



Plugin Geopicker Handbook

How to run a widget

Files **widget.js** and **widget.css** are required to be uploaded.

For example in HTML headers you can include the following lines:

```
<script type="text/javascript" src="https://ruch-osm.sysadvisors.pl/widget.js"> </script>
```

```
<link rel="stylesheet" href="https://ruch-osm.sysadvisors.pl/widget.css"/>
```

Linki do plików:

<https://ruch-osm.sysadvisors.pl/widget.js>

<https://ruch-osm.sysadvisors.pl/widget.css>

The widget is a Javascript program and all further commands and examples apply to this language. The widget is initialized by creating an object representing one instance widget (there may be one or more instances on the page, each of them with independently operating map) by calling the constructor.

Constructor

`wid = new RuchWidget(<page item id>, <options array>, <optional – the variable containing the object jQuery>);`

followed by call initiation `wid.init()` for created object.

Page item ID is the ID of the HTML element, where the widget instance is to be displayed.

Supported options:

loadCb - callback triggered when the widget is loaded

readCb - callback triggered after loading points

selectCb - callback triggered after the buyer selects a point with an object representing that point as a parameter. The point structure has the following fields: `id` – unique point ID, `p` – point number, `c` - service COD (cash on delivery), `r` – RUCH point type (PSD, PSP, PSF, APM, PKN, PPP), `t` – widget point type (R - PSD, P- PSP or PSF, A - APM or PKN or PPK, U - PPP), `m` - method (according to the table passed in the widget call), `a` - address, `h` - description, `o` – opening hours, `la`, `lo` – geographical location

initialAddress – map starting point (point id or address)

initialType – initial point type (PSD, APM, PKN, PSP, PSF, PPP, PPK)

initialDelivery – initial type of delivery

sandbox - 0 or 1 – whether to fetch data from the test API

showCodFilter - 0 or 1 – whether to show the COD filter

showPointTypeFilter - 0 or 1 – whether to show the point type filter

showDeliveryFilter - 0 or 1 – whether to show delivery type filter

In addition, it is possible to directly pass a variable containing a jQuery object, which can be useful in situations where it does not exist in the global \$ variable (e.g. in a Magento store). When the widget initializes, it remains invisible on the page, it can be displayed, for example, using the readyCb callback.

Attention:

Whenever the concept of a point type in a widget comes into play, this should be understood as follows: each point has a specific type in the RUCH database (PSD, PSP, etc.), but additionally, in the widget they are grouped into certain "classes" specified by letters ("point type in the widget" - R, P, etc.). They play an auxiliary role, type points belonging to a given "type in the widget" are connected by the same way of handling, i.e. the same icon on the map, the same price, the same way of calling the API when creating a label.

The purpose of this is that if in the future a new point type appears in the RUCH database, which does not require a separate handling method (both in the widget and in the program that uses the widget), it will be assigned to one of the already defined "types in the widget" and it will become immediately visible on the map, even if the program using the widget does not have information of this type, and it will be available for use in the program (e.g. to be selected by customers).

However, in the opposite situation, if a new type of point requiring different handling appears in the RUCH database (e.g. a different API call when creating a label), such a type will be assigned a different "type in the widget" with a new letter. The effect of this will be that until the widget call begins to include that letter in the showWidget call, those points will not be shown on the map. Thanks to this, problems related to the fact that e.g. customers of the store will start choosing points of a type that the store does not yet support will be avoided, and it will be impossible to generate labels for these customers.

init

The function init has no parameters.

showWidget:

Next method is **wid.showWidget** (<cod>,<price table>,<method table>) - it should be invoked when the widget is to become visible on the page, for example when the buyer selects the Traffic delivery method. The widget must first be initialized with a call to RuchWidget.init(), it can be done immediately before displaying, but it is recommended to do it immediately after loading the page, so that the map and points are loaded in the background.

cod - 0 or 1, if 1 then only COD service points will be displayed

price table - prices for points R (PSD), P (PSP, PSF), A (APM, PKN, PPK), U (PPP) in the form {'R': <price R>, 'P': <price P>, 'A': <price A>, 'U': <price U>}

method table - analogous table of delivery method symbols - price and symbol are returned in the data selected system

hideWidget

The function has no parameters.

Calling this function makes the widget invisible on the page. You can display it again by calling `showWidget`, without reinitializing.

show

This function should be called when the widget is displayed in windows, in the function that handles events related to window resizing. Failure to do so may result in incorrect map refresh.

The function `show` has no parameters.

setPointType

The `wid.setPointType(<point type>)` method allows you to set the point type in the filter. Types available: PSP, PPP, PSF, PSD, APM, PKN. The method can be called after `showWidget`.

setDelivery

The `wid.setDelivery(<delivery type>)` method allows you to set the delivery method in the filter. Available values: 1 = press, 2 = courier. The method can be called after `showWidget`.

Sample implementation

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" lang="en_US" xml:lang="en_US">

<head>

  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2"/>

  <meta http-equiv="Content-Language" content="pl"/>

  <title>Demo Ruch widget

  <script type="text/javascript" src="https://code.jquery.com/jquery-1.12.4.min.js"> </script>

  <script type="text/javascript" src="https://ruch-osm.sysadvisors.pl/widget.js?i=12"> </script>

  <link rel="stylesheet" href="https://ruch-osm.sysadvisors.pl/widget.css"/>
```

```
<style>
  table {
    width: 100%;
  }

  td#codtd, td#typetd, td#start_adr, td#deliverytd {
    height: 100%;
    width: 25%;
  }
```

```

table tbody tr {
    display: flex;
    flex-direction: column;
    font-size: 32px;
}
button#but_i, button#but_s, button#but_h {
    font-size: 40px;
    margin-bottom: 20px;
    width: 100%;
    height: 90px;
}
td#start_adr, td#codtd, td#typetd, td#deliverytd {
    font-size: 40px;
    margin-bottom: 20px;
    display: flex;
    justify-content: space-between;
    align-items: center;
    width: 100%;
}
input#start, input#cod, input#type, input#delivery {
    width: 50%;
    height: 70px;
    font-size: 40px;
    margin-left: 20px;
    border: 2px solid;
}
@media (min-width: 1200px) {
    table tbody tr {
        flex-direction: row;
        font-size: 18px;
        flex-wrap: wrap;
    }
    button#but_i, button#but_s, button#but_h {
        font-size: 24px;
        margin-bottom: 20px;
        width: 80%;
        height: 40px;
        margin-right: 50px;
    }
    td#start_adr, td#codtd, td#typetd, td#deliverytd {
        font-size: 18px;
        justify-content: flex-start;
        width: 20%;
    }
    input#start, input#cod, input#type, input#delivery {
        justify-content: flex-start;
        font-size: 18px;
        margin-bottom: 20px;
        height: 40px;
    }
}
</style>

```

</head>

<body>

```
<script type="text/javascript">
```

```
var wid;
```

```
function button_init() {
```

```
    wid = new RuchWidget('widget_html', // ID div, where the widget will be displayed
```

```
    {
```

```
        loadCb: on_load,          // Called when the map is loaded
```

```
        readyCb: on_ready,       // Called when the widget is ready to be displayed
```

```
        selectCb: on_select,     // Called whet a point is selected
```

```
        initialAddress: $('#start').val(), // Start map center – point ID or address
```

```
        sandbox: 0,              // 1 if widget is supposed to fetch test data instead of production data
```

```
        showCodFilter: 1,        // Display COD (cash on delivery) filter
```

```
        showPointTypeFilter: 1, // Display point type filter
```

```
        showDeliveryFilter: 1    // Display delivery type filter
```

```
    }
```

```
);
```

```
wid.init();
```

```
}
```

```
function button_show() {
```

```
    wid.showWidget(
```

```
        parseInt($('#cod').val()), // COD or not
```

```
        {                // Price list for types
```

```
            'R': 10+$('#cod').val()2,
```

```
            'P': 11+$('#cod').val()2,
```

```
            'U': 11+$('#cod').val()2,
```

```
            'A': 11+$('#cod').val()2
```

```
        },
```

```
        {                // List of methods for types
```

```

        'R': 'ruch' + $('#cod').val(),

        'P': 'partner' + $('#cod').val(),

        'U': 'partner' + $('#cod').val(),

        'A': 'orlen' + $('#cod').val()

    }

);

wid.setPointType($('#type').val()); // sets the display of point types

wid.setDelivery($('#delivery').val()); // sets the type of delivery

}

function button_hide() {

    wid.hideWidget();

}

function on_load() {

    $('#status').html('LOAD');

    $('#but_s').show();

    $('#but_h').show();

    $('#but_i').hide();

    $('#start_adr').hide();

}

function on_ready() {

    $('#status').html('READY');

    wid.showWidget(

        parseInt($('#cod').val()), // COD or not

        { // Price list for types

            'R': 10+$('#cod').val()2,

            'P': 11+$('#cod').val()2,

```

```

        'U': 11+$('#cod').val()*2,

        'A': 11+$('#cod').val()*2

    },

    {          // List of methods for types

        'R': 'ruch' + $('#cod').val(),

        'P': 'partner' + $('#cod').val(),

        'U': 'partner' + $('#cod').val(),

        'A': 'orlen' + $('#cod').val()

    }

);

wid.setPointType($('#type').val()); // sets the display of point types

wid.setDelivery($('#delivery').val()); // sets the type of delivery

}

function on_select(p) {

    //wid.hide();

    $('#status').html('Selected ID=' + p.id + ' Address=' + p.a + ' Method=' + p.m);

}

</script>

<table> <tr>

<td> <button onclick="button_init()" id="but_i">INIT</button></td>

<td> <button onclick="button_show()" id="but_s" style="display: none;">SHOW</button></td>

<td> <button onclick="button_hide()" id="but_h" style="display: none;">HIDE</button></td>

<td id="start_adr"> Start (adres lub id punktu):<input type="text" id="start"/></td>

<td id="codtd"> COD:<input type="text" id="cod" value="0"/></td>

<td id="typetd"> Typ:<input type="text" id="type" value="" placeholder="PSP, PPP, PSF, PSD, APM, PKN"/>

<td id="deliverytd"> DELIVERY: <input type="text" id="delivery" value="" placeholder="1 = press, 2 = courier"/></td>

```


1. Points finder (points search engine)
2. Filters
3. Points list
4. Map

[Lokalizuj mnie](#)

Pokaż punkty:

Wszystkie

Rodzaj punktu:

Wszystkie

Rodzaj dostawy:

Wszystkie

Punkty w zasięgu

Kiosk/Salonik

ALEJE JEROZOLIMSKIE 40, Warszawa

AL. JEROZOLIMSKIE PRZY SMYK

Otwarte: pon-pt 06.00-20.30, sob 07.00-16.00

Koszt dostawy do punktu: 10 zł

Wybierz ten punkt

Kiosk/Salonik

Nowy Świat 6/12, Warszawa

PS W CENTRUM BANKOWO-FINANSOWY

Otwarte: pon-pt 06.15-16.15

Koszt dostawy do punktu: 10 zł

Wybierz ten punkt

Kiosk/Salonik

PLAC TRZECH KRZYŻY 1/3, Warszawa

NIEDALEKO ŻURAWIEJ

Points search engine

[Lokalizuj mnie](#)

In this section it is possible to enter a city name or postal code. The search engine supports also a different name, e.g. PKiN, Bieszczady, etc., as well as an address with the street name and building number.

When pressing the **Show** or **Enter** button on the keyboard, the nearest points will be displayed on the map.

Filters

Pokaż punkty:

Wszystkie

Rodzaj punktu:

Wszystkie

Rodzaj dostawy:

Wszystkie

The filter section allows you to select the parameters you are interested in and hide points that do not meet the requirements.

There are 3 main filters:

- **Show points:** allows you to show or hide the points supporting the COD (cash on delivery)
- **Point type:** allows you to show or hide points of a given type
- **Delivery type:** allows you to show or hide points by delivery method

Points list

Punkty w zasięgu

ŚNIEGOCKIEJ 9, Warszawa
Kiosk z wejściem, znajdujący się przy ulicy Śniegockiej 9, obok skrzyżowania ulic Śniegockiej i Koźmińskiej.

Otwarte: pon-pt 06:00-18:00, sob 06:00-14:00

Koszt dostawy do punktu: 10 zł

Wybierz ten punkt

Francuska 16, Warszawa
FRANCUSKA 16 RÓG
OBROŃCÓW

Otwarte: pon-sob 06:00-21:00, niedz 08:00-17:00

Koszt dostawy do punktu: 10 zł

Wybierz ten punkt

RONDO WASZYNGTONA 1,
Warszawa
PS PRZY STADIONIE



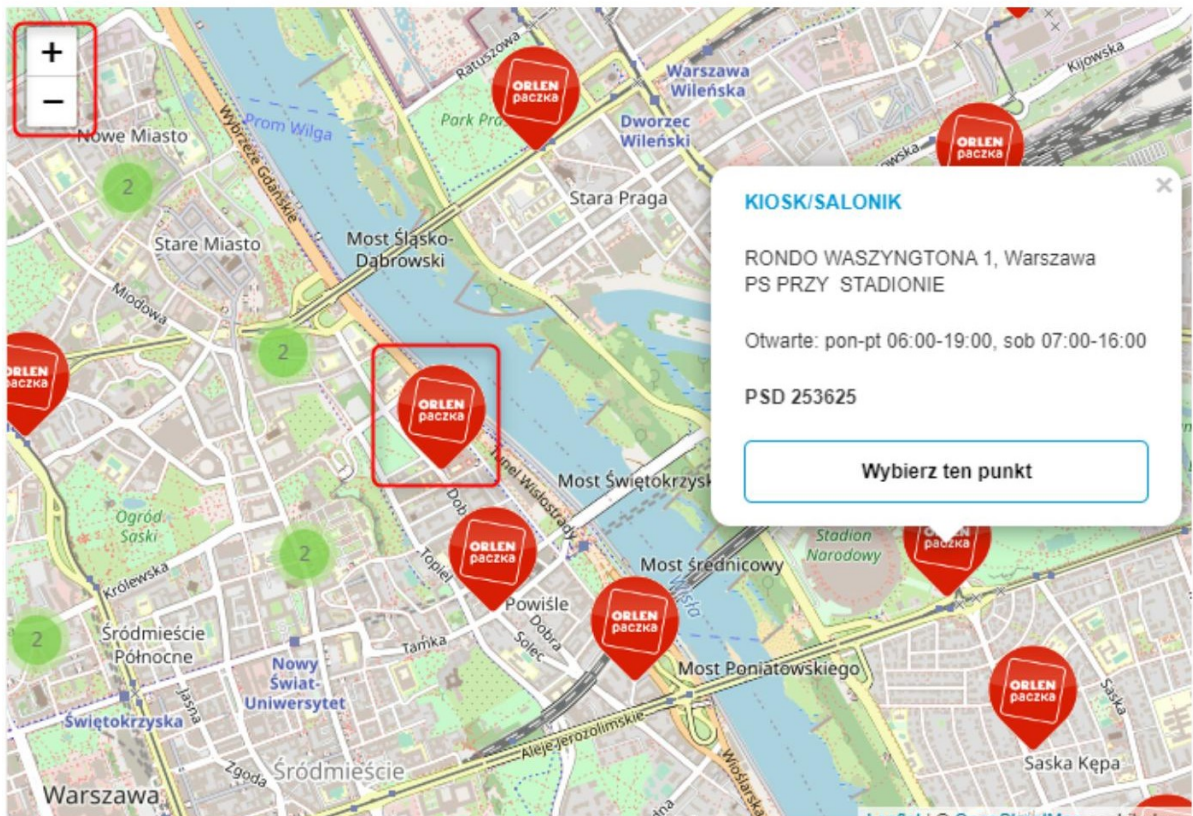
The list of points includes all points that are visible on the map. If we zoom in on the map the number of points on the list will decrease.

Each point contains information about the address, next to what it is located as well as opening hours and cost delivery to the point.

Map

There are special ways of marking points on the map (pins). We can zoom in and out of the map using the plus and minus on the map or use the mouse scroll.

The map can be moved by pressing the RMB on the map and holding down the mouse to move in the right direction.



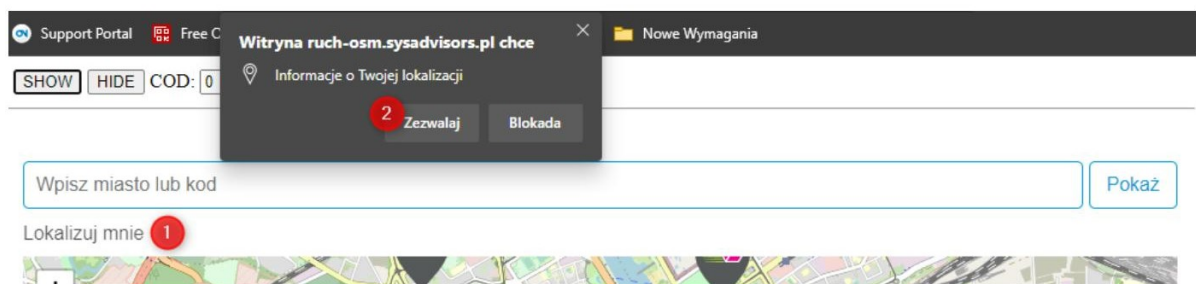
By pressing on the map on the point using LMB, the following information about the point appears: address, opening hours and description next to what is located.

Instructions for choosing a pick-up point

Searching the pickup points:

First way:

In the first way, you can use the **Locate Me** function. Press on this function as well as on consent to the use of our location will cause the view on the map to be moved to our address and showing pick-up points located in the area.



Second way:

The second way is to enter your address or zip code and then confirm with the **Enter (1)** or **Show (2)** button.



Third way:

The third way is to search for points manually. We can use the mouse to move around the map. Hold LMB on the map and move the mouse. Zooming in / zooming out is done with the scroll in the mouse or + and - located on the map..

Point selection

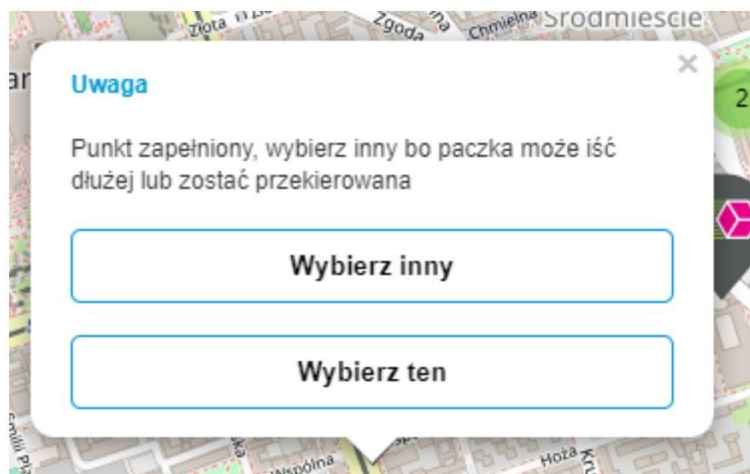
Selecting a point on the map

Once the appropriate pick-up point have been found, the user has the option of selecting a pick-up point on the map by clicking on a pin.

After selecting a pin on the map, information about the pickup point and a **Select this point** button are shown.

You must press this button.

After selecting the point, additional information may appear that the pick-up point is full and the delivery time of the parcel can be longer or the parcel might be redirected. In this case, the user can either select a different point or confirm selected.



Selecting a point form the list

After selecting the appropriate pickup point from the list, the selected point will be shown on the map.

When the point is overcrowded, an appropriate message will appear that the package may take longer or can be redirected.

02-495
Pokaż

Lokalizuj
mnie

Utwarte: pon-pt 11:00-19:00, sob 10:00-15:30
Odległość od szukanego punktu: 0.5 km
Koszt dostawy do punktu: 5 zł

Odbieram tutaj

Kiosk/Salonik
BOHATERÓW WARSZAWY 11, Warszawa
PRZY LOTTO, OBOK PRZYSTANEK
Otwarte: pon-pt 06:00-18:00, sob 06:00-15:00
Odległość od szukanego punktu: 0.5 km
Koszt dostawy do punktu: 5 zł

Wybierz ten punkt 1

Kiosk/Salonik
OBROŃCÓW HELU 3, Warszawa
OBOK PL. TYŚĄCLECIA I POCZTA
Otwarte: pon-pt 06:00-19:00, sob 06:00-15:00
Odległość od szukanego punktu: 0.6 km
Koszt dostawy do punktu: 5 zł

Wybierz ten punkt

Wybrany punkt: Bohaterów Warszawy 28, Warszawa 2

After selecting a point, the list will show which point the user has selected.

Technical requirements

The jQuery library version 1.12.4 or higher is required for the widget to work. For standard jQuery loading when the jQuery object is available in a global variable \$, the widget will detect it itself. For custom configurations, when jQuery is loaded into another (also local) variable, the variable containing the jQuery object should be pass as a parameter to the constructor of the widget object (see constructor description).

The widget works in browsers that support javascript:

Firefox 30+

Chrome 20+

Opera 30+

IE 7-11 (with incomplete layout support)

Edge