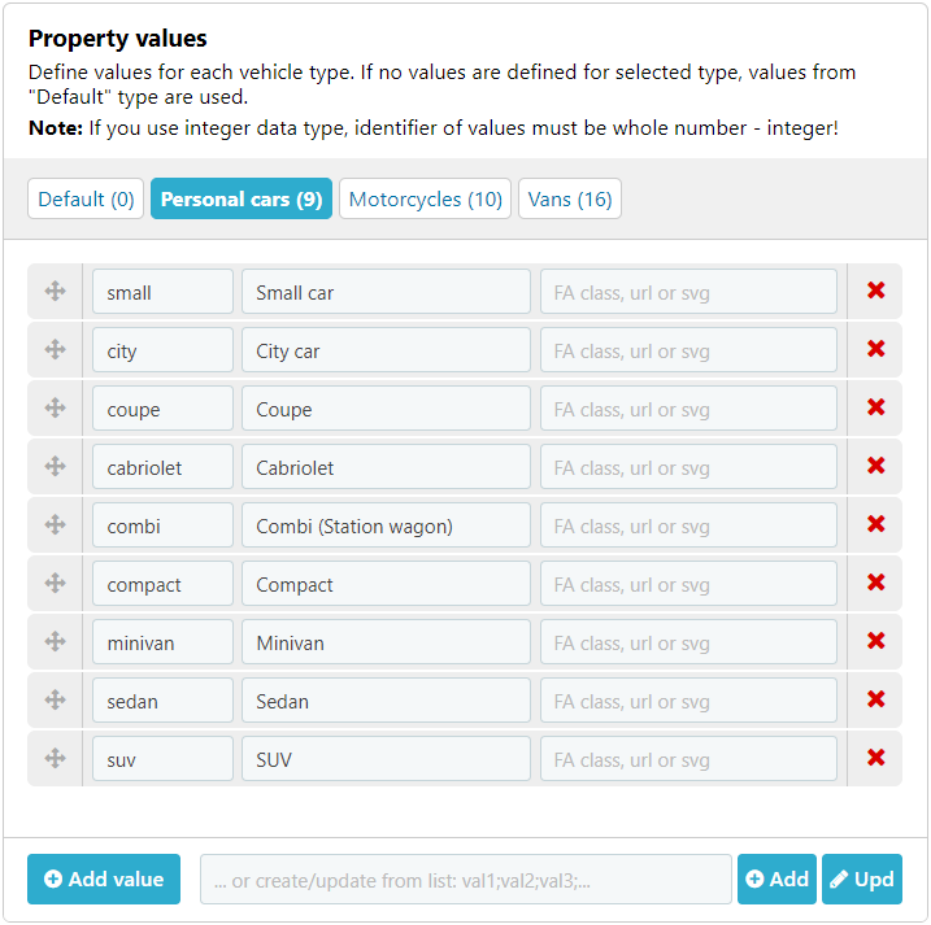
Maximize box size – for some properties, you might want to make them full-width if there is space for that. You can use this to enlarge property box.

Widget section – select in which section property is shown.

Add to search hook – check if you want to include property in search_form hook on search page.

Property values

Property values cannot be defined for reserved properties (vehicle type, make, model, category, location, pattern). These has own pages to setup values.



Values are defined per vehicle type.

First you may define “default” values used by more than one vehicle types (or all, if not defined).

Then you can specify different values for specific vehicle types.

If values are defined for specific vehicle type, only these are used. Default are not used or merged!

Configuration specific info:

Identifier – used in URLs, must be unique across whole property and there will be check done. Cannot be translated. Item attribute values use primary keys to store user selection so changing identifier does not impact existing data.

Name – can be translated.

Icon – SVG, image url or font awesome class.

Values can be added one by one or from list. If you added values from list and want to update different languages in same way, you can use “Upd” button that will change names on existing property values.

> **Warning**: Number of values is not limited, but intention to provide this feature is to predefine list of few values! We strictly recommend to not have more than 100 values per property.
>
> **Warning**: In order to limit number of queries to database, plugin preload all data and reuse it (this limit number of queries from thousands to 10). This means if you use too many properties and properties values, dataset could be large and your site slower. Make sure to have enough of available memory in PHP and avoid creating hundreds of properties with hundreds of values just because you saw it somewhere else as a good practice!

Configuration

Majority of parameters is directly on vehicle type, make, model or property, but there are still several important global configuration parameters.

Color – default color used by plugin. Synchronize with your theme color.

Select box type – user custom select box type with rich features or HTML box type. For HTML box type, multi-select is not available even specified on property.

Close values list after select – for custom select box type, when user select it’s value, values list is automatically closed.

Hide select if no values – ie if you enabled property in specific vehicle type, but forgot to define values, use this to hide property. Otherwise user see property, but cannot select anything.

Hide submit button – hide button in values list that says “OK” and close values list.

Disable search hook – no matter what is setting on property, plugin will not add any fields to search hook (search_form).

Enable more filters button – add “more filters” button to search hook. Clicking on button will bring search widget in modal. Keep in mind this widget has certain limitations and may be better to bring search widget externally to search page.

Required validation type – type of validation to be used on publish page for required attributes. jQuery use validator script (used by themes by default) and HTML5 is just about adding required attribute to input/select/textarea.

Hook form to publish page – always on until you do custom integration into publish/edit page.

Item detail attributes hook – by default hooked to item_detail, is list of all item attributes.

Item attributes hook – function to show attributes widget/compact box with top 5-10 attributes. This should be used in item loop.

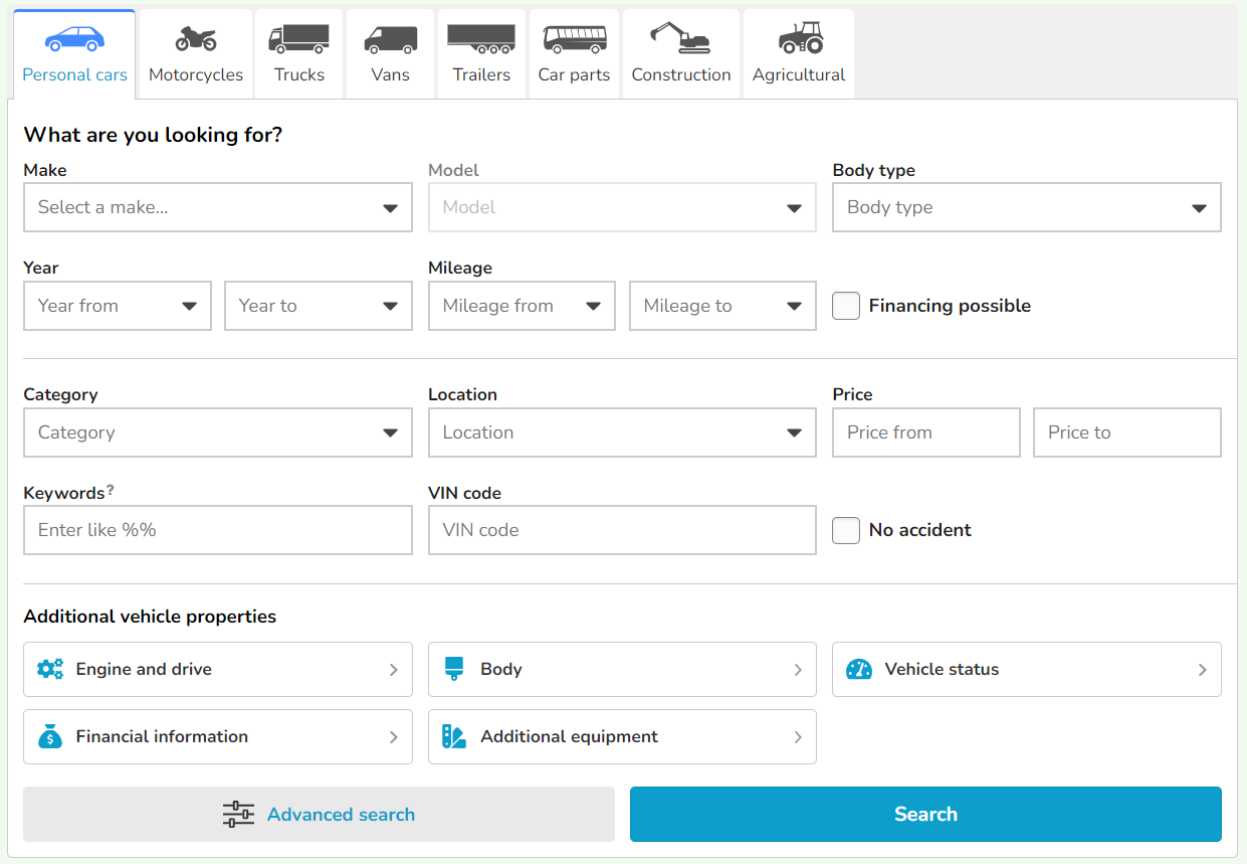
Enable showcase – show list of functions and their layout on item detail page. Helps to understand how to do custom integration of different plugin functions.

Search widget hooks – hook search widget (big) box.

Vehicle types widget hooks – hook vehicle types list/carousel/box.

Widget settings

Default settings for widget. When you are placing widget manually to theme, you may define own options in $options parameter and rewrite these settings.



Search widget design – specify how much attributes are shown by default:

- Simple – just few properties
- Medium – more properties
- Complex – all properties

Search widget menu design – design of vehicle types on top of widget:

- Simple – just list with horizontal scroll (when required)
- Advanced – first XY vehicle types is shown and others are hidden under “more” button/menu

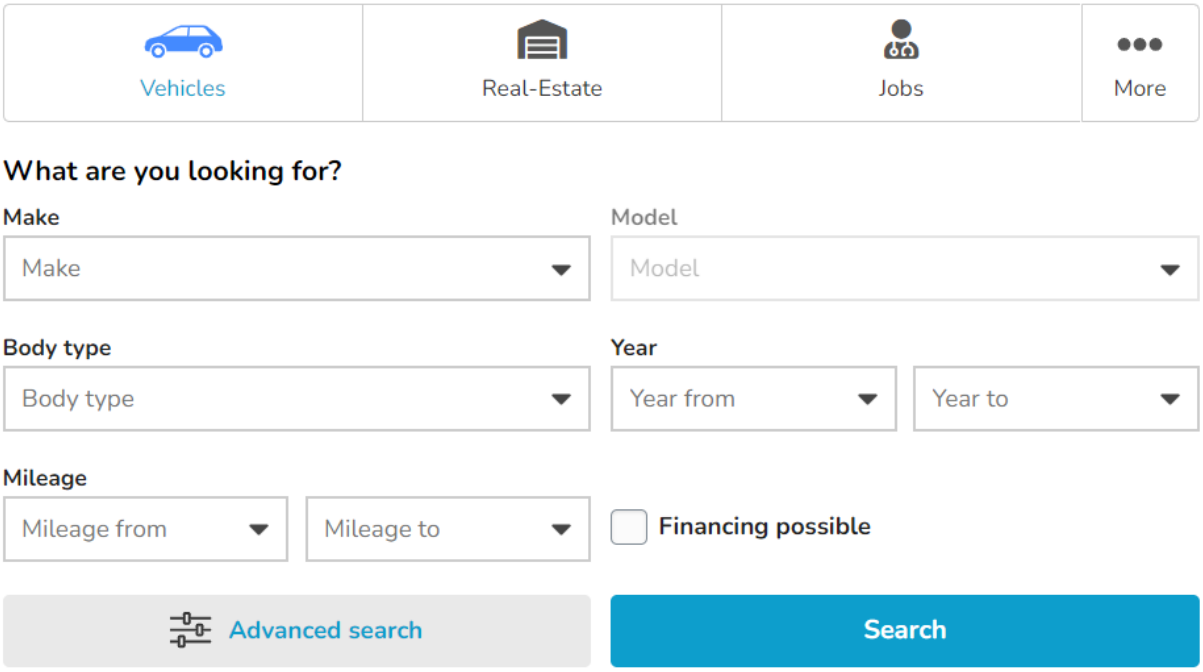
Search widget menu items design – for advanced menu design, if vehicle type elements are condensed (more shown on screen) or shown in standard way (3 visible, others under “more” menu).

Search widget style – borderless layout is more suitable if you want to directly integrate widget into existing design/theme. Boxed layout works better if borderless design does not look so good.

Search widget width – select large if widget is going to be placed in box with width larger than 700px. Otherwise use medium (if box is smaller).

Use not just as “Car attributes”

If you need to use plugin in general classifieds, you may want to expand it’s functionality to all categories. Solution is to use Vehicle types as just standard type (or category grouping) and create sections & properties for each of type.



If you still need to keep vehicle section more granular, you may keep ie “Cars, Motorbikes and Other vehicles” as first 3 types and then create Real-Estate, Jobs etc.

Custom functions & integration

Plugin provides many useful functions for you to do custom integration into your page. It's recommended to have at least basic PHP knowledge, it's essential for this plugin. If you are not sure, use forums or find developer.

Search box widget

```
<?php
cap_search_box(array(
      'with_form' => true/false,
      'ignore_form_tag' => true/false,
      'design' => 'simple/medium/complex',
      'menu' => simple/advanced',
      'menu_items' => 'standard/condensed',
      'style' => 'boxed/borderless',
      'width' => 'medium/large',
      'close_btn' => true/false,
      'search_params' => array()
));
?>
```

Item attributes box

```
<?php
cap_item_attributes(
      $item_id = osc_item_id(),
      $options = array(
            'display' => 'main / sub / label_value / label_value_alt /
            icon_value / icon_value_alt / icon_value_label / icon_label_value
            / value_only / checks / bool / standard',
            'with_header' => true/false,
            'compact' => true/false,
            'align' => 'left/center',
            'limit' => => 0
      ),
      $section = 'main/sub/extra/data/bool/detail',
      $prop_ids = array(1,2,3,4...)
);
?>
```

Vehicle types widget

```
<?php
cap_widget_vehicle_types(
      $with_header = true/false,
      $wrap_elements = true/false
);
?>
```

Makes widget

```
<?php
cap_widget_makes(
      $car_type_id = 123,
      $with_icon = true/false,
      $with_header = true/false,
      $wrap_elements = true/false
);
?>
```

Models widget

```
<?php
cap_widget_models(
      $make_id = 123,
      $car_type_id = 123,
      $with_header = true/false,
      $wrap_elements = true/false
);
?>
```

More sample codes

```
<?php cap_item_attributes(osc_item_id(), array('display' => 'main',
'with_header' => false, 'limit' => 5, 'compact' => true, 'align' =>
'center')); ?>
```

```
<?php cap_item_attributes(osc_item_id(), array('display' =>
'icon_value', 'with_header' => true, 'limit' => 5, 'compact' => false,
'align' => 'center')); ?>
```

```
<?php cap_item_attributes(osc_item_id(), array('display' =>
'icon_label_value', 'with_header' => false, 'limit' => 6)); ?>
```

```
<?php cap_item_attributes(osc_item_id(), array('display' =>
'value_only'), '', array('type','make','model','body_type')); ?>
```

```
<?php cap_item_attributes(osc_item_id(), array('display' =>
'label_value_alt', 'limit' => 5), 'main'); ?>
```

```
<?php cap_item_attributes(osc_item_id(), array('display' => 'checks',
'limit' => 5), 'sub'); ?>
```

```
<?php cap_item_attributes(osc_item_id(), array('display' => 'sub',
'limit' => 3)); ?>
```

```
<?php cap_item_attributes(osc_item_id(), array('display' => 'sub',
'compact' => true, 'limit' => 10)); ?>
```

```
<?php cap_item_attributes(osc_item_id(), array('display' => 'bool',
'with_header' => true)); ?>
```

Add property values to item title (as text, list of properties defined in configuration page of plugin):

```
<?php echo cap_props_to_item_title(osc_item_id()); ?>
```

Hooks & filters for further customization

Plugin utilize hooks and filters for custom integrations with other plugins or to bring extra features.

Hooks:

- cap_item_detail
- cap_item_attributes
- cap_search_box
- cap_search_hook
- cap_item_publish
- cap_item_publish_type_section
- cap_widget_vehicle_types
- cap_widget_makes
- cap_widget_models

Filters:

- cap_create_input_properties
- cap_create_input_selected_values
- cap_create_input_options
- cap_create_elem_properties
- cap_create_elem_values
- cap_create_elem_options
- cap_item_section_styles_list
- cap_type_styles_list
- cap_styles_list2
- cap_dbtypes_list
- cap_get_input_name
- cap_get_slug
- cap_get_name
- cap_get_name_alt
- cap_get_description
- cap_get_help_text_publish
- cap_get_help_text_search
- cap_get_help_text_item