[**1 The widget’s functions**](#_5jgc8dseqzpe) **3**

[1.1 The widget’s capabilities](#_x8cb6xvbbe7i) 3

[1.2 Requirements](#_jmhk529vhd61) 3

[1.3 Compatibility with other scripts and other requirements](#_wqzfj5d2bi4z) 3

[1.4 Questions and comments on the widget’s work](#_6v7dkc6c060h) 3

[**2 Installation**](#_to30kf7y8tb) **4**

[2.1 Ways of installing the widget](#_x9rrxmfq01bt) 4

[2.2 Unpacking the archive](#_ce4dwqid9ckm) 4

[2.3 Authorization and configuration](#_1pntz3c2xafk) 4

[2.4 Connecting scripts](#_xdaznq20wivs) 5

[2.5 Placing the widget on a page](#_pezkkkrqb87d) 5

[**3 The widget’s settings**](#_xvh8a1928vvb) **6**

[3.1 Example of creating the widget with all settings](#_etpdgyd1xx76) 6

[3.2 Description of server settings](#_4dzm2ukpgwf5) 7

[3.2.1 showWarns](#_rf9g8nvz04ew) 7

[3.2.2 showErrors](#_dk0v3dj1l2si) 7

[3.2.3 showLogs](#_ii4n4vhor4ae) 7

[3.2.4 hideMessages](#_ul2kkswhm4ir) 7

[3.2.5 path](#_gpintumrjc0d) 7

[3.2.6 templatepath](#_f37ft6ldzcec) 7

[3.2.7 servicepath](#_wl4uq4v6ch3d) 8

[3.2.8 apikey](#_phxhjfhfw5sx) 8

[3.3 Description of display settings](#_q1uk839ikqh4) 8

[3.3.1 choose](#_i03o56tff2u6) 8

[3.3.2 link](#_r55hl47v8zbz) 8

[3.3.3 popup](#_tj8k5iwmd2t5) 8

[3.3.4 hidedress](#_h17glzdob8wv) 8

[3.3.5 hidecash](#_zhkgwe4am5zt) 8

[3.3.6 hidedelt](#_bmujj0oiafjm) 9

[3.3.7 region](#_7rmtgyjoh4nv) 9

[3.3.8 detailAddress](#_dn52k8dak7yp) 9

[3.4 Delivery settings](#_1oddi49kfgfq) 9

[3.4.1 defaultCity](#_pe5gwvum6vxo) 9

[3.4.2 lang](#_d37gksfet6pj) 9

[3.4.3 currency](#_ul94jst0708r) 9

[3.4.4 country](#_1aq4ruyysmjn) 10

[3.4.5 cityFrom](#_klfvudbtczeu) 10

[3.4.6 goods](#_mm5ceyiuoffd) 10

[3.5 The widget’s events](#_cb4kte523lcs) 10

[3.5.1 Loading the widget (onReady)](#_i7jlymzfpwh7) 10

[3.5.2 Calculation of delivery (onCalculate)](#_a91elljm1l2e) 11

[3.5.3 Selecting pick-up point (onChoose)](#_syt3g9d034ca) 11

[3.5.4 Selecting delivery profile (Courier) (onChooseProfile)](#_6vg3w52w1cty) 12

[3.5.5 Selecting delivery address (Courier) (onChooseAddress)](#_pmqd3m2ysmno) 13

[3.6 The widget’s methods](#_v75t95f9wpd7) 14

[3.6.1 Operations with cities](#_lguke1fi5urv) 14

[city.get()](#_ikqn3y36k17j) 14

[city.set(city)](#_nssekhz0jly9) 14

[city.check(city)](#_k8sy25fpm3yw) 14

[3.6.2 Operations with parcels](#_oy93lfdulx5l) 14

[сargo.add (params)](#_ejwloqdmo3ce) 14

[cargo.get()](#_mi854zjzngkg) 14

[cargo.reset()](#_86c31zbjvzf9) 14

[3.6.3 User interface](#_rvjguk5vz50z) 15

[open()](#_j934jpuk5k5) 15

[close()](#_biyd82n4ji43) 15

[**4 Examples of work**](#_1lxcnnkrdt4p) **15**

[4.1 Creating an info-widget](#_fvm5ohkoetji) 15

[4.2 Widget for a product page](#_as1pmvxl39zu) 15

[4.3 Widget for ordering](#_pzdui1avj86) 16

[4.4 Widget for storefront](#_o2c5h5h9oc8f) 16

# **1 The widget’s functions**

The widget is designed to display relevant information about CDEK pick-up points linked to a map, with an opportunity to calculate delivery (time and price) for the specified dimensions of the goods. The widget can be displayed as a static element of the page, as well as a pop-up window.

## **1.1 The widget’s capabilities**

* Display list of pick-up points for the selected city
* Change of displayed city
* Calculation of delivery for the specified dimensions
* Opportunity for the buyer to select a pick-up point with transfer all the data necessary for processing to the handler function
* Display of detailed information for each pick-up point and parcel terminals
* Display of support information about the service
* Flexible widget display settings
* Minimization of integration conflicts
* Easy connection and configuration

## **1.2 Requirements**

Following components installed on the server are required for correct operation of the widget:

* PHP version not older than 5.3
* CURL extension for PHP

Requirements to the element (“link” parameter), in which the widget will be built:

* Height of the element must be specified; otherwise it may be equal to zero and the widget will not be displayed on the website.
* Make sure that this element has no styles related to the box model, excluding “width\height\float”. Otherwise the widget may appear distorted.
* It is recommended to set the element's width not less than 800 px, height – not less than 600 px

## **1.3 Compatibility with other scripts and other requirements**

* Stability of the widget is not guaranteed when used on a page with the Yandex Maps widget
* Problems with the widget’s work may appear when used on a page with the jquery.mCustomScrollbar widget

## **1.4 Questions and comments on the widget’s work**

Please send all your questions and comments to *integrator@cdek.ru*

If you have problems with connection, configuration or displaying the widget on your web-site, please indicate the link to your page where you tried to place the widget, so that we can answer your questions correctly.

To provide full assistance to our clients, we need the following information:

1. Link to a working example (if it’s impossible to provide the link, you need to save the page in html format and attach it to the email so that the picture on the client side is fully presented)
2. Comprehensive description of the problem
3. Which way of installation the client wants to use (see 2.1)
4. With which actions the error occurs

# **2 Installation**

## **2.1 Ways of installing the widget**

There are two ways of installing the widget:

1. Local, when all the files are stored on your server.  
 2. Minimal, when all the files are loaded from our web-site, and only authorization data is downloaded from your web-site.

Local installation allows you to keep all the files on your server, which provides faster loading of scripts, compared to the minimal installation. Also, this type of installation allows you to edit scripts and widget styles to suit your requirements.

On the other hand, the minimal installation may also be good for you because you need to edit only 1 file and include a link to the file in the script.

## **2.2 Unpacking the archive**

To install the widget, you need to unpack the archive with the widget. The archive contains:

* this guide - file ‘Documentation\_v.2.5.3.docx;
* the widget’s scripts – “widget” directory;
* examples of the widget’s work – “examples” directory;
* short instruction on the widget’s installation – “index.php” file.

It is important to know that the widget includes server files (widget/scripts/service.php and widget/scripts/template.php), correct function of which requires placing them on the web-server. This is why we recommend to see examples on the web-server.

**For local installation** of the widget, you need to copy “widget” directory to the web-site directory:

For instance, the web-site is located in /home/site/ directory; copy the catalogue with the widget and the entire path to the widget will look like this: /home/site/widget/

Dependency of the scripts’ locations must be preserved. Note that the widget’s assembly contains a server file (widget/scripts/service.php).

**For minimal installation** of the widget, you need to copy only server file (widget/scripts/service.php) to your web-site, for instance, to /home/site/widget/scripts/service.php directory

## **2.3 Authorization and configuration**

For the correct calculation of the price of delivery, the widget needs authorization data to access the CDEK integration service. To get the data for the Integration Account, you need to click the "Create Key" button in the Integration section in your personal account, then the Account ID and Password will appear in the integration section.

To enter the data, open scripts/service.php file. Enter the data in the 17th (account) and 20th (key) lines inside the quotation marks.

If you need to change priority of tariff calculations, change their order in the lines 11 (courier delivery) and 14 (customer pick-up from the pick-up point). You can find numbers of the tariffs in the documentation to the integration service <https://confluence.cdek.ru/x/gUju> in the section Appendix 1

## **2.4 Connecting scripts**

To connect the **local version of the widget**, you need to add the following code to the page: (we recommend to place it inside the <head> tag):  
*<****script type="text/javascript" src="https://your.site/path-of-widget-on-your-site/widjet.js" charset="utf-8" id="ISDEKscript"*** *></****script****>*

To connect the **minimal version of the widget**, you need to add the following code to the page: (we recommend to place it inside the <head> tag):  
*<****script type="text/javascript" id="ISDEKscript" src="https://widget.cdek.ru/widget/widjet.js" charset="utf-8"****></****script****>*

## **2.5 Placing the widget on a page**

You will need to add an element, in which the widget will be built, to the page. Height of the element must be specified; otherwise it may be equal to zero and the widget will not be displayed on the website.

*<****div id="forpvz" style="height****:600****px****;****"****></****div****>*

Next, you will need to create a handler to initialize the widget:

**For local version of the widget,** the handler with a minimal number of parameters will look like this:

*<****script type="text/javascript"****>* ***var widjet*** *=* ***new*** *ISDEKWidjet({* ***defaultCity****:* ***'Уфа'****,* ***cityFrom****:* ***'Омск'****,* ***link****:* ***'forpvz'*** *});  
 </****script****>*

See an example of a working local version of the widget in “example2-5.local.html” file

**For minimal version of the widget,** a minimal number of parameters will look like this:

*<****script type="text/javascript"****>* ***var widjet*** *=* ***new*** *ISDEKWidjet({* ***defaultCity****:* ***'Уфа'****,* ***cityFrom****:* ***'Омск'****,* ***link****:* ***'forpvz'****,* ***path****:* ***'***[***https://widget.cdek.ru/widget/scripts/***](https://widget.cdek.ru/widget/scripts/)***'****,* ***servicepath****:* ***'http://yoursite.net/service.php'*** *// link to “service.php” file on your web-site  
 });  
 </****script****>*

See an example of a working minimal version of the widget in “example2-5.min.html” file

# **3 The widget’s settings**

## **3.1 Example of creating the widget with all settings**

*<****script type="text/javascript"****>* ***var widjet*** *=* ***new*** *ISDEKWidjet({****showWarns****:* ***true****,* ***showErrors****:* ***true****,* ***showLogs****:* ***true****,* ***hideMessages****:* ***false****,* ***path****:* ***'***[***https://widget.cdek.ru/widget/scripts/***](https://widget.cdek.ru/widget/scripts/)***'****,* ***servicepath****:* ***'https://yoursite.net/service.php'****, //link to “service.php” file on your web-site* ***templatepath****:* ***'***[***https://widget.cdek.ru/widget/scripts/template.php***](https://widget.cdek.ru/widget/scripts/template.php)***'****,* ***choose****:* ***true****,* ***popup****:* ***true****,* ***country****:* ***'Россия'****,* ***defaultCity****:* ***'Уфа'****,* ***cityFrom****:* ***'Омск'****,* ***link****:* ***null****,* ***hidedress****:* ***true****,* ***hidecash****:* ***true****,* ***hidedelt****:* ***false****,* ***detailAddress****:* ***true****,* ***region: true,*** ***apikey****:* ***'API-key Yandex.MAP’****,* ***goods****: [{* ***length****: 10,* ***width****: 10,* ***height****: 10,* ***weight****: 1  
 }],  
 onReady: onReady,  
 onChoose: onChoose,  
 onChooseProfile: onChooseProfile,  
 onChooseAddress: onChooseAddress,  
 onCalculate: onCalculate  
 });*

***function*** *onReady() {  
 alert(****'Виджет загружен'****);  
 }*

***function*** *onChoose(wat) {  
 alert(* ***'Выбран пункт выдачи заказа '*** *+ wat.id +* ***"\n"*** *+* ***'цена '*** *+ wat.****price*** *+* ***"\n"*** *+* ***'срок '*** *+ wat.****term*** *+* ***" дн.\n"*** *+* ***'город '*** *+ wat.****cityName*** *+* ***', код города '*** *+ wat.****city****);  
 }*

***function*** *onChooseProfile(wat) {  
 alert(* ***'Выбрана доставка курьером в город '*** *+ wat.****cityName*** *+* ***', код города '*** *+ wat.****city*** *+* ***"\n"*** *+* ***'цена '*** *+ wat.****price*** *+* ***"\n"*** *+* ***'срок '*** *+ wat.****term*** *+* ***' дн.'****);  
 }*

***function*** *onChooseAddress(wat) {*

*alert(*

***'Выбрана доставка курьером по адресу ‘*** *+ wat.****address*** *+* ***”, \n “*** *+*

***'цена '*** *+ wat.****price*** *+* ***"\n"*** *+*

***'срок '*** *+ wat.****term*** *+* ***' дн.'***

*);*

*}*

***function*** *onCalculate(wat) {  
 alert(****'Расчет стоимости доставки произведен'****);  
 }  
</****script****>*

See an example of the widget’s initialization will all settings in “example3-1.html” file

## **3.2 Description of server settings**

### ***3.2.1 showWarns***

Values: true / false

Default value: true

If “true” value is set, the browser’s console will display warnings of non-critical errors in the widget’s work.

### 3.2.2 showErrors

Values: true / false

Default value: true

If “true” value is set, the browser’s console will display warnings of critical errors in the widget’s work.

### 3.2.3 showLogs

Values: true / false

Default value: true

If “true” value is set, the browser’s console will display information of all stages of the widget’s work: requests, responses, connection.

### 3.2.4 hideMessages

Values: true / false

Default value: false

A universal key to switch off all notifications of the module. Единый ключ для отключения всех оповещений модуля. “true” value is equal to setting all the above-mentioned notification settings in “false”.

### 3.2.5 path

Values: string

Default value: scripts directory in relation to the “widjet.js” file.

Thus, if your “widjet.js” file is located at<http://yoursite.net/scripts/widget/widjet.js>, the variable will direct to <http://yoursite.net/scripts/widget/scripts/>

Path to the widget’s scripts if they are located separately from the loader (widjet.js).

### 3.2.6 templatepath

Values: string

Default value: {path}/template.php

Path to the widget’s template, if it is located separately from the loader (widjet.js).

### 3.2.7 servicepath

Values: string

Default value: {path}/service.php

Path to the php file for the widget’s calculations, if it is located separately from the loader (widjet.js).

### 3.2.8 apikey

Values: string

Default value: 9720c798-730b-4af9-898a-937b264afcdd

API key for the Yandex Maps service. We recommend to follow the link below to generate a new key for your web-site. Otherwise “429 Too Many Requests” errors may occur.

<https://tech.yandex.ru/maps/jsapi/doc/2.1/quick-start/index-docpage/#get-api-key>

Important! We recommend to register your API key for the Yandex Maps so that the widget on your web-site works with your keys only and the restrictions on the free keys is applied to your web-site only.

## **3.3 Description of display settings**

### 3.3.1 choose

Values: true / false

Default value: true

Managing «Choose» button in the pick-up point’s description. If “false” is set, the button will not be displayed, which is suitable for an info-widget in the “Delivery” section. If “true” is set, the button will be displayed with an option to subscribe to select the pick-up point with the help of “onChoose” event.

### ***3.3.2 link***

Values: id элемента

Default value: false

If an option is set (not false), then the widget is considered static (always located on the page). The option’s value must be set as an id of the tag in which the widget will be placed. If the option is not set, the widget will automatically expand to the entire screen (unless the “popup” option is set).

Value of this parameter was set in almost all the examples; we set this parameter’s value as =false in example3-3-2.html.

### ***3.3.3 popup***

Values: true/false

Default value: false

If the option is set (true), the widget is considered as popup, and will be displayed via “open” method only.

See an example of this parameter’s work in example3-3-3.html

### ***3.3.4 hidedress***

Values: true/false

Default value: false

If the option is set (true), the filter of pick-up points with a try-on option will be hidden. In other words, it hides a button with this filter and all pick-up points for this city are displayed.

See an example of this parameter’s work in example3-3-4.html

### ***3.3.5 hidecash***

Values: true/false

Default value: false

If the option is set (true), the filter of pick-up points with a cashless payments option will be hidden. The pick-up points widget will be displayed without the “cashless payments” button.

See an example of this parameter’s work and hidedress in example3-3-5.html

### 3.3.6 hidedelt

Values: true/false

Default value: false

If the option is set (true), the panel with delivery options will be hidden.

See an example of this parameter’s work in example3-3-6.html

### 3.3.7 region

Values: true/false

Default value: false

If the option is set (true), pick-up points for the entire region will be displayed. For Moscow and the Moscow region, as well as for St. Petersburg and the Leningrad region, pick-up points in the city and the region are displayed. Thus, if you select the city of Gatchina (Leningrad Region), then the pick-up points in the city of Gatchina will be displayed, and pick-up points in the entire Leningrad region and St. Petersburg will be displayed on the map. At that, if you choose a pick-up point in another city (St. Petersburg, for example), then the price of delivery will be recalculated.

See an example of this parameter’s work in example3-3-7.html

### ***3.3.8 detailAddress***

Values: true/false

Default value: false

If the option is set (true), users will be offered to specify the exact delivery address after choosing courier delivery.

See an example of this parameter’s work in example3-3-8.html

there’s an opportunity to specify the address

## **3.4 Delivery settings**

### 3.4.1 defaultCity

Values: a city’s name, it is also possible to use ‘auto’ parameter, to automatically find the cient’s city via Yandex tools and show it on the map. You can use ID of the city, a number.

Default value: Moscow

The city that will be displayed on the map at the start of the widget and where the pick-up points will be loaded.

Important! The city must be located in this country when the “country” parameter is used

Important! When the “lang” parameter is used, the city’s name must be written in the same language as in the “lang” parameter. Thus, “Moscow, Astana” must be written for an English translation.

See an example of defaultCity parameter with ‘auto’ value in example3-4-1.html

### 3.4.2 lang

Values: "rus" or "eng".

Default value: "rus"

Which translation language to use in the widget – for now, only Russian and English are available.

See English edition of the widget in example3-4-2.html.

### 3.4.3 currency

Values: currency code (ISO 4217) from list: RUB, KZT, USD, EUR, GBP, CNY, BYN, UAH, KGS, AMD, TRY, THB, KRW, AED, UZS, MNT

Default value: ‘RUB’

See an example of working pick-up points list for Kazakhstan in example3-4-2.html.

### ***3.4.***4 ***country***

Values: a country's name (Russia, China (PRC), Kazakhstan etc)

Default value: all

A country, for which cities and pick-up points are selected.

See an example of working pick-up points list for Kazakhstan in example3-4-2.html.

Important! When the “lang” parameter is used, the country’s name must be written in the same language as in the “lang” parameter. Thus, “Russia, Kazakhstan” must be written for an English translation.

### ***3.4.***5 ***cityFrom***

Values: a city's name

Default value: Moscow

A city from which the parcel will be sent.  
 Important! The city must be located in this country when the “country” parameter is used

The city of Astana for Kazakhstan is shown in example3-4-2.html.

### 3.4.6 goods

Values: [{length:<длина груза >, width: <ширина груза >, height: <высота груза >, weight: <вес груза>}]

Array of the cargoes sent. Dimensions are set in centimeters (cm), weight – in kilograms (kilos).

An example of setting a parcel’s dimensions with the following characteristics: length 25 cm, width 17 cm, height 7 cm, weight 6 kilos.

**goods: [{  
 length: 25,  
 width: 17,  
 height: 7,  
 weight: 6  
 }]**

See an example of setting a parcel’s dimensions for calculation of delivery price in example3-4-4.html

## **3.5 The widget’s events**

During the operation of the widget, there are 4 types of events to which you can subscribe the handler function. It will allow receiving data from the widget: the widget loading, delivery calculation and selection of a pick-up point.

**Note: Points 3.5.1-3.5.3 include various options of creating events handlers; you can use any of them. However, *we recommend using the 3.5.1 option*; see an example of using several events handler in example3-1.htm**

### 3.5.1 Loading the widget (onReady)

The event is triggered when the widget has loaded all the styles, scripts, maps, as well as information about cities and pick-up points. It means that you can already use the widget’s methods. Please note that loading of html and maps markup may not be finished at this point.

There are no transferred parameters in the event.

To work with this event, you need to specify the name of the handler function in the parameter and describe execution of this function:  
**<script type="text/javascript">**  
 **var widjet = new** ISDEKWidjet({  
 **defaultCity: 'Омск',  
 cityFrom: 'Уфа',  
 country: 'Россия',  
 link: 'forpvz',**  
 onReady: startWidget  
 });

***function*** *startWidget() {  
 alert(****'Виджет загружен'****);  
 }  
</****script****>*

See an example of creating of a handler for this event in example3-5-1.html

### ***3.5.2 Calculation of delivery (onCalculate)***

The event is triggered when the widget has received information about delivery price and dates.

The event transfers to the handler function the following object:

{ profiles: <profiles object>, city: <city code from CDEK database>, cityName: <current city>}.

Profiles object: <profile (courier/pickup)> :{price: <price>, currency: <currency>, term: <time>, tariff: <calculated tariff>}.

Here’s an example of a such object in JSON format:

**{** **"city": "270",**  **"cityName": "Новосибирск",** **"profiles": {** **"courier": { "price": "1150", "currency": "RUB", "term": "2-3", "tarif": 11 },** **"pickup": { "price": "600", "currency": "RUB","term": "3-4", "tarif": 62 }** **}****}**

You can use the following code to create a handler:  
<**script type="text/javascript"**>  
**var *widjet*** = **new** *ISDEKWidjet*({  
**defaultCity**: **'auto'**,  
**cityFrom**: **'Уфа'**,  
**country**: **'Россия'**,  
**link**: **'forpvz'**,  
onCalculate: **function**(wat) {  
alert(**'Доставка в город '**+ wat.**cityName** +  
**"\nкурьером: "** + wat.**profiles**.**courier**.**price** + **' (тариф '**+wat.**profiles**.**courier**.**tarif** +  
**")\nдо ПВЗ: "**+wat.**profiles**.**pickup**.**price** + **' (тариф '**+wat.**profiles**.**pickup**.**tarif** +**')'**);  
**console**.log(**'Расчет доставки '**, wat);  
}  
});  
</**script**>

In this example, each delivery calculation will be accompanied with a message on the cost of delivery to the selected city indicating the selected tariff, and the text “Delivery calculation” and the contents of the object with delivery parameters will be displayed in the browser console.  
 See subscription example in example3-5-2.html

### ***3.5.3 Selecting pick-up point (onChoose)***

The event is triggered when a buyer chooses a pick-up point by pressing the “Choose” button in a detailed description of a pick-up point.

The event sends into a handler function an array {id: <id of the selected pick-up point>, PVZ: <information about pick-up point>, price: <delivery price>, currency: <delivery currency>, term: <delivery time>, tariff: <calculated tariff>, city: <selected city>}

An example of object in JSON format:  
{   
 **"id": "NSK41",** **"city": "270",** **"cityName": "Новосибирск",** **"price": "1150",** **"currency": "RUB",** **"term": "3-4",** **"tarif": 62,** **"PVZ":** {  
 **"Name": "Кольцово", "WorkTime": "Пн-Пт 10:00-19:00, Сб 10:00-16:00",** **"Address": "р.п. Кольцово, 18А", "Phone": "+79639494987, +79833160261",**  **"Note": "", "cX": "83.1851620", "cY": "54.9422560"**  
 }  
}

You can use the following code to create a handler for this event:  
 *<****script type="text/javascript"****>* ***var widjet*** *=* ***new*** *ISDEKWidjet({* ***defaultCity****:* ***'auto'****,* ***cityFrom****:* ***'Уфа'****,* ***country****:* ***'Россия'****,* ***link****:* ***'forpvz'*** *});*

***widjet****.****binders****.add(choosePVZ,* ***'onChoose'****);*

***function*** *choosePVZ(wat) {  
 alert(****'Доставка в город '*** *+ wat.****cityName*** *+* ***"\nдо ПВЗ с кодом: "*** *+ wat.id +* ***', цена '****+wat.****price*** *+****' руб.'*** *);* ***console****.log(****'Выбран ПВЗ '****, wat);  
 }  
 </****script****>*

In this example, when a pick-up point is selected (“Choose” button for a pick-up point must be displayed, that is, parameter choose=true), a message is displayed, where you can see the city, code of the pick-up point and delivery price, and a message with the text and an object with the delivery parameters will be displayed in the browser’s console.  
 See an example of creating of a handler for this event in example3-5-3.html

### 3.5.4 Selecting delivery profile (Courier) (onChooseProfile)

The event is triggered when the buyer chooses courier delivery to a particular city.

The event sends into a handler function an array {id: <courier>, price: <delivery price>, currency: <delivery currency>, term: <delivery time>, tariff: <calculated tariff>, city: <selected city>}

An example of object transferred to the function in JSON format:  
**{  
 "id": "courier",  
 "city": "270",  
 "cityName": "Новосибирск",  
 "price": "1150",  
 "currency": "RUB",  
 "term": "2-3",  
 "tarif": 11  
}**  
See an example of creating a handler for this event below. In the example, a message is displayed in which the city, delivery price and time are indicated, and a message with text and an object with delivery parameters is displayed in the browser console.  
**<script type="text/javascript">**  
 **var widjet** = **new** ISDEKWidjet({  
 **defaultCity: 'auto',  
 cityFrom: 'Уфа',  
 country: 'Россия',  
 link: 'forpvz',  
 hidedress: true,  
 hidecash: true,**  
 onChooseProfile: onChooseProfile  
 });

***function*** *onChooseProfile(wat) {  
 alert(****'Выбрана доставка курьером в город '*** *+ wat.****cityName*** *+* ***"\n"*** *+* ***'цена '*** *+ wat.****price*** *+* ***"\n"*** *+* ***'срок '*** *+ wat.****term*** *+* ***' дн.'*** *);* ***console****.log(****'Выбрана доставка курьером '****, wat);  
 }  
 </****script****>*

See the entire code of an example of creating of a handler for this event in example3-5-4.html

### ***3.5.***5 ***Selecting delivery address (Courier) (onChooseAddress)***

The event is triggered when the buyer chooses courier delivery to a concrete delivery address.

The event sends into a handler function an array {id: <courier>, price: <delivery price>, currency: <delivery currency>, term: <delivery time>, tariff: <calculated tariff>, city: <selected city>, cityName: <name of selected city>, address: <selected delivery address>}

An example of object transferred to the function in JSON format:  
**{  
 "id": "courier",  
 "city": "270",  
 "cityName": "Новосибирск",  
 "price": "1150",**

**"currency": "RUB",  
 "term": "2-3",  
 "tarif": 11,  
 "address": "Россия, Новосибирск, Большевистская улица, 101",  
}**

See an example of creating a handler for this event below. In the example, a message is displayed in which the city, delivery price and time are indicated, and a message with text and an object with delivery parameters is displayed in the browser console.  
 *<****script type="text/javascript"****>* ***var widjet*** *=* ***new*** *ISDEKWidjet({* ***defaultCity****:* ***'auto'****,* ***cityFrom****:* ***'Уфа'****,* ***country****:* ***'Россия'****,* ***link****:* ***'forpvz'****,* ***hidedress****:* ***true****,* ***hidecash****:* ***true****,*

***detailAddress****:* ***true****,  
 onChooseAddress: onChooseAddress  
 });*

***function*** *onChooseAddress(wat) {  
 alert(****'Выбрана доставка курьером по адресу '*** *+ wat.****address*** *+* ***"\n"*** *+* ***'цена '*** *+ wat.****price*** *+* ***"\n"*** *+* ***'срок '*** *+ wat.****term*** *+* ***' дн.'*** *);* ***console****.log(****'Выбрана доставка курьером '****, wat);  
 }  
 </****script****>*

See the entire code of an example of creating of a handler for this event in example3-5-5.html

## **3.6 The widget’s methods**

### ***3.6.1 Operations with cities***

#### **city.get()**

Returns the identifier of the current city which is displayed on the map.

#### **city.set(city)**

Sets “city” city as current. Can take both the city name and the city identifier as a parameter.

#### **city.check(city)**

Checks if the city “city” is available and returns its identifier if successful.  
See an example of the widget’s work and operations with cities in example3-6-1.html

### ***3.6.2 Operations with parcels***

Requirements for the transferred object (params) are unified for setting goods/parcels:

* *length –* cargo’s length (cm),
* *width –* cargo's width (cm),
* height – cargo's height (cm),
* weight – cargo’s weight (kilos)

#### **сargo.add (params)**

Adds a cargo with the indicated parameters (params) to the parcel without resetting it to zero. Price will be re-calculated automatically.

#### **cargo.get()**

Returns data on goods (all available fields) in the form of an object, the format description of which looks as follows in JSON:  
***{*** ***"0":{"length":"15","width":"10","height":"10","weight":"0.4"},*** ***"1":{"length":"25","width":"7","height":"17","weight":"2.4"}******}***

#### cargo.reset()

Resets all information about cargos.

See examples of operations with parcels (add, receive current information, reset parcels’ parameters to calculate delivery price) in example3-6-2.html

### ***3.6.3 User interface***

#### **open()**

Displays the widget if it is in the popup mode (unless options “link” and “popup=true” are set).

#### **close()**

Closes the widget if it is in the popup mode (unless options “link” and “popup=true” are set).  
See an example of work with the widget display in example3-6-3.html

# **4 Examples of work**

Now, we will combine the widget’s parameters to create a widget with a particular purpose.

## **4.1 Creating an info-widget**

An info-widget is a widget designed to inform clients on a possibility of delivery in a particular city, with displaying information about pick-up points.

*<****script type="text/javascript"****>* ***var infoWidjet*** *=* ***new*** *ISDEKWidjet ({* ***defaultCity****:* ***'Барнаул'****, // The city of Barnaul is default* ***cityFrom****:* ***'Санкт-Петербург'****, // delivered from St.Petersburg* ***choose****:* ***false****, // hide the “Choose” button for pick-up points* ***link****:* ***'infowidjet'*** *// link the widget to the indicated element  
 </****script****>  
 <****div id='infowidjet'****>The widget will appear here after loading </****div****>*

Note that in this example, the parameters of the element, in which the widget is built, are indicated in CSS-styles. See full text of this widget in example4-1.html

## **4.2 Widget for a product page**

For the product’s page, we will create a widget similar to an info-widget: it is impossible to choose a pick-up point or type of delivery; however, in order not to clutter the page, it will be displayed in a pop-up window.

*<****script type="text/javascript"****>* ***var cartWidjet*** *=* ***new*** *ISDEKWidjet ({* ***cityFrom****:* ***'Екатеринбург'****, delivered from Yekaterinburg* ***choose****:* ***false****, // hide the “Choose” button for pick-up points* ***hidedelt****:* ***true****, // hide an option of choosing delivery type* ***goods****: [ // set information about the product so that delivery price is calculated correctly  
 {* ***length****: 20,* ***width****: 20,* ***height****: 20,* ***weight****: 2 }  
 ],* ***popup****:* ***true****, // switch on popup mode  
 onCalculate: calculated  
 });  
 // configure the widget so that information in the block with delivery details was updated after calculation of price of delivery to a pick-up point* ***function*** *calculated(params){* ***ipjq****(****'#delPrice'****).****html****(params.****profiles****.****pickup****.****price****);* ***ipjq****(****'#delTime'****).****html****(params.****profiles****.****pickup****.****term****);  
 }  
 </****script****>   
 Put a block with delivery information here:  
 <****div****>  
 Delivery price: <****span id='delPrice'****>loadin</****span****> руб.<****br****>  
 Delivery time: <****span id='delTime'****>грузится</****span****> дн.<****br****>  
 <****a href='javascript:void(0)' onclick='cartWidjet****.open()****'****>Показать ПВЗ</****a****>  
 </****div****>*

See full code of this example in example4-2.html

## **4.3 Widget for ordering**

The main feature of the widget for ordering is the option of selecting a pick-up point to save the information about the choice, as well as the price and delivery time.  
 In addition, in this example, we set the minimum price of delivery equal to 500 rubles. If the price of delivery is more than 500, then we round it up to a hundred.  
 *<****script type="text/javascript"****>* ***var orderWidjet*** *=* ***new*** *ISDEKWidjet({* ***popup****:* ***true****,* ***defaultCity****:* ***'Казань'****,* ***cityFrom****:* ***'Казань'****,* ***goods****: [ // set information about the goods in the shopping bag  
 {* ***length*** *: 10,* ***width*** *: 20,* ***height*** *: 20,* ***weight*** *: 5 }  
 ],  
 onReady :* ***function****(){ // display information about delivery to a pick-up point for the widget’s loading* ***ipjq****(****'#linkForWidjet'****).css(****'display'****,****'inline'****);  
 },  
 onChoose :* ***function****(info){ // when choosing a pick-up point: enter information in the text fiields* ***ipjq****(****'[name="chosenPost"]'****).val(info.id);* ***ipjq****(****'[name="addresPost"]'****).val(info.****PVZ****.Address);  
 // calculating delivery price* ***var*** *price = (info.****price*** *< 500) ? 500 :* ***Math****.ceil( info.****price****/100 ) \* 100;* ***ipjq****(****'[name="pricePost"]'****).val(price);* ***ipjq****(****'[name="timePost"]'****).val(info.****term****);* ***orderWidjet****.close(); // close the widget  
 }   
 });  
 </****script****>  
 <****p****> <****a href='javascript:void(0)' onclick='orderWidjet****.open()****'****>Выбрать ПВЗ</****a****> </****p****>  
 <****div id="linkForWidjet" style="display****:* ***none****;****"****>  
 <****p****>Pick-up point chosen: <****input type='text' name='chosenPost' value=''****/></****p****>  
 <****p****>Pick-up point’s address: <****input type='text' name='addresPost' value=''****/></****p****>  
 <****p****>Delivery price: <****input type='text' name='pricePost' value=''****/></****p****>  
 <****p****>Approximate delivery time: <****input type='text' name='timePost' value=''****/></****p****>  
 </****div****>*

See full code of this example in example4-3.html

## **4.4 Widget for storefront**

Consider a storefront of a web-site where you can add products to the basket and information about the total price of delivery has to be updated.  
 *<****script type="text/javascript"****>* ***var sectionWidjet*** *=* ***new*** *ISDEKWidjet ({* ***link****:* ***"forpvz"****,* ***defaultCity****:* ***'Казань'****, // suppose the client is in the city of Kazan* ***cityFrom****:* ***"Бердск"****,  
 onCalculate: calculated  
 });  
 // configure the widget so that information in the block with delivery details was updated after calculation of price of delivery* ***function*** *calculated(params){* ***ipjq****(****'#delPricePVZ'****).****html****(params.****profiles****.****pickup****.****price*** *+* ***" руб."****);* ***ipjq****(****'#delPriceCourier'****).****html****(params.****profiles****.****courier****.****price*** *+* ***" руб."****);  
 }  
 // write a handler to add goods to the shopping bag. Let it take an object { gabs:<габариты>,weight:<вес>}* ***function*** *goodAdded(params){* ***sectionWidjet****.****cargo****.add ({* ***length****: params.****gabs****[0],* ***width****: params.****gabs****[1],* ***height****: params.****gabs****[2],* ***weight****: params.****weight*** *});  
 }  
 </****script****>  
<****div****>Price of delivery of your shopping bag to a pick-up point: <****span id='delPricePVZ'****>calculated…</****span****></****div****>  
<****div****>Price of courier delivery of your shopping bag: <****span id='delPriceCourier'****>calculated…</****span****></****div****>  
<****button onclick="****goodAdded({****gabs****:[20,30,10],****weight****:2.5})****"*** *>To the shopping bag</****button****>  
<****div id="forpvz"****></****div****>*

See a detailed example in example4-4.html