**Структура базы.**

**Структура базы** предполагает наличие 2 таблиц ведущая (основная) и ведомая (языковая):

Пример:

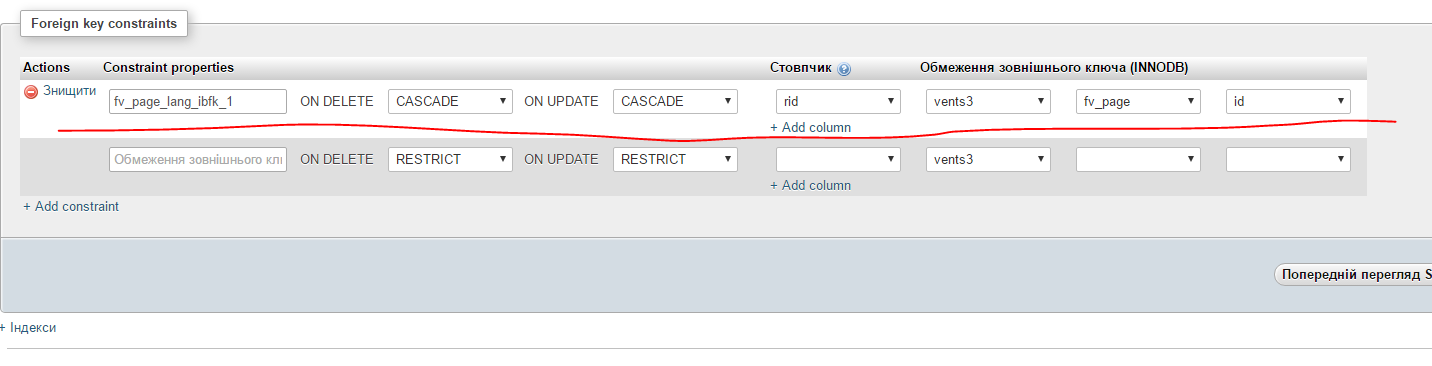
|  |  |
| --- | --- |
| **основная** | fv\_page |
| **языковая** | fv\_page\_lang |



В основной таблице: id

В языковой 2 поля: rid, lang

Связь во второй таблице



**Модуль**

Делится на 3 или 4 части.

1. Если модуль новый добавляется в проект он создается относительно директории **/common – 3 части**
   1. В случае, если модуль пишеться в ядро системы, то относительно директории **/thread – 4 части**
2. Структура модуля:

assets  
interfaces  
components  
extentions  
widgets  
 name\_of\_widget  
models  
 behaviors  
 events  
 search  
 query  
controllers  
 events  
views  
tests

1. Миграции ложим в определенный модуль не в единый скоп.
   1. Все миграции по проекту запускаются через единое окно **php yii migrate**
   2. Путь к миграциям подключаем в конфиге **/console/config/main.php**

'controllerMap' => [  
 'migrate' => [  
 'class' => \thread\app\console\controllers\MigrateController::*class*,  
 'migrationPaths' => [  
 '@thread/modules/page/migrations',  
 ],  
 ],  
],

**Подключение модуля к БД**

формируем в классе модуля и соответственно подключаем в моделях модуля

**class** Page **extends** aModule  
{  
  
 **public** $name = 'Page';  
 **public** $translationsBasePath = *\_\_DIR\_\_* . '/messages';  
 **public** $configPath = *\_\_DIR\_\_* . '/config.php';  
  
  
 */\*\*  
 \* Db connection  
 \*  
 \** ***@return*** *null|object  
 \** ***@throws*** *\yii\base\InvalidConfigException  
 \*/* **public static function** getDb()  
 {  
 **return** Yii::*$app*->get('db-core');  
 }  
  
}

**class** Page **extends** ActiveRecord  
{  
 */\*\*  
 \* Using db connection  
 \** ***@return*** *string  
 \*/* **public static function** getDb()  
 {  
 **return** PageModule::*getDb*();  
 }

}

**Модели данных**

**Для основной таблицы**

**Дополнительные параметры для запросов создаем на основе use** thread\app\base\models\query\ActiveQuery;

Свой клас можно переопределить в классе

*/\*\*  
 \** ***@var*** *\*/***public static** *$commonQuery* = query\ActiveQuery::*class*;

Базвый метод find() переопределен

*/\*\*  
 \** ***@return*** *mixed  
 \*/***public static function** find()  
{  
 **return new static**::*$commonQuery*(get\_called\_class());  
}

**use** thread\app\base\models\ActiveRecord;  
**use** thread\app\base\models\query\ActiveQuery;  
**use** thread\modules\page\Page **as** PageModule;

**class** Page **extends** ActiveRecord  
{  
 */\*\*  
 \* Using db connection  
 \** ***@return*** *string  
 \*/* **public static function** getDb()  
 {  
 **return** PageModule::*getDb*();  
 }  
  
 */\*\*  
 \* Page table name  
 \** ***@return*** *string  
 \*/* **public static function** tableName()  
 {  
 **return** '{{%page}}';  
 }

*/\*\*  
 \** ***@inheritdoc*** *\** ***@return*** *array  
 \*/***public function** scenarios()  
{  
 **return** [  
 'published' => ['published'],  
 'deleted' => ['deleted'],  
 'backend' => ['alias', 'image\_link', 'group\_id', 'parent\_id', 'published', 'deleted', 'gallery\_image'],  
 ];  
}

*/\*\*  
 \*  
 \** ***@return*** *ActiveQuery  
 \*/* **public function** getLang()  
 {  
 **return** $this->hasOne(PageLang::*class*, ['rid' => 'id']);  
 }  
}

**Для языковой**

Базвый метод find() переопределен

*/\*\*  
 \** ***@return*** *$this  
 \*/***public static function** find()  
{  
 **return parent**::*find*()->onCondition([**static**::*tableName*() . '.lang' => Yii::*$app*->language]);  
}

**<?php  
  
namespace** thread\modules\page\models;  
  
**use** thread\app\base\models\ActiveRecordLang;  
**use** thread\app\base\models\query\ActiveQuery;  
**use** thread\app\model\Language;  
**use** thread\modules\page\Page **as** PageModel;  
  
*/\*\*  
 \* Class PageLang  
 \*  
 \** ***@property*** *integer $rid  
 \** ***@property*** *string $lang  
 \** ***@property*** *string $title  
 \** ***@property*** *string $content  
 \*  
 \** ***@property*** *Page[] $pages  
 \** ***@property*** *PageLang[] $langs  
 \*  
 \*/***class** PageLang **extends** ActiveRecordLang  
{  
 */\*\*  
 \* Db connection  
 \** ***@return*** *string  
 \*/* **public static function** getDb()  
 {  
 **return** PageModel::*getDb*();  
 }  
  
 */\*\*  
 \** ***@inheritdoc*** *\** ***@return*** *string  
 \*/* **public static function** tableName()  
 {  
 **return** '{{%page\_lang}}';  
 }  
  
 */\*\*  
 \** ***@inheritdoc*** *\** ***@return*** *array  
 \*/* **public function** scenarios()  
 {  
 **return** [  
 'backend' => ['title', 'content', 'alias'],  
 ];  
 }  
}

Базовый сценарий при сохранении в админке 'backend'

**Административная часть.**

**Контроллер**

**<?php  
namespace** backend\modules\page\controllers;  
  
**use** thread\actions\fileapi\DeleteAction;  
**use** Yii;  
//  
**use** thread\actions\fileapi\UploadAction;  
**use** thread\app\base\controllers\BackendController;  
//  
**use** backend\modules\page\models\{  
 Page, PageLang  
};  
  
**use** yii\helpers\ArrayHelper;  
**class** PageController **extends** BackendController  
{  
 **public** $model = Page::*class*;  
 **public** $modelLang = PageLang::*class*;  
 **public** $title = 'Pages';

}

**Модели**

**Основная**

**<?php  
namespace** backend\modules\page\models;  
  
**use** backend\components\HtmlTemplate;  
**use** Yii;  
**use** yii\helpers\ArrayHelper;  
//  
**use** common\modules\page\models\Page **as** CommonPageModel;  
//  
**use** thread\actions\fileapi\UploadBehavior;  
**use** thread\modules\seo\behaviors\SeoBehavior;  
**use** thread\app\model\interfaces\BaseBackendModel;  
**use** yii\helpers\Url;  
  
**class** Page **extends** CommonPageModel **implements** BaseBackendModel  
{  
  
 */\*\*  
 \* Behaviors  
 \*  
 \** ***@return*** *array  
 \*/* **public function** behaviors()  
 {  
  
 **return** ArrayHelper::*merge*(  
 **parent**::*behaviors*(),  
 [  
 'SeoBehavior' => [  
 'class' => SeoBehavior::*class*,  
 ],  
 'uploadBehavior' => [  
 'class' => UploadBehavior::*class*,  
 'attributes' => [  
 'gallery\_image' => [  
 'path' => Yii::*$app*->getModule('page')->getPageGalleryUploadPath(),  
 'tempPath' => Yii::*getAlias*('@root') . '/temp',  
 'url' => Yii::*$app*->getModule('page')->getPageGalleryUploadPath(**false**),  
 ],  
 ]  
 ]  
 ]  
 );  
 }  
  
 */\*\*  
 \** ***@return*** *string  
 \*/* **public function** getTitle()  
 {  
 **return** ($this->lang->title)?? "{'ru-RU' " . $this->default\_title . "}";  
 }  
  
 */\*\*  
 \** ***@param*** *$params  
 \** ***@return*** *\yii\data\ActiveDataProvider  
 \*/* **public function** search($params)  
 {  
 **return** (**new** search\Page())->search($params);  
 }  
  
 */\*\*  
 \** ***@param*** *$params  
 \** ***@return*** *\yii\data\ActiveDataProvider  
 \*/* **public function** trash($params)  
 {  
 **return** (**new** search\Page())->trash($params);  
 }  
  
}

**Языковая**

**<?php  
namespace** backend\modules\page\models;  
  
**use** common\modules\page\models\PageLang **as** CommonPageLangModel;  
  
*/\*\*  
 \** ***@author*** *Roman Gonchar <****roman.gonchar92@gmail.com****>  
 \** ***@copyright*** *(c) 2016, VipDesign  
 \*/***class** PageLang **extends** CommonPageLangModel  
{  
   
}

**Для** ActiveDataProvider

**<?php  
namespace** backend\modules\page\models\search;  
  
**use** Yii;  
**use** yii\data\ActiveDataProvider;  
**use** yii\base\Model;  
//  
**use** thread\app\base\models\query\ActiveQuery;  
**use** thread\app\model\interfaces\search\BaseBackendSearchModel;  
//  
**use** backend\modules\page\Page **as** PageModule;  
**use** backend\modules\page\models\{  
 PageLang, Page **as** PageModel  
};  
  
**class** Page **extends** PageModel **implements** BaseBackendSearchModel  
{  
 **public** $title;  
  
 */\*\*  
 \** ***@return*** *array  
 \*/* **public function** rules()  
 {  
 **return** [  
 [['alias', 'title'], 'string', 'max' => 255],  
 [['published'], 'in', 'range' => array\_keys(**self**::*statusKeyRange*())],  
 ];  
 }  
  
 */\*\*  
 \*  
 \** ***@return*** *array  
 \*/* **public function** scenarios()  
 {  
 **return** Model::*scenarios*();  
 }  
  
 */\*\*  
 \** ***@param*** *ActiveQuery $query  
 \** ***@param*** *array $params  
 \** ***@return*** *ActiveDataProvider  
 \*/* **public function** baseSearch($query, $params)  
 {  
 */\*\** ***@var*** *PageModule $module \*/* $module = Yii::*$app*->getModule('page');  
 $dataProvider = **new** ActiveDataProvider([  
 'query' => $query,  
 'pagination' => [  
 'pageSize' => $module->itemOnPage  
 ],  
 'sort' => [  
 'defaultOrder' => [  
 'id' => *SORT\_ASC* ]  
 ]  
 ]);  
  
 **if** (!($this->load($params) && $this->validate())) {  
 **return** $dataProvider;  
 }  
  
 $query->andFilterWhere(['like', 'alias', $this->alias])  
 ->andFilterWhere(['like', 'published', $this->published]);  
 //  
 $query->andFilterWhere(['like', PageLang::*tableName*() . '.title', $this->title]);  
  
 **return** $dataProvider;  
 }  
  
 */\*\*  
 \** ***@param*** *array $params  
 \** ***@return*** *ActiveDataProvider  
 \*/* **public function** search($params)  
 {  
 $query = PageModel::*find*()->innerJoinWith(['lang'])->undeleted();  
 **return** $this->baseSearch($query, $params);  
 }  
  
 */\*\*  
 \** ***@param*** *array $params  
 \** ***@return*** *ActiveDataProvider  
 \*/* **public function** trash($params)  
 {  
 $query = PageModel::*find*()->innerJoinWith(['lang'])->deleted();  
 **return** $this->baseSearch($query, $params);  
 }  
}

**Во Views /views/**

**Список ./page/list.php**

**<?php**//  
**use** backend\themes\inspinia\widgets\GridView;  
  
$filter = **new** \backend\modules\page\models\search\Page();  
$filter->setAttributes(Yii::*$app*->getRequest()->get('Page'));  
  
*/\*\*  
 \** ***@var*** *\backend\modules\page\models\search\Page $model  
 \*/***echo** GridView::*widget*([  
 'dataProvider' => $model->search(Yii::*$app*->request->queryParams),  
 'tableOptions' => ['class' => 'table table-striped table-bordered'],  
 'filterModel' => $filter,  
 'columns' => [  
 [  
 'attribute' => 'title',  
 'label' => Yii::*t*('app', 'Title'),  
 ],  
 [  
 'class' => \thread\widgets\grid\ActionCheckboxColumn::*class*,  
 'attribute' => 'published',  
 'action' => 'published'  
 ],  
 [  
 'class' => \backend\themes\inspinia\widgets\gridColumns\ActionColumn::*class* ],  
 ]  
]);

**Корзина ./page/trash.php**

**<?php  
  
use** thread\widgets\grid\{  
 ActionDeleteColumn, ActionRestoreColumn  
};  
**use** backend\themes\inspinia\widgets\GridView;  
  
*/\*\*  
 \*  
 \** ***@package*** *admin\modules\page\view  
 \** ***@author*** *FilamentV <****vortex.filament@gmail.com****>  
 \** ***@copyright*** *(c) 2015, Thread  
 \*/***echo** GridView::*widget*([  
 'dataProvider' => $model->trash(Yii::*$app*->request->queryParams),  
 'tableOptions' => ['class' => 'table table-striped table-bordered'],  
 'columns' => [  
 'lang.title',  
 [  
 'class' => ActionDeleteColumn::*class*,  
 ],  
 [  
 'class' => ActionRestoreColumn::*class* ],  
 ]  
]);

**Форма создание/редактирования /page/\_form.php**

**<?php  
use** thread\modules\seo\widgets\seo\SeoWidget;  
//  
**use** backend\modules\page\models\{  
 Page, PageLang  
};  
  
**use** backend\themes\inspinia\widgets\{  
 forms\ActiveForm, Tabs, forms\Form  
};  
  
**<?=** Tabs::*widget*([  
 'items' => [  
 [  
 'label' => Yii::*t*('app', 'General'),  
 'content' => $this->render('parts/\_main', [  
 'form' => $form,  
 'model' => $model,  
 'modelLang' => $modelLang,  
 ])  
 ],  
 [  
 'label' => Yii::*t*('app', 'SEO'),  
 'content' => SeoWidget::*widget*(['model' => $model])  
 ],  
 ],  
]); **?>  
  
<?=** Form::*submit*($model, $this) **?>  
  
<?php** ActiveForm::*end*(); **?>**

**/page/part/\_main**

**<?php***/\*\*  
 \** ***@var*** *backend\modules\page\models\Page $model  
 \** ***@var*** *backend\modules\page\models\PageLang | \thread\app\base\models\ActiveRecord $modelLang  
 \** ***@var*** *\backend\themes\inspinia\widgets\forms\ActiveForm $form  
 \*/***?>  
<?=** $form->field($model, 'parent\_id')->dropDownList(['' =>'- root -' ] + $model::*listParent*()); **?>  
<?=** $form->field($modelLang, 'title')->textInput(['maxlength' => **true**]) **?>  
<?=** $form->field($modelLang, 'alias')->textInput(['maxlength' => **true**]) **?>  
<?=** $form->field($modelLang, 'content')->editor() **?>  
<?=** $form->field($model, 'published')->checkbox() **?>**

**Добавление новых языков для админки**

**/backend/config/params.php**

**<?php  
return** [  
//themes base settings  
 'themes' => [  
 'language' => 'en-EN',  
 'languages' => [  
 'en-EN' => 'English',  
 'ru-RU' => 'Русский',  
 'uk-UA' => 'Українська',  
 ]  
 ],  
];

**Пользовательская часть**

**Контроллер**

**<?php  
  
namespace** frontend\modules\page\controllers;  
  
  
**use** frontend\components\BaseController;  
**use** frontend\modules\seo\models\Seo;  
**use** Yii;  
**use** yii\web\NotFoundHttpException;  
**use** frontend\modules\page\models\Page;  
  
**class** PageController **extends** BaseController  
{  
 **public** $title = "Page";  
 **public** $layout = "@app/layouts/base";  
 **public** $defaultAction = 'index';  
 **public** $model;  
  
  
 */\*\*  
 \** ***@return*** *array  
 \*/* **public function** behaviors()  
 {  
 **return** [  
 'verbs' => [  
 'class' => \yii\filters\VerbFilter::*class*,  
 'actions' => [  
 'index' => ['get'],  
 'contacts' => ['get'],  
 'view' => ['get'],  
 'solutions' => ['get']  
  
 ],  
 ]  
 ];  
 }  
  
 */\*\*  
 \*  
 \** ***@return*** *string  
 \*/* **public function** actionIndex()  
 {  
 **return** $this->run('view', ['alias' => 'start']);  
 }  
  
 */\*\*  
 \*  
 \** ***@param*** *string $alias  
 \** ***@return*** *string  
 \** ***@throws*** *NotFoundHttpException  
 \*/* **public function** actionView($alias)  
 {  
 $model = Page::*findByAlias*($alias);  
  
 **if** ($model === **null**) {  
 **return** $this->redirect(['/home/home/error'], 301);  
 **die**();  
 }  
  
 **return** $this->render('with-menu', [  
 'model' => $model,  
 ]);  
 }  
}

**Обязательно!!**

yii\filters\VerbFilter::*class – насройка фильтрации входящего трацика*

$model = Page::*findByAlias*($alias); -- взе запросы только в модели данных, они не должны быть разбросаны по контроллерам виджетам и т.д.

**Модели**

**Основная**

**Обязательно!!**

**Все методы не используемые должны быть сброшены**

**<?php  
  
namespace** frontend\modules\page\models;  
  
**use** frontend\components\HtmlTemplate;  
**use** yii\helpers\Url;  
  
*/\*\*  
 \* Class Page  
 \*  
 \** ***@package*** *frontend\modules\page\models  
 \** ***@author*** *Andrii Bondarchuk  
 \** ***@copyright*** *(c) 2016  
 \*/***class** Page **extends** \common\modules\page\models\Page  
{  
 **public** $model = 'page';  
 */\*\*  
 \*  
 \** ***@return*** *array  
 \*/* **public function** behaviors()  
 {  
 **return** [];  
 }  
  
 */\*\*  
 \*  
 \** ***@return*** *array  
 \*/* **public function** scenarios()  
 {  
 **return** [];  
 }  
  
 */\*\*  
 \*  
 \** ***@return*** *array  
 \*/* **public function** attributeLabels()  
 {  
 **return** [];  
 }  
  
 */\*\*  
 \*  
 \** ***@return*** *array  
 \*/* **public function** rules()  
 {  
 **return** [];  
 }  
  
 */\*\*  
 \*  
 \** ***@return*** *\yii\db\ActiveQuery  
 \*/* **public static function** find\_base()  
 {  
 **return** parent::*find*()->enabled()->innerJoinWith('lang');  
 }  
  
 */\*\*  
 \*  
 \** ***@param*** *string $alias  
 \** ***@return*** *ActiveRecord|null  
 \*/* **public static function** findByAlias($alias)  
 {  
 **return self**::*find\_base*()->andWhere([PageLang::*tableName*().'.alias' => $alias])->one();  
 }  
  
 */\*\*  
 \* Url route to view particular page  
 \** ***@param*** *null $schema  
 \** ***@return*** *string  
 \*/* **public function** getUrl($schema = **null**)  
 {  
 **return** Url::*toRoute*(['/page/page/view', 'alias' => $this['lang']->alias], $schema);  
 }  
  
}

**<?php  
  
namespace** frontend\modules\page\models;  
  
*/\*\*  
 \* Class PageLang  
 \*  
 \** ***@package*** *frontend\modules\page\models  
 \** ***@author*** *Andrii Bondarchuk  
 \** ***@copyright*** *(c) 2016  
 \*/***class** PageLang **extends** \common\modules\page\models\PageLang {  
  
 */\*\*  
 \*   
 \** ***@return*** *array  
 \*/* **public function** behaviors() {  
 **return** [];  
 }  
  
 */\*\*  
 \*   
 \** ***@return*** *array  
 \*/* **public function** scenarios() {  
 **return** [];  
 }  
  
 */\*\*  
 \*   
 \** ***@return*** *array  
 \*/* **public function** attributeLabels() {  
 **return** [];  
 }  
  
 */\*\*  
 \*   
 \** ***@return*** *array  
 \*/* **public function** rules() {  
 **return** [];  
 }  
  
}

**Views**

**<?=** $model['lang']['content'] **?>**

**Обращение к свойствам обїекта ведем через иттератор массива!!!, никаких методо вили аттрибутов**