

PHP

CMS, CMS, CMS: "

•

•



IntellectMoneyCommon.zip

• •

— *

— *

— *

□

- Currency.php - . ;
- Customer.php - - ;
- IntellectMoneyBase.php - , . ;
- Item.php - - . ;
- LanguageHelper.php - , ;
- MerchantReceiptHelper.php - . ;
- Order.php - - ;
- Payment.php - - , ;
- ProcessingResponseResultStruct.php - - . ;
- Result.php - - ;
- Status.php - , , ;
- StatusWeight.php - , . ;
- UserSettings.php - - . ;
- VATs.php - . ;
- Validator.php - , . ;
- Exceptions - - , ;
- Languages - , - 2 : . .

LanguageHelper

LanguageHelper , .

```
get_instance($lang = 'ru', $isCp1251 = false). $lang    'ru', ru - Languages, -. $isCp1251 - , ,
cp1251.
```

```
setLanguage($lang, $isCp1251 = null), . , ( ).
```

```
getTitle($name), getDesc($name) getError($name) , . $name , Languages.
```

getLanguage() .

UserSettings

- LanguageHelper
- UserSettings
- Payment
- Order
- Customer
- Result
- ProcessingResponse
- ResultStruct
- VATs
- - CMS CMS
OpenCart (MVC)
- CMS OpenCart
(MVC)
- CMS OpenCart (MVC)

UserSettings , , , , .

getInstance(\$params = array(), \$isAllowChangeValues = true), \$params - ,
\$isAllowChangeValues .
setParams(\$params, &\$errors = array())

```
array(  
    'eshopId' => '456789',  
    'secretKey' => '12345678',  
    ...  
);
```

\$errors , .

```
array(  
    'eshopId',  
    'group',  
    ...  
);
```

, , ., () .

setIsAllowChangeValues(\$allow = false) , , , , . true.

getParams() , .

getIsAllowChangeValues() \$isAllowChangeValues.

resetParams() , , .

get() , , , , , getNamesOfOrganizationParamsToSave(), - getNamesOfP2PPParamsToSave().

set(\$value) , , \$isAllowChangeValues true. . .

Payment

Payment.

getInstance(\$UserSettings = NULL, \$Order = NULL, \$Customer = NULL, \$lang = 'ru') . \$UserSettings
- UserSettings, \$Order - \$Order, \$Customer - Customer. \$lang (, LanguageHelper:::
getInstance(\$lang));

generateForm(\$isOrganization = true, \$isAutoSubmit = false) , , UserSettings, Order Customer,
getInstance(). \$isOrganization IntellectMoney: . . . \$isAutoSubmit javascript- .

getParamsForOrganization() , , .

getParamsForP2P() , , .

Order

Order.

getInstance(\$invoiceId = NULL, \$orderId = NULL, \$originalAmount = NULL, \$recipientAmount = NULL,
\$paidAmount = NULL, \$deliveryAmount = NULL, \$recipientCurrency = NULL, \$discount = NULL, \$status
= NULL) Order. :
\$invoiceId - IntellectMoney. ;
\$orderId - CMS;
\$originalAmount - . \$recipientAmount;
\$recipientAmount - . \$originalAmount;
\$paidAmount - ;
\$deliveryAmount - ;
\$recipientCurrency - . TST. , TestMode UserSettings true;
\$discount - . , .
\$status - IntellectMoney.

```
setParams($invoiceId = NULL, $orderId = NULL, $originalAmount = NULL, $recipientAmount = NULL, $paidAmount = NULL, $deliveryAmount = NULL, $recipientCurrency = NULL, $discount = NULL, $status = NULL) , getInstance().
```

```
addItem($price, $quantity, $title, $tax = 0) . :  
$price - ;  
$quantity - ;  
$title - ;  
$tax - . UserSettings->getTax().
```



```
. UserSettings->getDeliveryTax().
```

```
getItems() .  
getParams() , , getItems().  
get() , . getInstance().  
set() , .  
resetParams() , .
```

Customer

Customer.

```
getInstance($email = NULL, $name = NULL, $phone = NULL) Customer. :  
$email - , .  
$name - .  
$phone - .  
  
get() , . getInstance().  
set() , .  
resetParams() , .
```

Result

Result .

```
getInstance($request = array(), $UserSettings = "", $Order = "", $lang = 'ru', $isCp1251 = false)  
Result. :  
$request - $_REQUEST;  
$UserSettings - UserSettings;  
$Order - Order;  
$lang - ( , LanguageHelper::getInstance($lang));  
$isCp1251 - cp1251.  
  
setParams($request, $userSettings, $order, $lang = 'ru', $isCp1251 = false) , processingResponse().  
 , getInstance().  
  
resetParams() , getInstance() setParams().  
  
processingResponse() . ProcessingResponseResultStruct.  
  
getMessage() "" , , . IntellectMoney. :
```

getMessage()

```
echo Result->getMessage();  
die();
```

ProcessingResponseResultStruct

```
ProcessingResponseResultStruct - . :  
$changeStatusResult - , -.  
$statusCMS - CMS, - IntellectMoney. .
```

VATs

VATs- .

getList() , - CMS.

- CMS CMS OpenCart (MVC)

View twig

```
{{ header }}{{ column_left }}
<div id="content">
    <div class="page-header">
        <div class="container-fluid">
            <div class="pull-right">
                <button type="submit" form="form-intellectmoney" class="btn btn-primary"><i class="fa fa-
check-circle"></i> {{ button_save }}</button>
                <a href="{{ cancel }}" class="btn btn-default"><i class="fa fa-reply"></i> {{ button_cancel
}}</a></div>
                <h1 style="background:url('/admin/view/image/payment/intellectmoney.png') center left;
background-repeat: no-repeat; background-size: 28%; padding: 10px 0 10px 90px;">{{ heading_title }}</h1>
                <ul class="breadcrumb">
                    {% for breadcrumb in breadcrumbs %}
                        <li><a href="{{ breadcrumb.href }}">{{ breadcrumb.text }}</a></li>
                    {% endfor %}
                </ul>
            </div>
        </div>
    <div class="container-fluid">
        {% if error_warning %}
            <div class="alert alert-danger"><i class="fa fa-exclamation-circle"></i> {{ error_warning }}
            <button type="button" class="close" data-dismiss="alert">x</button>
            </div>
        {% endif %}
        <div class="panel panel-default">
            <div class="panel-heading">
                <h3 class="panel-title"><i class="fa fa-pencil"></i> {{ text_edit }}</h3>
            </div>
            <div class="panel-body">
                <form action="{{ action }}" method="post" enctype="multipart/form-data" id="form-
intellectmoney" class="form-horizontal">

                    <div class="form-group required">
                        <label class="col-sm-2 control-label" for="payment_intellectmoney_eshopId">{{
entry_eshopId }}</label>
                        <div class="col-sm-10">
                            <input type="text" name="payment_intellectmoney_eshopId" value="{{
payment_intellectmoney_eshopId }}" class="form-control" id="payment_intellectmoney_eshopId" />
                            {% if error_eshopId %}
                                <div class="text-danger">{{ error_eshopId }}</div>
                            {% endif %}
                        </div>
                    </div>

                    <div class="form-group required">
                        <label class="col-sm-2 control-label" for="payment_intellectmoney_secretKey">{{
entry_secretKey }}</label>
                        <div class="col-sm-10">
                            <input type="text" name="payment_intellectmoney_secretKey" value="{{
payment_intellectmoney_secretKey }}" class="form-control" id="payment_intellectmoney_secretKey" />
                            {% if error_secretKey %}
                                <div class="text-danger">{{ error_secretKey }}</div>
                            {% endif %}</td>
                        </div>
                    </div>

                    <div class="form-group">
                        <label class="col-sm-2 control-label" for="payment_intellectmoney_total">{{
```

```

entry_total }}</label>
        <div class="col-sm-10">
            <input type="text" name="payment_intellectmoney_total" value="{{
payment_intellectmoney_total }}" class="form-control" id="payment_intellectmoney_total" />
        </div>
    </div>

    <div class="form-group required">
        <label class="col-sm-2 control-label" for="payment_intellectmoney_resultUrl">{{
entry_resultUrl }}</label>
        <div class="col-sm-10">
            <div class="input-group"><span class="input-group-addon"><i class="fa fa-link"><
/i></span>
                <input type="text" readonly="readonly" value="{{
payment_intellectmoney_resultUrl }}" id="payment_intellectmoney_resultUrl" class="form-control">
                {% if error_resultUrl %}
                    <div class="text-danger">{{ error_resultUrl }}</div>
                {% endif %}
            </div>
        </div>
    </div>

    <div class="form-group">
        <label class="col-sm-2 control-label" for="payment_intellectmoney_testMode">{{
entry_testMode }}</label>
        <div class="col-sm-10">
            <input type="checkbox" name="payment_intellectmoney_testMode" {% if
payment_intellectmoney_testMode %}checked {% endif %} class="form-control" id="
payment_intellectmoney_testMode" />
        </div>
    </div>

    <div class="form-group">
        <label class="col-sm-2 control-label" for="payment_intellectmoney_holdMode">{{
entry_holdMode }}</label>
        <div class="col-sm-10">
            <input type="checkbox" name="payment_intellectmoney_holdMode" {% if
payment_intellectmoney_holdMode %}checked {% endif %} class="form-control" id="
payment_intellectmoney_holdMode" />
        </div>
    </div>

    <div class="form-group">
        <label class="col-sm-2 control-label" for="payment_intellectmoney_holdTime">{{
entry_holdTime }}</label>
        <div class="col-sm-10">
            <input type="text" name="payment_intellectmoney_holdTime" value="{{
payment_intellectmoney_holdTime }}" class="form-control" id="payment_intellectmoney_holdTime" />
            {% if error_holdTime %}
                <div class="text-danger">{{ error_holdTime }}</div>
            {% endif %}
        </div>
    </div>

    <div class="form-group">
        <label class="col-sm-2 control-label" for="payment_intellectmoney_expireDate">{{
entry_expireDate }}</label>
        <div class="col-sm-10">
            <input type="text" name="payment_intellectmoney_expireDate" value="{{
payment_intellectmoney_expireDate }}" class="form-control" id="payment_intellectmoney_expireDate" />
            {% if error_expireDate %}
                <div class="text-danger">{{ error_expireDate }}</div>
            {% endif %}
        </div>
    </div>

    <div class="form-group">
        <label class="col-sm-2 control-label" for="payment_intellectmoney_successUrl">{{
entry_successUrl }}</label>
        <div class="col-sm-10">
            <input type="text" name="payment_intellectmoney_successUrl" value="{{

```

```

payment_intellectmoney_successUrl }}" class="form-control" id="payment_intellectmoney_successUrl" />
    {% if error_successUrl %}
        <div class="text-danger">{{ error_successUrl }}</div>
    {% endif %}
</div>
</div>

<div class="form-group">
    <label class="col-sm-2 control-label" for="payment_intellectmoney_preference">{{
entry_preference }}</label>
    <div class="col-sm-10">
        <input type="text" name="payment_intellectmoney_preference" value="{{
payment_intellectmoney_preference }}" class="form-control" id="payment_intellectmoney_preference" />
        {% if error_preference %}
            <div class="text-danger">{{ error_preference }}</div>
        {% endif %}
    </div>
</div>

<div class="form-group">
    <label class="col-sm-2 control-label" for="payment_intellectmoney_tax">{{ entry_tax
}}</label>
    <div class="col-sm-10">
        <select name="payment_intellectmoney_tax" id="payment_intellectmoney_tax" class="
form-control">
            {% for VAT in VATs %}
                <option value="{{ VAT.id }}" {% if VAT.id == payment_intellectmoney_tax
%} selected="selected" {% endif %}>{{ VAT.name }}</option>
            {% endfor %}
        </select>
    </div>
</div>

<div class="form-group">
    <label class="col-sm-2 control-label" for="payment_intellectmoney_deliveryTax">{{
entry_deliveryTax }}</label>
    <div class="col-sm-10">
        <select name="payment_intellectmoney_deliveryTax" id="
payment_intellectmoney_deliveryTax" class="form-control">
            {% for VAT in VATs %}
                <option value="{{ VAT.id }}" {% if VAT.id ==
payment_intellectmoney_deliveryTax %} selected="selected" {% endif %}>{{ VAT.name }}</option>
            {% endfor %}
        </select>
    </div>
</div>

<div class="form-group">
    <label class="col-sm-2 control-label" for="payment_intellectmoney_group">{{
entry_group }}</label>
    <div class="col-sm-10">
        <input type="text" name="payment_intellectmoney_group" value="{{
payment_intellectmoney_group }}" class="form-control" id="payment_intellectmoney_group" />
        {% if error_group %}
            <div class="text-danger">{{ error_group }}</div>
        {% endif %}
    </div>
</div>

    {% for statusIm in statusesIm %}
        <div class="form-group required">
            <label class="col-sm-2 control-label" for="{{ statusIm.dataName }}">{{ statusIm.
name }}</label>
            <div class="col-sm-10">
                <select name="{{ statusIm.dataName }}" id="{{ statusIm.dataName }}" class="
form-control">
                    {% for order_status in order_statuses %}
                        <option value="{{ order_status.order_status_id }}" {% if
order_status.order_status_id == statusIm.value %} selected="selected" {% endif %}>{{ order_status.name }}<
/option>
                    {% endfor %}
                </select>
            </div>
        </div>
    {% endfor %}

```

```

        </select>
    </div>
</div>
{% endfor %}

<div class="form-group">
    <label class="col-sm-2 control-label" for="payment_intellectmoney_geo_zone_id">{{
entry_geo_zone }}</label>
    <div class="col-sm-10">
        <select name="payment_intellectmoney_geo_zone_id" id="
payment_intellectmoney_geo_zone_id" class="form-control">
            <option value="0">{{ text_all_zones }}</option>
            {% for geo_zone in geo_zones %}
                {% if geo_zone.geo_zone_id == payment_intellectmoney_geo_zone_id %}
                    <option value="{{ geo_zone.geo_zone_id }}" selected="selected">{{
geo_zone.name }}</option>
                {% else %}
                    <option value="{{ geo_zone.geo_zone_id }}">{{ geo_zone.name }}<
/option>
                {% endif %}
            {% endfor %}
        </select>
    </div>
</div>

<div class="form-group required">
    <label class="col-sm-2 control-label" for="payment_intellectmoney_status">{{
entry_status }}</label>
    <div class="col-sm-10">
        <select name="payment_intellectmoney_status" id="payment_intellectmoney_status"
class="form-control">
            <option value="1" {% if payment_intellectmoney_status == 1 %} selected="
selected" {% endif %}>{{ text_enabled }}</option>
            <option value="0" {% if payment_intellectmoney_status == 0 %} selected="
selected" {% endif %}>{{ text_disabled }}</option>
        </select>
    </div>
</div>

<div class="form-group">
    <label class="col-sm-2 control-label" for="payment_intellectmoney_sort_order">{{
entry_sort_order }}</label>
    <div class="col-sm-10">
        <input type="text" name="payment_intellectmoney_sort_order" value="{{
payment_intellectmoney_sort_order }}" class="form-control" id="payment_intellectmoney_sort_order" />
    </div>
</div>
</form>
</div>
</div>
</div>
{{ footer }}

```

Controller PHP

```

<?php

class ControllerExtensionPaymentIntellectmoney extends Controller {

    private $error = array();

    public function index() {
        $this->load->language('extension/payment/intellectmoney');

        if (method_exists($this->document, 'setTitle'))
            $this->document->setTitle($this->language->get('heading_title'));
        else

```

```

        $this->document->title = $this->language->get('heading_title');

        $this->load->model('setting/setting');
        $this->load->model('IntellectMoney/intellectmoney_model');
        $im_lang = $this->model_IntellectMoney_intellectmoney_model->getLanguageHelperInstance($this->language->data['code']);
        $im_userSettings = $this->model_IntellectMoney_intellectmoney_model->getUserSettingsInstance();

        if (($this->request->server['REQUEST_METHOD'] == 'POST') && ($this->validate($im_lang, $im_userSettings))) {
            foreach ($this->request->post as &$value) {
                $value = htmlspecialchars_decode($value);
            }

            $this->model_setting_setting->editSetting('payment_intellectmoney', $this->request->post);
            $this->session->data['success'] = $this->language->get('text_success');

            $this->response->redirect($this->url->link('marketplace/extension', 'user_token=' . $this->session->data['user_token'] . '&type=payment', true));
        }

        //map breadcrums
        $data['breadcrumbs'] = array();
        $data['breadcrumbs'][] = array(
            'text' => $this->language->get('text_home'),
            'href' => $this->url->link('common/dashboard', 'user_token=' . $this->session->data['user_token'], true)
        );
        $data['breadcrumbs'][] = array(
            'text' => $this->language->get('text_payment'),
            'href' => $this->url->link('marketplace/extension', 'user_token=' . $this->session->data['user_token'], true)
        );
        $data['breadcrumbs'][] = array(
            'text' => $this->language->get('heading_title'),
            'href' => $this->url->link('extension/payment/intellectmoney', 'user_token=' . $this->session->data['user_token'], true)
        );
        $data['action'] = $this->url->link('extension/payment/intellectmoney', 'user_token=' . $this->session->data['user_token'], true);
        $data['cancel'] = $this->url->link('marketplace/extension', 'user_token=' . $this->session->data['user_token'] . '&type=payment', true);

        //map Titles
        foreach ($im_userSettings->getNamesOfOrganizationParamsToSave() as $value) {
            $data['entry_' . $value] = $im_lang->getTitle($value);
        }
        $data['entry_total'] = $this->language->get('entry_total');
        $data['entry_resultUrl'] = $im_lang->getTitle('resultUrl');
        $data['heading_title'] = $this->language->get('heading_title');
        $data['text_edit'] = $this->language->get('text_edit');
        $data['button_save'] = $this->language->get('button_save');
        $data['button_cancel'] = $this->language->get('button_cancel');
        $data['tab_general'] = $this->language->get('tab_general');
        $data['entry_geo_zone'] = $this->language->get('entry_geo_zone');
        $data['entry_status'] = $this->language->get('entry_status');
        $data['entry_sort_order'] = $this->language->get('entry_sort_order');
        $data['text_enabled'] = $this->language->get('text_enabled');
        $data['text_disabled'] = $this->language->get('text_disabled');
        $data['text_all_zones'] = $this->language->get('text_all_zones');

        //map Errors
        $data['error_warning'] = isset($this->error['warning']) ? $this->error['warning'] : "";
        foreach ($this->error as $key => $value) {
            $data['error_' . $key] = $value;
        }

        //map Values
        foreach ($im_userSettings->getNamesOfOrganizationParamsToSave() as $value) {
            if (isset($this->request->post['payment_intellectmoney_' . $value])) {
                $funcName = 'get' . ucfirst($value);
            }
        }

```



```

        $data['payment_intellectmoney_' . $value] = $this->request->post['payment_intellectmoney_' .
$value];
    } else {
        $data['payment_intellectmoney_' . $value] = $this->config->get('payment_intellectmoney_' .
$value);
    }
}
$this->load->model('localisation/geo_zone');
$data['geo_zones'] = $this->model_localisation_geo_zone->getGeoZones();
$this->load->model('localisation/order_status');
$data['order_statuses'] = $this->model_localisation_order_status->getOrderStatuses();
$data['payment_intellectmoney_total'] = isset($this->request->post['payment_intellectmoney_total'])
? $this->request->post['payment_intellectmoney_total'] : $this->config->get('payment_intellectmoney_total');
$data['payment_intellectmoney_geo_zone_id'] = isset($this->request->post
['payment_intellectmoney_geo_zone_id']) ? $this->request->post['payment_intellectmoney_geo_zone_id'] : $this-
>config->get('payment_intellectmoney_geo_zone_id');
$data['payment_intellectmoney_status'] = isset($this->request->post
['payment_intellectmoney_status']) ? $this->request->post['payment_intellectmoney_status'] : $this->config-
>get('payment_intellectmoney_status');
$data['payment_intellectmoney_sort_order'] = isset($this->request->post
['payment_intellectmoney_sort_order']) ? $this->request->post['payment_intellectmoney_sort_order'] : $this-
>config->get('payment_intellectmoney_sort_order');
$data['payment_intellectmoney_total'] = isset($this->request->post['payment_intellectmoney_total'])
? $this->request->post['payment_intellectmoney_total'] : $this->config->get('payment_intellectmoney_total');
$data['payment_intellectmoney_geo_zone_id'] = isset($this->request->post
['payment_intellectmoney_geo_zone_id']) ? $this->request->post['payment_intellectmoney_geo_zone_id'] : $this-
>config->get('payment_intellectmoney_geo_zone_id');
$data['payment_intellectmoney_status'] = isset($this->request->post
['payment_intellectmoney_status']) ? $this->request->post['payment_intellectmoney_status'] : $this->config-
>get('payment_intellectmoney_status');
$data['payment_intellectmoney_sort_order'] = isset($this->request->post
['payment_intellectmoney_sort_order']) ? $this->request->post['payment_intellectmoney_sort_order'] : $this-
>config->get('payment_intellectmoney_sort_order');
$data['payment_intellectmoney_resultUrl'] = HTTP_CATALOG . 'index.php?route=extension/payment
/intellectmoney/callback';

$this->load->model('localisation/order_status');
$data['order_statuses'] = $this->model_localisation_order_status->getOrderStatuses();
$data['VATs'] = $this->model_IntellectMoney_intellectmoney_model->getVATsList();
$data['statusesIm'] = array(
    array(
        'name' => $data['entry_statusCreated'],
        'dataName' => 'payment_intellectmoney_statusCreated',
        'value' => $data['payment_intellectmoney_statusCreated']
    ),
    array(
        'name' => $data['entry_statusCancelled'],
        'dataName' => 'payment_intellectmoney_statusCancelled',
        'value' => $data['payment_intellectmoney_statusCancelled']
    ),
    array(
        'name' => $data['entry_statusPaid'],
        'dataName' => 'payment_intellectmoney_statusPaid',
        'value' => $data['payment_intellectmoney_statusPaid']
    ),
    array(
        'name' => $data['entry_statusHoled'],
        'dataName' => 'payment_intellectmoney_statusHoled',
        'value' => $data['payment_intellectmoney_statusHoled']
    ),
    array(
        'name' => $data['entry_statusPartiallyPaid'],
        'dataName' => 'payment_intellectmoney_statusPartiallyPaid',
        'value' => $data['payment_intellectmoney_statusPartiallyPaid']
    ),
    array(
        'name' => $data['entry_statusRefunded'],
        'dataName' => 'payment_intellectmoney_statusRefunded',
        'value' => $data['payment_intellectmoney_statusRefunded']
    ),
);
);

```

```

        $data['header'] = $this->load->controller('common/header');
        $data['column_left'] = $this->load->controller('common/column_left');
        $data['footer'] = $this->load->controller('common/footer');

        $this->response->setOutput($this->load->view('extension/payment/intellectmoney', $data));
    }

    private function validate($im_lang, $im_userSettings) {
        if (!$this->user->hasPermission('modify', 'extension/payment/intellectmoney')) {
            $this->error['warning'] = $this->language->get('error_permission');
        }

        foreach ($im_userSettings->getNamesOfOrganizationParamsToSave() as $paramName) {
            $methodName = 'set' . ucfirst($paramName);
            $paramValue = isset($this->request->post['payment_intellectmoney_' . $paramName]) ? $this->request->post['payment_intellectmoney_' . $paramName] : "";
            if (in_array($paramName, array('holdMode', 'testMode')) && $paramValue == 'on') {
                $paramValue = 1;
            }
            if (!$im_userSettings->$methodName($paramValue)) {
                $this->error[$paramName] = $im_lang->getDesc($paramName);
            }
        }
        return empty($this->error);
    }
}

?>

```

Model PHP

```

<?php

require_once("IntellectMoneyCommon/LanguageHelper.php");
require_once("IntellectMoneyCommon/UserSettings.php");
require_once("IntellectMoneyCommon/VATs.php");

class ModelIntellectMoneyIntellectmoneyModel extends Model {

    public function getLanguageHelperInstance($lang = 'ru', $isCpl251 = NULL) {
        return \PaySystem\LanguageHelper::getInstance($lang, $isCpl251);
    }

    public function getUserSettingsInstance($params = array(), $isAllowChangeValues = true) {
        return \PaySystem\UserSettings::getInstance($params, $isAllowChangeValues);
    }

    public function getVATsList() {
        return \PaySystem\VATs::getList();
    }

}

```

CMS OpenCart (MVC)

View twig

```
{{ form }}
<div class="buttons">
  <div class="pull-right">
    <input type="submit" value="{{ button_confirm }}" class="btn btn-primary" onclick="getElementById
('IntellectMoneyPaymentForm').submit();" />
  </div>
</div>
```

Controller PHP

<?php

```
class ControllerExtensionPaymentIntellectmoney extends Controller {

    public function index() {
        $im_userSettings = $this->loadSettings();
        $orderInfo = $this->model_checkout_order->getOrder($this->session->data['order_id']);
        $im_customer = $this->model_extension_payment_intellectmoney->getCustomerInstance($orderInfo
['email'], $orderInfo['shipping_firstname'] . " " . $orderInfo['shipping_lastname'], $orderInfo
['telephone']);
        $im_order = $this->model_extension_payment_intellectmoney->getOrderInstance(NULL, $orderInfo
['order_id'], $orderInfo['total'], $orderInfo['total'], NULL, NULL, $orderInfo['currency_code'], NULL, NULL);
        //products
        foreach ($this->cart->getProducts() as $product) {
            $im_order->addItem($product['price'], $product['quantity'], $product['name'], $this-
>loadSettings()->getTax());
        }
        //delivery
        if (isset($this->session->data['shipping_methods'])) {
            $im_order->addItem($this->session->data['shipping_methods']['flat']['quote']['flat']['cost'], 1,
$this->session->data['shipping_methods']['flat']['quote']['flat']['title'], $this->loadSettings()-
>getDeliveryTax());
        }

        $im_payment = $this->model_extension_payment_intellectmoney->getPaymentInstance($im_userSettings,
$im_order, $im_customer, $this->language->data['code']);
        $data['form'] = $im_payment->generateForm(true);
        $data['button_confirm'] = $this->language->get('button_confirm');

        return $this->load->view('extension/payment/intellectmoney', $data);
    }

    private function loadSettings() {
        $this->load->language('extension/payment/intellectmoney');
        $this->load->model('extension/payment/intellectmoney');
        $this->load->model('checkout/order');

        $im_userSettings = $this->model_extension_payment_intellectmoney->getUserSettingsInstance();
        $params = array();
        foreach ($im_userSettings->getNamesOfOrganizationParamsToSave() as $value) {
            if (in_array($value, array('holdMode', 'testMode'))) && $this->config->get
('payment_intellectmoney_' . $value) == 'on') {
                $params[$value] = 1;
            } else {
                $params[$value] = $this->config->get('payment_intellectmoney_' . $value);
            }
        }
        $im_userSettings->setParams($params);
        $im_userSettings->setMerchantUrl('https://merchant.intellectmoney.ru');
        return $im_userSettings;
    }
}

?>
```

Model PHP

<?php

```
require_once("admin/model/IntellectMoney/IntellectMoneyCommon/LanguageHelper.php");
require_once("admin/model/IntellectMoney/IntellectMoneyCommon/UserSettings.php");
require_once("admin/model/IntellectMoney/IntellectMoneyCommon/Order.php");
require_once("admin/model/IntellectMoney/IntellectMoneyCommon/Customer.php");
require_once("admin/model/IntellectMoney/IntellectMoneyCommon/Payment.php");
```

```

require_once("admin/model/IntellectMoney/IntellectMoneyCommon/Result.php");

class ModelExtensionPaymentIntellectMoney extends Model {

    public function getMethod($address, $total) {
        $this->load->language('extension/payment/intellectmoney');

        $query = $this->db->query("SELECT * FROM " . DB_PREFIX . "zone_to_geo_zone WHERE geo_zone_id = '" .
(int) $this->config->get('payment_intellectmoney_geo_zone_id') . "' AND country_id = '" . (int) $address
['country_id'] . "' AND (zone_id = '" . (int) $address['zone_id'] . "' OR zone_id = '0')");

        if ($this->config->get('payment_intellectmoney_total') > 0 && $this->config->get
('payment_intellectmoney_total') > $total) {
            $status = false;
        } elseif (!$this->config->get('payment_intellectmoney_geo_zone_id')) {
            $status = true;
        } elseif ($query->num_rows) {
            $status = true;
        } else {
            $status = false;
        }

        $method_data = array();
        if ($status) {
            $method_data = array(
                'code' => 'intellectmoney',
                'title' => $this->language->get('text_title'),
                'terms' => '',
                'sort_order' => $this->config->get('payment_intellectmoney_sort_order')
            );
        }

        return $method_data;
    }

    public function getLanguageHelperInstance($lang = 'ru', $isCpl251 = NULL) {
        return \PaySystem\LanguageHelper::getInstance($lang, $isCpl251);
    }

    public function getUserSettingsInstance($params = array(), $isAllowChangeValues = true) {
        return \PaySystem\UserSettings::getInstance($params, $isAllowChangeValues);
    }

    public function getOrderInstance($invoiceId = NULL, $orderId = NULL, $originalAmount = NULL,
$recipientAmount = NULL, $paidAmount = NULL, $deliveryAmount = NULL, $recipientCurrency = NULL, $discount =
NULL, $status = NULL) {
        return \PaySystem\Order::getInstance($invoiceId, $orderId, $originalAmount, $recipientAmount,
$paidAmount, $deliveryAmount, $recipientCurrency, $discount, $status);
    }

    public function getCustomerInstance($email = NULL, $name = NULL, $phone = NULL) {
        return \PaySystem\Customer::getInstance($email, $name, $phone);
    }

    public function getPaymentInstance($UserSettings = NULL, $Order = NULL, $Customer = NULL, $lang = 'ru') {
        return \PaySystem\Payment::getInstance($UserSettings, $Order, $Customer, $lang);
    }

    public function getResultInstance($request = array(), $UserSettings = "", $Order = "", $lang = 'ru',
$isCpl251 = false) {
        return \PaySystem\Result::getInstance($request, $UserSettings, $Order, $lang, $isCpl251);
    }

    public function changeOrderStatusCMS($orderId, $statusCMS) {
        $this->load->model('checkout/order');
        $this->model_checkout_order->addOrderHistory($orderId, $statusCMS, 'IntellectMoney', False);
        return true;
    }

    public function getImStatusFromOrder($orderId) {
        return $this->getParamFromOrder($orderId, 'imStatus');
    }

```

```

    }

    public function getInvoiceIdFromOrder($orderId) {
        return $this->getParamFromOrder($orderId, 'invoiceId');
    }

    public function getParamFromOrder($orderId, $paramName) {
        $params = $this->getCustomDataFromOrder($orderId);
        return isset($params['custom_field'][$paramName]) ? $params['custom_field'][$paramName] : NULL;
    }

    public function saveImStatusToOrder($orderId, $imStatus) {
        return $this->saveParamToOrder($orderId, 'imStatus', $imStatus);
    }

    public function saveInvoiceIdToOrder($orderId, $invoiceId) {
        return $this->saveParamToOrder($orderId, 'invoiceId', $invoiceId);
    }

    public function saveParamToOrder($orderId, $paramName, $value) {
        $params = $this->getCustomDataFromOrder($orderId);
        $params['custom_field'][$paramName] = $value;
        return $this->saveCustomDataToOrder($orderId, $params);
    }

    private function saveCustomDataToOrder($orderId, $params) {
        $this->load->model('checkout/order');
        $this->model_checkout_order->editOrder($orderId, $params);
        return true;
    }

    private function getCustomDataFromOrder($orderId) {
        $this->load->model('checkout/order');
        $orderData = $this->model_checkout_order->getOrder($orderId);
        return isset($orderData['custom_field']) ? $orderData : false;
    }
}

```

CMS OpenCart (MVC)

Controller PHP

```
<?php

class ControllerExtensionPaymentIntellectmoney extends Controller {

    public function callback() {
        $im_userSettings = $this->loadSettings();

        $orderId = $this->model_checkout_order->getOrder($this->request->get['orderId']);
        $im_order = $this->model_extension_payment_intellectmoney->getOrderInstance($this->model_extension_payment_intellectmoney->getInvoiceIdFromOrder($orderId), $orderId, $orderId['total'], $orderId['total'], NULL, NULL, $orderId['currency_code'], NULL, $this->model_extension_payment_intellectmoney->getImStatusFromOrder($orderId));
        $im_result = $this->model_extension_payment_intellectmoney->getResultInstance($this->request->get, $im_userSettings, $im_order, $this->language->data['code']);

        $response = $im_result->processingResponse();
        if ($response->changeStatusResult) {
            if (!$this->model_extension_payment_intellectmoney->getInvoiceIdFromOrder($this->request->get['orderId'])) {
                $this->model_extension_payment_intellectmoney->saveInvoiceIdToOrder($this->request->get['orderId'], $this->request->get['paymentId']);
            }
            $this->model_extension_payment_intellectmoney->saveImStatusToOrder($this->request->get['orderId'], $this->request->get['paymentStatus']);
            $this->model_extension_payment_intellectmoney->changeOrderStatusCMS($this->request->get['orderId'], $response->statusCMS);
        }

        echo $im_result->getMessage();
        die;
    }

    private function loadSettings() {
        $this->load->language('extension/payment/intellectmoney');
        $this->load->model('extension/payment/intellectmoney');
        $this->load->model('checkout/order');

        $im_userSettings = $this->model_extension_payment_intellectmoney->getUserSettingsInstance();
        $params = array();
        foreach ($im_userSettings->getNamesOfOrganizationParamsToSave() as $value) {
            if (in_array($value, array('holdMode', 'testMode'))) && $this->config->get('payment_intellectmoney_' . $value) == 'on') {
                $params[$value] = 1;
            } else {
                $params[$value] = $this->config->get('payment_intellectmoney_' . $value);
            }
        }
        $im_userSettings->setParams($params);
        $im_userSettings->setMerchantUrl('https://merchant.intellectmoney.ru');
        return $im_userSettings;
    }
}

?>
```

Model PHP

```
<?php

require_once("admin/model/IntellectMoney/IntellectMoneyCommon/LanguageHelper.php");
require_once("admin/model/IntellectMoney/IntellectMoneyCommon/UserSettings.php");
require_once("admin/model/IntellectMoney/IntellectMoneyCommon/Order.php");
require_once("admin/model/IntellectMoney/IntellectMoneyCommon/Customer.php");
require_once("admin/model/IntellectMoney/IntellectMoneyCommon/Payment.php");
```

```

require_once("admin/model/IntellectMoney/IntellectMoneyCommon/Result.php");

class ModelExtensionPaymentIntellectMoney extends Model {

    public function getMethod($address, $total) {
        $this->load->language('extension/payment/intellectmoney');

        $query = $this->db->query("SELECT * FROM " . DB_PREFIX . "zone_to_geo_zone WHERE geo_zone_id = '" .
(int) $this->config->get('payment_intellectmoney_geo_zone_id') . "' AND country_id = '" . (int) $address
['country_id'] . "' AND (zone_id = '" . (int) $address['zone_id'] . "' OR zone_id = '0')");

        if ($this->config->get('payment_intellectmoney_total') > 0 && $this->config->get
('payment_intellectmoney_total') > $total) {
            $status = false;
        } elseif (!$this->config->get('payment_intellectmoney_geo_zone_id')) {
            $status = true;
        } elseif ($query->num_rows) {
            $status = true;
        } else {
            $status = false;
        }

        $method_data = array();
        if ($status) {
            $method_data = array(
                'code' => 'intellectmoney',
                'title' => $this->language->get('text_title'),
                'terms' => '',
                'sort_order' => $this->config->get('payment_intellectmoney_sort_order')
            );
        }

        return $method_data;
    }

    public function getLanguageHelperInstance($lang = 'ru', $isCpl251 = NULL) {
        return \PaySystem\LanguageHelper::getInstance($lang, $isCpl251);
    }

    public function getUserSettingsInstance($params = array(), $isAllowChangeValues = true) {
        return \PaySystem\UserSettings::getInstance($params, $isAllowChangeValues);
    }

    public function getOrderInstance($invoiceId = NULL, $orderId = NULL, $originalAmount = NULL,
$recipientAmount = NULL, $paidAmount = NULL, $deliveryAmount = NULL, $recipientCurrency = NULL, $discount =
NULL, $status = NULL) {
        return \PaySystem\Order::getInstance($invoiceId, $orderId, $originalAmount, $recipientAmount,
$paidAmount, $deliveryAmount, $recipientCurrency, $discount, $status);
    }

    public function getCustomerInstance($email = NULL, $name = NULL, $phone = NULL) {
        return \PaySystem\Customer::getInstance($email, $name, $phone);
    }

    public function getPaymentInstance($UserSettings = NULL, $Order = NULL, $Customer = NULL, $lang = 'ru') {
        return \PaySystem\Payment::getInstance($UserSettings, $Order, $Customer, $lang);
    }

    public function getResultInstance($request = array(), $UserSettings = "", $Order = "", $lang = 'ru',
$isCpl251 = false) {
        return \PaySystem\Result::getInstance($request, $UserSettings, $Order, $lang, $isCpl251);
    }

    public function changeOrderStatusCMS($orderId, $statusCMS) {
        $this->load->model('checkout/order');
        $this->model_checkout_order->addOrderHistory($orderId, $statusCMS, 'IntellectMoney', False);
        return true;
    }

    public function getImStatusFromOrder($orderId) {
        return $this->getParamFromOrder($orderId, 'imStatus');
    }

```



```

    }

    public function getInvoiceIdFromOrder($orderId) {
        return $this->getParamFromOrder($orderId, 'invoiceId');
    }

    public function getParamFromOrder($orderId, $paramName) {
        $params = $this->getCustomDataFromOrder($orderId);
        return isset($params['custom_field'][$paramName]) ? $params['custom_field'][$paramName] : NULL;
    }

    public function saveImStatusToOrder($orderId, $imStatus) {
        return $this->saveParamToOrder($orderId, 'imStatus', $imStatus);
    }

    public function saveInvoiceIdToOrder($orderId, $invoiceId) {
        return $this->saveParamToOrder($orderId, 'invoiceId', $invoiceId);
    }

    public function saveParamToOrder($orderId, $paramName, $value) {
        $params = $this->getCustomDataFromOrder($orderId);
        $params['custom_field'][$paramName] = $value;
        return $this->saveCustomDataToOrder($orderId, $params);
    }

    private function saveCustomDataToOrder($orderId, $params) {
        $this->load->model('checkout/order');
        $this->model_checkout_order->editOrder($orderId, $params);
        return true;
    }

    private function getCustomDataFromOrder($orderId) {
        $this->load->model('checkout/order');
        $orderData = $this->model_checkout_order->getOrder($orderId);
        return isset($orderData['custom_field']) ? $orderData : false;
    }
}

```