

Yii2 中使用Gii生成代码

Created by 杨超, last modified on 2016 Dec 04

- 背景介绍
- Gii的使用
 - 日常使用
 - 创建自己的gii
- 源码分析

背景介绍

首先github地址, yii2 gii扩展github地址: <https://github.com/yiisoft/yii2-gii>

yii-gii分为gui和console两个入口, 一般使用console比较方便快捷

Gii的使用

`yii help gii`

可以看到yii-gii的所有支持的功能

```
DESCRIPTION

This is the command line version of Gii - a code generator.

You can use this command to generate models, controllers, etc. For example,
to generate an ActiveRecord model based on a DB table, you can run:

$ ./yii gii/model --tableName=city --modelClass=City


SUB-COMMANDS

- gii/ajaxcrud      Ajax CRUD Generator
- gii/controller    Controller Generator
- gii/crud           CRUD Generator
- gii/extension     Extension Generator
- gii/form          Form Generator
- gii/index (default)
- gii/model         Model Generator
- gii/module        Module Generator


To see the detailed information about individual sub-commands, enter:

yii help <sub-command>
```

日常使用

生成Model

```
./yii gii/model --generateLabelsFromComments --ns="common\models" --modelClass="Depot1" --tableName="depot1" --db="db"
```

解释一下各个参数的含义

序号	参数	解释
1	--generateLabelsFromComments	使用db table里的对每个字段的comment来生成 model属性的labels
2	--ns	namespace 生成的Model的命名空间
3	--modelClass	类名，不需要包含命名空间

序号	参数	解释
4	--tableName	数据库中对应的表名
5	--db	app component里面配置的数据库id, 默认为db

根据Model生成CRUD

```
/yii gii/crud --modelClass="pay\models\PayStatInfo" --controllerClass="\mis\controllers\PaystatinfoController" --searchModelClass="
```

创建自己的gii

gii包括generator/template两部分，第一部分是主要的生成的控制代码，后面的template顾名思义是被生成的代码的模板

1 首先编写config文件，位于 console/config/main-local.php

2 然后新建myCrud目录 位于 yii2-gii文件夹下: www/vendor/yiisoft/yii2-gii/generators/mycrud

3 简单编写两个 getName & getDescription方法，yii help gii显示前者，yii help gii/myCrud展示后者

4 值得一提的是 对应的template文件我们一般重写放到 自己的app下，这里一般放到console/template底下，略去不写

```
47     /**
48      * @inheritdoc
49      */
50     public function getName()
51     {
52         return 'HELLO, MY CRUD Generator';
53     }
54
55     /**
56      * @inheritdoc
57      */
58     public function getDescription()
59     {
60         return 'My crud generator. Too young even naive, but it is mine...';
```

```
61     }  
62
```

Here comes the fresh meat

```
[yangchao@dev:www(weixin-app)]$ ./yii help gii/myCrud
```

```
DESCRIPTION
```

```
My crud generator. The same as a native, but he is mine...
```

```
[yangchao@dev:mycrud]$ ~/www/yii help gii/myCrud
```

```
DESCRIPTION
```

```
My crud generator. Too young even naive, but it is mine...
```

yii gii 生成model和crud的命令，主要探究model的生成原理，各参数的意义 以及 平时用不到的参数，从源码中获得答案

源码分析

大致过程: gii 会根据你的sub-command 进入不同的generator，以model为例，会进入 vendor/yiisoft/yii2-gii/generators/model/Generator.php

然后根据参数 生成 对应的变量值，接着根据 template里面的配置，生成对应的文件

假设输入了：

```
./yii gii/model --generateLabelsFromComments --ns="pay\models" --modelClass="Partner" --tableName="partner" --db="paydb"
```

1 首先根据console中输入的参数将这些成员变量赋值

```

32 public $db = 'db';
33 public $ns = 'app\models';
34 public $tableName;
35 public $modelClass;
36 public $baseClass = 'yii\db\ActiveRecord';
37 public $generateRelations = self::RELATIONS_ALL;
38 public $generateLabelsFromComments = false;
39 public $useTablePrefix = false;
40 public $useSchemaName = true;
41 public $generateQuery = false;
42 public $queryNs = 'app\models';
43 public $queryClass;
44 public $queryBaseClass = 'yii\db\ActiveQuery';
45

```

2 主要阅读generate函数

```

public function generate()
{
    $files = [];
    // relations 调出所有有relation关系的表, 执行partner时候只有audit的几张表和它的自增id相关联
    $relations = $this->generateRelations();
    // 读取出所有paydb中的schemas, 包括所有的表名和表中的字段信息 (包括字段名/类型/comments等) 有些big
    $db = $this->getDbConnection();

    // var_dump($relations);var_dump($db);die;

    // 获取tablename, 然后循环生成table; 一般我们常用的只有单个表
    foreach ($this->getTableNames() as $tableName) {
        // model :
        $modelName = $this->generateClassName($tableName);
        $queryClassName = ($this->generateQuery) ? $this->generateQueryClassName($modelName) : false;
        $tableSchema = $db->getTableSchema($tableName);
        // 根据参数和db connections里的数据, 生成tablename/className等放到params里面

```

其中

```
$params = [
```

```

        'tableName' => $tableName,
        'className' => $modelName,
        'queryClassName' => $queryClassName,
        'tableSchema' => $tableSchema,
        // 这个地方可以跳进去看看, 如果设置了generateLabelsFromComments, 就是 id=>comments; 否则就是id=>name, 这里的name因为不是
        'labels' => $this->generateLabels($tableSchema),
        // 根据每个字段的属性, 增加rules的规则, 比如不为空的字段设为required, varchar类型的字段会有长度限制, unique index的字段会
        'rules' => $this->generateRules($tableSchema),
        'relations' => isset($relations[$tableName]) ? $relations[$tableName] : [],
    ];
}
// 这一步就是生产各个code文件的步骤, 对于model来讲其实只会生成对应的model文件
// CodeFile类是用来生成文件的类
// @ns 就是拿到namespace对应的dir + className 从而生成对应的php文件对应的地址 /home/yangchao/www/pay/models/Partner.php
// this->render就把params里面的内容渲染到了php文件中, 实际上这时候就已经生成了model文件里的所有内容
// 然后render的 template文件放在 console/template/model/model.php里面
$files[] = new CodeFile(
    Yii::getAlias('@' . str_replace('\\', '/', $this->ns)) . '/' . $modelName . '.php',
    $this->render('model.php', $params)
);

```

Like 2 people like this

No labels

地址：北京市朝阳区建国路86号佳兆业广场北塔6层梦想加空间601室

以太资本由艾普拉斯投资顾问(北京)有限公司运营，提供早期互联网项目的投融资对接服务

©2014-2017 以太资本 京ICP备14028208号