Widget user guide

ORLEN Paczka – point map

Document version: 1.4 | November 2026

TABLE OF CONTENTS

1 BEFORE YOU START 2
1.1 What the map widget can do 2
1.2 Data required to run the widget 3
1.3 Recommended order of work 3
2 OBTAINING AND PREPARING THE TOKEN 3
2.1 How to obtain a widget token 3
2.2 Testing on localhost 4
2.3 Widget token and API credentials 4
3 EMBEDDING THE MAP ON THE PAGE 4
3.1 Adding the script in the head section 4
3.2 Adding the map element in the body section 4
3.3 What to check after initial setup 5
4 CHOOSING THE DISPLAY MODE 5
4.1 Map as a modal window 5
4.2 Map embedded in the page content 5
4.3 Forcing modal mode with the data-modal attribute 5
4.4 Opening the same modal from multiple elements 5
5 CONFIGURING WIDGET PARAMETERS 6
5.1 Saving the point ID and name 6
5.2 Point type: data-type 6
5.3 Configuring point type visibility 6
5.4 Initial address: data-address and data-address-id 8
5.5 Map and list layout: data-layout 8
5.6 Additional buttons on the map 8
5.7 Cash on delivery (COD) 9
5.8 Parameter summary 9
6 RECEIVING DATA FOR THE SELECTED POINT 10
6.1 The simplest method – form fields 10
6.2 The orlenSelectPoint event 10

6.3 What data the event provides ... 11
7 READY-TO-USE IMPLEMENTATION EXAMPLES ... 11
7.1 Example: button and saving the point in the cart ... 11
7.2 Example: map embedded in a form ... 11
7.3 Example: form with an additional trigger ... 11
7.4 When to use event handling ... 12
8 TESTING THE IMPLEMENTATION ... 12
8.1 Checks after embedding ... 12
8.2 Testing after a template or configuration change ... 12
9 TROUBLESHOOTING ... 13
9.1 Common issues ... 13
9.2 Quick browser diagnostics ... 14
9.3 Information required for an issue report ... 14
10 FREQUENTLY ASKED QUESTIONS ... 15
10.1 Can I use the same token in the test and production environments? ... 15
10.2 Can I save the ID and address of the selected pickup point? ... 15
10.3 Does the ORLEN Paczka map support cash on delivery (COD)? ... 15
11 CHANGE HISTORY AND SOURCES ... 16

1 BEFORE YOU START

This guide is intended for online store owners and anyone implementing ORLEN Paczka point selection on a website, in a shopping cart, or in a form. It contains simple steps, ready-to-use HTML and JavaScript snippets, and tips for verifying the implementation.

1.1 What the map widget can do

- Displays an interactive map and a list of ORLEN Paczka points from which the user can select a point.
- Works as a **modal window** opened with a button or as an **embedded map** in a specified place on the page.
- Lets you specify the point type, the initial search address, and the layout of the list and the map.
- Lets you insert the point ID and name into form fields or read the information through the **orlenSelectPoint** event.
- Lets you limit map visibility to selected point types, for example, parcel lockers, ORLEN fuel stations, or partner points.
- Lets you specify whether the user can change the active point filters on the map.

> **Important:** The widget is used to select a point. It does not replace the connection to the ORLEN Paczka API or the functions for creating shipping labels and ordering courier pickups.

1.2 Data required to run the widget

Data	Where to get it	What it is used for
Widget token	ORLEN Paczka	Lets the map load on the assigned website
Website address	The address of the domain where you will embed the widget	Used to assign the token to the store
PWR customer number	From the ORLEN Paczka customer account data	You provide it in the token request
Access to website HTML	Website administrator or platform provider	Lets you add the script and the map container
Point data storage	Cart, checkout, or order form	Stores the ID of the point selected by the customer

1.3 Recommended order of work

This order reduces the risk of publishing a solution that does not work.

Step 1. Get a token assigned to the website domain.
Step 2. Insert the widget script in the <head> section of the page.
Step 3. Add an element with the **orlen-widget** class in the <body> section.
Step 4. Configure the mode, the form fields, and the optional map parameters.
Step 5. Test point selection and saving on a computer and on a phone.
Step 6. Make the solution available to customers only after a successful test.

2 OBTAINING AND PREPARING THE TOKEN

2.1 How to obtain a widget token

To run the ORLEN Paczka map, you need an individual access token assigned to the address of the website where the map will be used.

To get a token, contact your ORLEN Paczka account manager or send a request to integracje@orlenpaczka.pl.

Step 1. Prepare the address of the website where the map will be placed, for example, **https://sklep.example.pl**
Step 2. Prepare your PWR customer number. If you do not have one, provide your Polish tax identification number (NIP).
Step 3. In your request, state that you need a token for the ORLEN Paczka map widget. Provide the website address and specify whether the map will be used in a production, test, or local environment (localhost).

> **Tip:** If you plan to use the map on several domains or subdomains, include all addresses in your request to confirm that they can be supported.

2.2 Testing on localhost

For local testing, you need a **separate, temporary token**. This token expires **3 months after it is generated**.

Step 1. State in your request that the widget will run on localhost.
Step 2. Use the local token you receive in the code of the test version.
Step 3. Before publishing, replace it with the token intended for the production domain.

2.3 Widget token and API credentials

The map widget token is independent of the **PartnerID / PartnerKey** credentials used for integration with API services. Do not place the PartnerKey in the page code or pass it in widget parameters.

The script runs in the browser. The map token is technically visible in the script download request, so the main mechanism for securing it is correctly linking the token to a specific website when the token is generated.

3 EMBEDDING THE MAP ON THE PAGE

To embed the map, add a short script and specify the HTML element where the map should appear. You can place the code directly in the website template or in a feature for adding custom HTML / JavaScript.

3.1 Adding the script in the head section

Where to place it: Page source code → <head> section

Step 1. Open the template of the page where the widget will be used.
Step 2. Insert the following code in the **<head>** section.
Step 3. Replace **INSERT_TOKEN_HERE** with the token you received from ORLEN Paczka.
Step 4. Save the changes and publish.

```
<script>
(function (o, r, l, e, n) {
  o[l] = o[l] || [];
  var f = r.getElementsByTagName('head')[0];
  var j = r.createElement('script');
  j.async = true;
  j.src = e + 'widget.js?token=' + n + '&v=1.0.0&t=' +
    Math.floor(new Date().getTime() / 1000);
  f.insertBefore(j, f.firstChild);
})(window, document, 'orlenpaczka',
   'https://mapa.orlenpaczka.pl/', 'INSERT_TOKEN_HERE');
</script>
```

Sample initialization script

3.2 Adding the map element in the body section

Where to place it: Page template → <body> section → point selection area

The widget searches for elements with the **orlen-widget** class. The tag type determines the default display mode: a button or a link opens a modal, and any other element (for example, a div) embeds the widget inside the element.

```
<!-- Button: map in a modal window -->
<button type="button" class="orlen-widget">Select an ORLEN Paczka point</button>

<!-- Container: map in the page content -->
<div class="orlen-widget"></div>
```

Result: The page shows a button that opens the map or a map placed in the selected area.

3.3 What to check after initial setup

Step 1. Open the page that contains the widget in private browsing mode.
Step 2. Check that the widget script loads without errors.
Step 3. Open the map or scroll to the container to confirm that the points are visible.
Step 4. Select any point.

The following sections explain how to save the selected point to the form.

4 CHOOSING THE DISPLAY MODE

4.1 Map as a modal window

A modal works well when you want to show only a "Select a point" button on the page and the map should open after a click.

```
<button type="button" class="orlen-widget">
  Find a point on the map
</button>
```

Step 1. Add a button with the **orlen-widget** class.
Step 2. Check that clicking it opens the map window.
Step 3. Select a point and check that the window closes as the widget is designed to.

4.2 Map embedded in the page content

This option makes it easier to show the map without an additional click, for example, in a form for selecting a pickup location.

```
<div class="orlen-widget"></div>
```

Tip: Make sure the container on the page is wide enough and check how the map works on mobile devices. The surrounding page layout depends on your platform.

4.3 Forcing modal mode with the data-modal attribute

The **data-modal="true"** parameter lets you display the map in a separate window. For <button> and <a> elements, this is the default setting. If the map should open from another type of element (for example, a div), you must also specify this parameter.

```
<div class="orlen-widget" data-modal="true">Check on the map</div>
```

4.4 Opening the same modal from multiple elements

The **data-modal-id** parameter gives the window its own ID. Another element can open the same window after you add the **orlen-widget-open** class and the **data-modal-target-id** attribute.

```
<input type="text" class="orlen-widget-open"
       data-modal-target-id="moj-modal" readonly
       placeholder="Click to select a point" />
<button type="button" class="orlen-widget"
        data-modal="true" data-modal-id="moj-modal">
  Select a point
</button>
```

Note: If you do not specify data-modal-id, the widget assigns an automatic name orlen-widget-modal-X, where X is the widget number starting from 0. When opening the modal from multiple elements, it is more convenient to assign a fixed ID.

5 CONFIGURING WIDGET PARAMETERS

You add configuration parameters to the HTML tag with the orlen-widget class as data-* attributes.

5.1 Saving the point ID and name

data-target specifies the element into which the widget writes the ID of the selected point. **data-label** specifies the place into which it writes the point name (address and city). This can be an input or, for example, a span.

```
<button type="button" class="orlen-widget"
        data-target="#orlen-point-id"
        data-label="#orlen-point-name">Select a point</button>
<input type="hidden" id="orlen-point-id" name="orlen_point_id" />
<span id="orlen-point-name">No point selected</span>
```

Important: The selectors specified in data-target and data-label must match elements that exist on the page. Before you continue, check that the selected ID actually reaches the form.

5.2 Point type: data-type

Value	What it displays
pickup	Parcel pickup points
dropoff	Parcel drop-off points
all	All points – default value

```
<div class="orlen-widget" data-type="pickup"></div>
```

Tip: The **data-type** parameter specifies whether the map should show pickup points, drop-off points, or all points. If you also want to limit the map to specific location types, for example, only parcel lockers, use the parameters described in the next subsection.

5.3 Configuring point type visibility

The ORLEN Paczka map widget lets you adjust point visibility to the needs of your website. You can make all locations available or limit the map to selected point types.

To do this, use the appropriate parameters in the HTML element with the **orlen-widget** class. You can combine parameters to get the map display you want.

Parameter	Point type
data-apm="true"	ORLEN Paczka parcel lockers (APM)
data-pkn="true"	ORLEN fuel stations (PKN)
data-ppp="true"	Partner points that support drop-off and pickup (PPP)
data-ppk="true"	Partner points that support pickup only (PPK)

Choose a scenario for your website:

Goal	HTML settings	What the user sees
Parcel pickup	data-type="pickup"	Points that support parcel pickup
Parcel drop-off	data-type="dropoff"	Points that support parcel drop-off
All points	data-type="all"	All available points – default setting
Parcel lockers only	data-apm="true"	APM parcel lockers only
Lockers and ORLEN stations	data-apm="true" data-pkn="true"	APM parcel lockers and ORLEN fuel stations
Partner points	data-ppp="true" data-ppk="true"	PPP and PPK partner points
PPK points only	data-ppk="true"	Only partner points that support parcel pickup
All points except lockers	data-pkn="true" data-ppp="true" data-ppk="true"	ORLEN fuel stations (PKN) and PPP and PPK partner points, without APM parcel lockers.
Lockers only, filters cannot be changed	data-apm="true" data-filter-checkbox="false"	Parcel lockers only, with no option to change the active filters

Important: Do not combine the **data-type="dropoff"** parameter with the **data-ppk="true"** filter as the only selected point type. PPK points support parcel pickup only, so this configuration will not display any points on the map.

How do visibility limits work?

If you specify at least one point type, the map shows only the specified location types. Other point types do not appear on the map, and the corresponding filter buttons are hidden automatically.

If you do not specify any of the **data-apm**, **data-pkn**, **data-ppp**, and **data-ppk** parameters, the map makes all point types available by default.

Blocking filter changes

If you do not want the user to change the active filters, add the **data-filter-checkbox="false"** parameter.

This parameter works together with at least one setting that specifies the point type. It blocks changes to the active filters and hides the filter buttons.

```
<div class="orlen-widget"
     data-apm="true"
     data-filter-checkbox="false"></div>
```

Important: Parameters that specify point types require the exact value **true**, for example, **data-apm="true"**. Writing data-apm="1" or only data-apm does not enable the filter.

5.4 Initial address: data-address and data-address-id

data-address passes the address to search for when the map starts. **data-address-id** passes the address ID of the point to search for.

```
<div class="orlen-widget"
     data-address="ul. Przykładowa 1, 00-001 Warszawa"></div>

<div class="orlen-widget" data-address-id="123"></div>
```

5.5 Map and list layout: data-layout

If you do not set the **data-layout** parameter, the widget automatically applies a layout that matches the selected map display mode.

Value	Appearance
normal	Point list next to the map. Default layout for a map embedded directly on the page.
tabs	Map and point list displayed in tabs. Default layout for a modal window.
orlen	Point list displayed below the map.

```
<div class="orlen-widget" data-layout="tabs"></div>
```

5.6 Additional buttons on the map

The ORLEN Paczka map widget lets you display additional buttons that make it easier for the user to get information about the selected point or to use the available functions.

Parameter	Function
data-report-bug-button="true"	Displays a button that lets the user report a bug.
data-navigation-button="true"	Displays a navigation button.
data-details-button="true"	Displays a "Details" button that leads to the page of the selected point on the orlenpaczka.pl website.

All three buttons are disabled by default. You can enable selected functions by adding the appropriate parameters to the map element.

```
<div class="orlen-widget"
     data-report-bug-button="true"
     data-navigation-button="true"
     data-details-button="true"></div>
```

5.7 Cash on delivery (COD)

The cash on delivery (COD) service has been disabled since January 2, 2025. Do not use the **data-cod="true"** parameter.

5.8 Parameter summary

Attribute	Meaning / default value
data-target="#ID"	Inserts the ID of the selected point into the specified field
data-label="#ID"	Inserts the point description (address + city) into the element
data-type	pickup / dropoff / all (default: all)
data-address	Address searched for when the map starts
data-address-id	Address ID of the point to search for
data-modal	Default: true for <button> and <a>, false for other elements
data-layout	normal / tabs / orlen. Default: normal for maps embedded on the page and tabs for modal windows
data-report-bug-button	true / false (default: false)
data-navigation-button	true / false (default: false)
data-modal-id	Custom modal window ID
data-modal-target-id	ID of the modal opened by an additional element

Attribute	Meaning / default value
data-cod	Legacy parameter; do not use for COD
data-apm	Displays APM points; value true required
data-pkn	Displays PKN points; value true required
data-ppp	Displays PPP points only, excluding PPK; value true required
data-ppk	Displays PPK points, which support pickup only; value true required
data-filter-checkbox	Specifies whether the active filters can be changed. Default: true; the value false blocks changes when point types are declared
data-details-button	Enables the "Details" button; default: false

6 RECEIVING DATA FOR THE SELECTED POINT

6.1 The simplest method – form fields

The simplest way to pass information about the selected pickup point is to use the **data-target** and **data-label** parameters.

The first one lets you pass the ID of the selected point to the specified form field. The second one lets you show the customer information about the selected location.

Step 1. In the order form, add a field to store the ID of the selected point.
Step 2. Connect this field to the map with the **data-target** parameter.
Step 3. If you want the customer to see the address of the selected location, add an appropriate element on the page and connect it to the map with **data-label**.
Step 4. Select a point on the map and check that its ID and description were correctly passed to the specified elements.
Step 5. Make sure that the store system saves the point ID in the order data.

Important: The map widget lets you select a point and pass its data to the specified page elements. However, filling in the form field alone does not guarantee that the point is saved in the order. How this information is passed and saved depends on the system in which the map is embedded.

6.2 The orlenSelectPoint event

The **orlenSelectPoint** event is triggered on the **.orlen-widget-modal** element after a point is selected. It lets you capture detailed information about the selected point. You get the detailed information from **e.detail.marker.options**.

```
document.querySelectorAll('.orlen-widget-modal').forEach(function(widget) {
  widget.addEventListener('orlenSelectPoint', function(e) {
    const pointData = e.detail.marker.options;
    console.log(pointData);
  });
});
```

6.3 What data the event provides

The event data includes the point ID, address, city, coordinates, type, destination code, opening hours, and information on whether the point supports drop-off and pickup. The abbreviated example below preserves the original field names.

```
{
  "id": "241050", "postalCode": "00-031",
  "addressLine": "Szpitalna 5", "name": "241050",
  "province": "mazowieckie", "city": "Warszawa",
  "longitude": 21.0144942, "latitude": 52.2330747,
  "pointType": {"name": "PPP"},
  "destinationCode": "WS-241050-U6-12",
  "cashOnDelivery": false,
  "dropOffPoint": true, "pickupPoint": true,
  "openingHours": "Poniedziałek - Piątek: 09:00-18:00<br />..."
}
```

7 READY-TO-USE IMPLEMENTATION EXAMPLES

The following examples show how to add a button for selecting an ORLEN Paczka point and how to pass the ID and description of the selected location to the appropriate page elements.

Place the code snippet in the appropriate place in the order form. For the selected point to be assigned to the order, the store system must also support saving the ID passed by the widget.

Result: After a point is selected, its ID is passed to the specified field, and the customer sees the description of the selected location. Saving the ID in the order data requires appropriate support on the store system side.

7.1 Example: button and saving the point in the cart

```
<label>Pickup at an ORLEN Paczka point</label>
<button type="button" class="orlen-widget"
        data-type="pickup"
        data-target="#orlen-point-id"
        data-label="#orlen-point-label">Select a point</button>
<input type="hidden" id="orlen-point-id" name="orlen_point_id" />
<p id="orlen-point-label">No point selected.</p>
```

7.2 Example: map embedded in a form

```
<div class="orlen-widget" data-type="pickup"
     data-layout="tabs" data-target="#orlen-point-id"
     data-label="#orlen-point-label"></div>
<input type="hidden" id="orlen-point-id" name="orlen_point_id" />
<span id="orlen-point-label">Select a point on the map.</span>
```

7.3 Example: form with an additional trigger

A custom modal ID lets you open the same widget with a button and with an additional field.

```
<button type="button" class="orlen-widget"
        data-modal="true" data-modal-id="wybor-orlen"
        data-target="#orlen-point-id">Select a location</button>
<input type="text" class="orlen-widget-open" readonly
       data-modal-target-id="wybor-orlen"
       placeholder="Open the point map" />
<input type="hidden" id="orlen-point-id" name="orlen_point_id" />
```

7.4 When to use event handling

The **orlenSelectPoint** event is useful when, in addition to the point ID, the system must save additional fields, for example, the destination code or the point type. If you need only the ID and the text name, **data-target** and **data-label** are enough.

8 TESTING THE IMPLEMENTATION

8.1 Checks after embedding

Step 1. Open the page after saving the changes and make sure that the map is visible or opens after you click the appropriate button.
Step 2. Search for an ORLEN Paczka pickup point and check that you can select it on the map.
Step 3. If the visibility of point types on the map is limited, check that only the specified location types are displayed.
Step 4. Make sure that point types excluded from the configuration do not appear on the map and that the corresponding filter buttons are not available.
Step 5. If the data-filter-checkbox="false" parameter is used, check that the user cannot change the active filters.
Step 6. If the "Details" button is enabled, check that it leads to the page of the correct ORLEN Paczka point.
Step 7. After you select a point, check that its ID was passed to the appropriate field and that the customer sees information about the selected location.
Step 8. Go to the next step of placing the order and make sure that the selected point was not lost.
Step 9. Place a test order and check that the ID of the selected point was correctly saved in the store system.
Step 10. Repeat the test on a phone and in another web browser.

Result: The map works correctly, lets you select a point, passes its data, and works with the order form. The selected location is available in the saved order data.

8.2 Testing after a template or configuration change

At a minimum, repeat point selection and a test form submission after every change to the checkout, the page template, the script consent mechanism, or scripts that load content asynchronously.

9 TROUBLESHOOTING

9.1 Common issues

Problem	What to check
The map does not load	Check that you are using the correct token assigned to the domain. Also make sure that the configuration contains the correct address of the ORLEN Paczka map service.
The map does not open in a separate window	Check that the element the customer uses to open the map is configured correctly. Depending on the element type, you might need to additionally enable displaying the map in a separate window.
The embedded map is not visible on the page	Make sure that a place where the map should be displayed has been prepared on the page and that the element intended for displaying it has been added correctly.
The ID of the selected point is not saved	Check that the map is correctly connected to the field where the ID of the point selected by the customer should be saved.
The description of the selected point does not appear	Make sure that the page has a place intended for displaying information about the selected point and that it is correctly linked to the map.
The map cannot be opened from an additional place on the page	If the map should open from several places, for example, with a button and a point selection field, check that all these elements are correctly linked to the same map window.
Information about the selected point is not passed	Check that the integration is configured to receive information about the point selected by the customer. If the problem continues, ask the person responsible for the implementation to verify the handling of the orlenSelectPoint event.
The map does not work in the test environment (localhost)	Make sure that you are using a separate token intended for local testing. Also check that it has not expired. The test token is valid for 3 months after it is generated.
Cash on delivery (COD) cannot be selected	This is expected behavior. The cash on delivery (COD) service was disabled on January 2, 2025, and is no longer available.
Not all point types are visible on the map	Check that the map configuration does not limit visibility to selected location types. If such a limit is applied, other points will not be displayed.
The filters on the map cannot be changed	Check that the option for the user to change the filters is not disabled in the map settings.

Problem	What to check
The "Details" button is not displayed	Make sure that the function for displaying detailed point information is enabled in the map configuration.

9.2 Quick browser diagnostics

The following steps let you check whether the browser correctly downloads the map script and whether the website settings or security features interfere with how it works.

Step 1. Open **Developer Tools → Console** and check for JavaScript errors.
Step 2. On the **Network tab**, check that the request for **widget.js** was sent and handled correctly.
Step 3. Verify the code of the map element and the target field: **class**, **id**, **data-target**, **data-label**.
Step 4. Temporarily disable your own modifications to the test page that might hide the container or delay its creation.
Step 5. Try to reproduce the issue in another browser or in private browsing mode.

9.3 Information required for an issue report

If the ORLEN Paczka map widget still does not work correctly after you check the configuration, report the issue using the form available at https://www.orlenpaczka.pl/formularz-integracje/

To help identify the problem quickly and resolve it faster, prepare the following information:

1. **Company details and website address** – provide the company name, the PWR customer number (or NIP if you do not have a PWR number), and the address of the website where the map is placed.
2. **Environment where the problem occurs** – specify whether the problem affects a live online store, a test environment, or local testing (localhost).
3. **Problem description** – explain what does not work correctly. Describe which actions were performed before the error occurred, what result was expected, and what actually happened.
4. **Circumstances of the error** – specify which browser and device you use and when the problem occurred. If the error appears only in specific situations, include this information as well.
5. **Map display mode** – specify whether the map opens in a separate window or is visible directly on the page. If the problem concerns passing the data of the selected point or additional map settings, describe which functions do not work as expected.
6. **Additional error information** – if possible, prepare a screenshot that shows the problem or the text of the displayed message. They can help determine the cause of the map malfunction.

Important: Do not include passwords, PartnerKey access keys, or full authorization tokens in your issue report. Before you share screenshots, make sure that they do not contain confidential data.

10 FREQUENTLY ASKED QUESTIONS

10.1 Can I use the same token in the test and production environments?

No. To use the ORLEN Paczka map in a test environment, including localhost, you need a separate test token, which is valid for 3 months from the date it is generated.

10.2 Can I save the ID and address of the selected pickup point?

Yes. The ORLEN Paczka map widget lets you pass information about the point selected by the customer to the appropriate website elements.

Using the **data-target** parameter, you can pass the point ID to the specified form field. The **data-label** parameter lets you display a description of the selected location, including the address and city.

If you need additional information about the point, you can use the **orlenSelectPoint** event, which provides more detailed data, for example, the postal code, the point type, or the opening hours.

If the point information should be saved in the order, the store system must support saving it. The widget itself is responsible for selecting the point and passing its data.

10.3 Does the ORLEN Paczka map support cash on delivery (COD)?

No. Since January 2, 2025, the cash on delivery (COD) service has been disabled and is not available in the ORLEN Paczka map widget.

For this reason, do not use the **data-cod="true"** parameter when you configure the map; this parameter previously let you display only points that support cash on delivery. The absence of this option is expected and does not indicate a configuration error.

11 CHANGE HISTORY AND SOURCES

Date	Version	Scope of changes
08.2023	1.0	First release of the widget documentation
05.2024	1.1	Added the data-cod flag (legacy function)
03.2025	1.2	COD disabled; token requirements expanded
05.2025	1.3	Information about the orlenSelectPoint event
11.2026	1.4	Updated the guide layout and template, expanded the implementation descriptions, configuration examples, and help section.