

# Вывести на карту пункты выдачи доставки СДЭК

1. В ТДС магазина передаем файл, в котором хранятся данные пунктов выдачи выбранного города

сдек.xml

```
$cityName = $_SESSION['shop_country_location_city'];  
$feed = simplexml_load_file('sdek.xml');  
$arResult = $feed->xpath("/PvzList/Pvz[@City = '$city_name1']");
```

2. Передаем в XML нужные нам данные для дальнейшего вывода на страницу.

```
$i = 1;  
foreach ($arResult as $City)  
{  
    $dName[$i] = ($City['Name']);  
    $dWorkTime[$i] = $City['WorkTime'];  
    $dAddress[$i] = $City['Address'];  
    $dPhone[$i] = $City['Phone'];  
    $dCoordX[$i] = $City['coordX'];  
    $dCoordY[$i] = $City['coordY'];  
  
    $Shop_Controller_Show  
    ->addEntity(  
        Core::factory('Core_Xml_Entity')  
            ->name('dName')  
            ->value($dName[$i])  
    )  
    ->addEntity(  
        Core::factory('Core_Xml_Entity')  
            ->name('dWorkTime')  
            ->value($dWorkTime[$i])  
    )  
    ->addEntity(  
        Core::factory('Core_Xml_Entity')  
            ->name('dAddress')  
            ->value($dAddress[$i])  
    )  
    ->addEntity(  
        Core::factory('Core_Xml_Entity')  
            ->name('dPhone')  
            ->value($dPhone[$i])  
    )  
    ->addEntity(  
        Core::factory('Core_Xml_Entity')  
            ->name('dCoordX')
```

```

        ->value($dCoordX[$i])
    )
    ->addEntity(
        Core::factory('Core_Xml_Entity')
        ->name('dCoordY')
        ->value($dCoordY[$i])
    );

    $i++;
    $Shop_Controller_Show
    ->addEntity(
        Core::factory('Core_Xml_Entity')
        ->name('i')
        ->value($i)
    );
}

$Shop_Controller_Show
->addEntity(
    Core::factory('Core_Xml_Entity')
        ->name('dcity_name')
        ->value($cityName)
);

```

3. В xsl вставляем в нужное место код для всплывающего окна.

```

<div class="block">
    <a href="#sdek_map" id="sdek_map" class="calculator" >
        Рассчитать доставку</a>
</div>

```

4. Вывод всплывающего окна с картой пунктами выдачи. Высплывающее окно можно сделать с помощью Fancybox [Пункт 4](#).

```

<div id="sdek_map" style="display: none">
    <form id="sdek_maps" method="POST" action=".">
        <h1 style="color: black">Пункты самовывоза в вашем городе</h1>
        <div class="container_map">
            <div class="delivery_map">
                <br /><h3>Доставка:</h3><hr /><ul><xsl:call-template name="dfor">
                    <xsl:with-param name="n" select="/shop/i"/>
                </xsl:call-template></ul>
            </div>
            <div class="map">
                <div id="map"></div>
            </div>
            <div class="text_map">
                <xsl:value-of disable-output-escaping="yes"
select="/shop/documen_sdek"/>
            </div>
        </div>
    </div>

```

```
</form>
</div>
```

5. Вставляем в xsl скрипт вывода карты.

```
<xsl:variable name="city" select="/shop/dcity_name" />
<xsl:choose>
<xsl:when test="/shop/dCoordY != ''">
<script type="text/javascript">
$(document).ready(function(){
    var mapLoaded = false;
    $(".calculator").fancybox().click(function ()
    {
        if(!mapLoaded){
            ymaps.ready(init);
        }
        function init () {
            mapLoaded = true;
            var myMap = new ymaps.Map("map", {
                center: [<xsl:value-of select="/shop/dCoordY"/>, <xsl:value-of
select="/shop/dCoordX"/>],
                zoom: 10,
                controls: ['zoomControl', 'fullscreenControl']
            }, {
                suppressMapOpenBlock: true,
                searchControlProvider: 'yandex#search'
            }),
            placemarks = [],
            <xsl:call-template name="placemark_for">
            <xsl:with-param name="n" select="/shop/i"/>
            </xsl:call-template>

            // Поиск координат центра .
            ymaps.geocode('<xsl:value-of select="$city"/>', {
                /**
            * Опции запроса
            * @see
            https://api.yandex.ru/maps/doc/jsapi/2.1/ref/reference/geocode.xml
            */
            // Сортировка результатов от центра окна карты.
            //   boundedBy: myMap.getBounds(),
            //   strictBounds: true,
            // Вместе с опцией boundedBy будет искать строго внутри области, указанной в
            boundedBy.
            // Если нужен только один результат, экономим трафик пользователей.
            results: 1
            }).then(function (res) {
            // Выбираем первый результат геокодирования.
            var firstGeoObject = res.geoObjects.get(0),
            // Координаты геообъекта.
            coords = firstGeoObject.geometry.getCoordinates(),
```

```

// Область видимости геообъекта.
    bounds = firstGeoObject.properties.get('boundedBy');
// Добавляем первый найденный геообъект на карту.
    // myMap.geoObjects.add(firstGeoObject);
// Масштабируем карту на область видимости геообъекта.
    myMap.setBounds(bounds, {
// Проверяем наличие тайлов на данном масштабе.
    checkZoomRange: true,
    });
});
//открытие закрытие информации метки
$('.delivery_map a').each(function (index)
{
$(this).bind('click', function ()
{
if (!placemarks[index].balloon.isOpen()) {
placemarks[index].balloon.open();
} else {
placemarks[index].balloon.close();
}
return false;
});
$(this)
.bind('mouseenter', function () {
// Ссылку на объект, вызвавший событие,
// можно получить из поля 'target'.
placemarks[index].options.set('preset', 'islands#greenIcon');
})
.bind('mouseleave', function () {
placemarks[index].options.unset('preset');
});
});

//Выделение активной ссылке в списке
$(".delivery_map li").click(function () {
if(!$(this).hasClass('active'))
{
$(".delivery_map li").removeClass('active');
$(this).toggleClass('active');
}
else{
$(this).toggleClass('active');
}
});
});
});
</script>
</xsl:when>
<xsl:otherwise>
    <script type="text/javascript">

```

```

//Скрипт вывода карты местности(города который определили) если пунктов выдачи в нем
нету нету.
$(document).ready(function(){
    var mapLoaded = false;
    $(".calculator").fancybox().click(function ()
    {
//Загрузка карты и показ пунктов выдачи на карте
        if(!mapLoaded)
        {
            ymaps.ready(init);
        }
        function init () {
            mapLoaded = true;

// Поиск координат центра
            ymaps.geocode('<xsl:value-of select="$city"/>', {
                results: 1
            }).then(function (res) {
                var firstGeoObject = res.geoObjects.get(0),
                coords = firstGeoObject.geometry.getCoordinates();
                var myMap = new ymaps.Map('map', {
                    center: coords,
                    zoom: 10
                });
                bounds = firstGeoObject.properties.get('boundedBy');
                //myMap.geoObjects.add(firstGeoObject);
                myMap.setBounds(bounds, {
                    checkZoomRange: true
                });
            });
        }
    });
</script>
</xsl:otherwise>
</xsl:choose>

```

6.В список пунктов доставки выводим название

```

<xsl:template name="dfor">
    <xsl:param name="i" select="1"/>
    <xsl:param name="n"/>
    <xsl:if test="$i < $n">
        <a href="" ><li>
            <xsl:value-of disable-output-escaping="yes"
select="/shop/dName[$i]"/>
        </li></a>
        <xsl:call-template name="dfor">
            <xsl:with-param name="i" select="$i + 1"/>
            <xsl:with-param name="n" select="$n"/>
        </xsl:call-template>
    }

```

```
</xsl:if>
</xsl:template>
```

## 7. Получаем данные координат и данные для метки на карту из xml

```
<xsl:template name="placemark_for">
  <xsl:param name="i" select="1"/>
  <xsl:param name="n"/>
  <xsl:if test="$i < $n">
    myPlacemark = new ymaps.Placemark([<xsl:value-of
select="/shop/dCoordY[$i]"/>, <xsl:value-of select="/shop/dCoordX[$i]"/>], {
  // Чтобы балун и хинт открывались на метке, необходимо задать ей определенные
свойства.
    balloonContentHeader: " ",
    balloonContentBody: "<table>
      <tr>
        <td>
          <xsl:value-of disable-output-escaping="yes"
select="/shop/dAddress[$i]"/>
        </td>
      </tr>
      <tr>
        <td>
          <xsl:value-of disable-output-escaping="yes"
select="/shop/dPhone[$i]"/>
        </td>
      </tr>
      <tr>
        <td>
          <xsl:value-of disable-output-escaping="yes"
select="/shop/dWorkTime[$i]"/>
        </td>
      </tr>
    </table>",
    balloonContentFooter: " ",
    hintContent: "<xsl:value-of disable-output-escaping="yes"
select="/shop/dName[$i]"/>"
    });
    myMap.geoObjects.add(myPlacemark);
    placemarks.push(myPlacemark);
    <xsl:call-template name="placemark_for">
      <xsl:with-param name="i" select="$i + 1"/>
      <xsl:with-param name="n" select="$n"/>
    </xsl:call-template>
  </xsl:if>
</xsl:template>
```