

目录

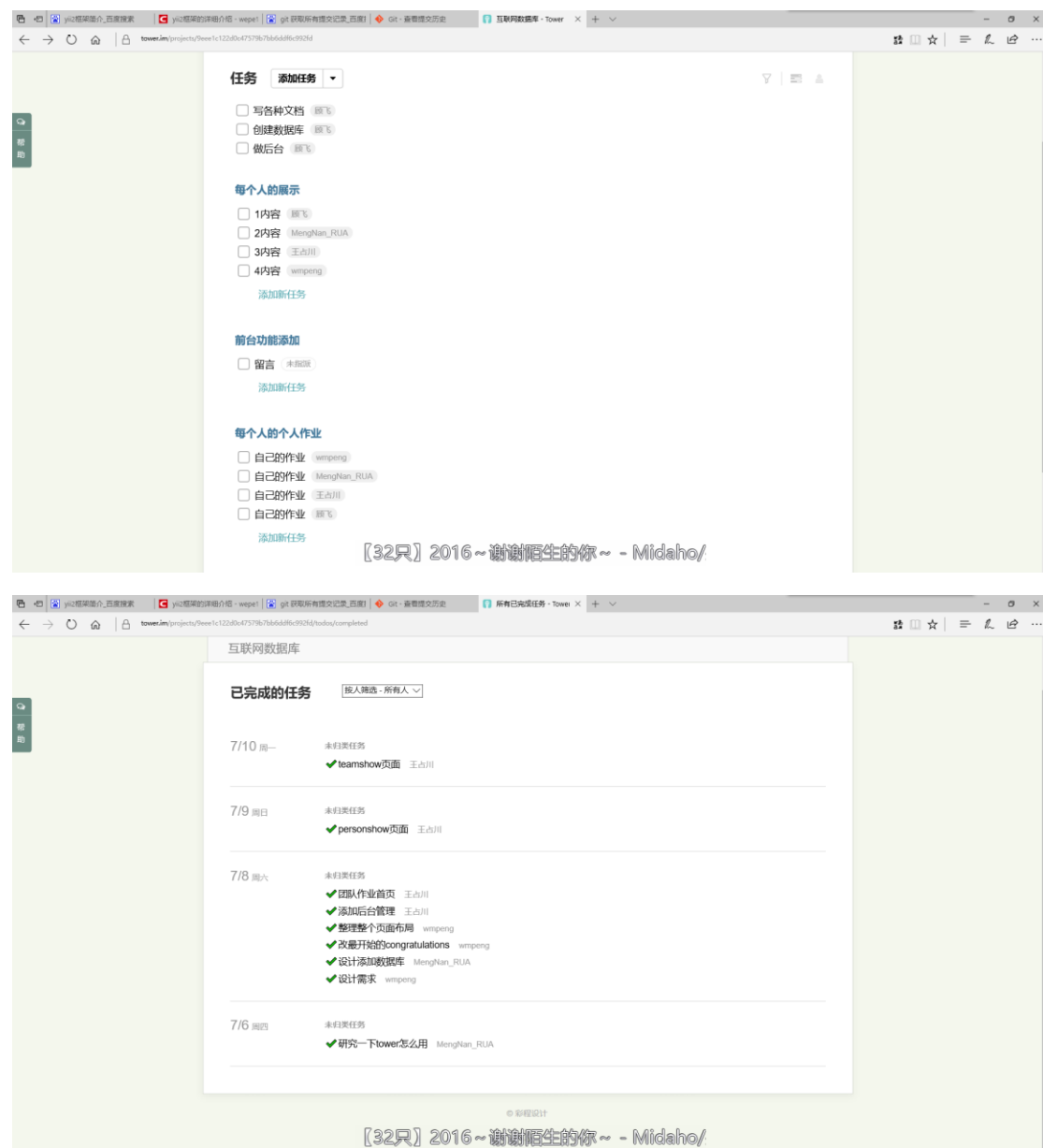
- 任务及分工2
 - Tower.....2
 - 表格.....3
- Git 记录3
- 主要代码.....4
 - 前台.....4
 - 留言4
 - 团队展示5
 - 个人展示6
 - 后台.....8
 - Controller.....8
 - View9
- 项目目录..... 10

任务及分工

本来是用的 Tower，但是由于联系上不如 qq 方便，再加上计划总是不如变化快，所以后来就放弃掉了。

Tower

以下是前期用 Tower 的记录：



表格

以下表格展示了成员的分工：

任务			人员
前台	主模板		王明鹏&王占川
	主跳转逻辑		王明鹏&王占川
	团队展示	模板&设计	王明鹏
		内容(MVC)	王占川
	个人展示	模板&设计	顾飞&王明鹏
		内容(MVC)	顾飞
	团队&个人作业		王明鹏
留言编辑&展示		王文豪	
后台	主模板		顾飞&王明鹏
	跳转逻辑		王明鹏
	创建团队		王明鹏
	创建队员		王明鹏
	修改团队		王明鹏
	修改队员		王明鹏
数据库设计			顾飞&王文豪&王明鹏&王占川
文档	需求文档		王明鹏
	设计文档		王文豪&王明鹏
	实现文档		王明鹏
	用户手册		王明鹏
	项目展示 PPT		王文豪
	部署文档		王明鹏

Git 记录

我们使用 github 来进行版本管理，我们都把代码直接提交到了 master 分支了，所以看起来有点乱，地址：<https://github.com/wmpeng/advanced2>

主要代码

前台

留言

Model

Gii 工具生成的默认的代码，就不贴了

Controller

重写了一部分代码，实现了自动存储日期

```
public function actionCreate()
{
    $model = new MsgMessage();
    if ($model->load(Yii::$app->request->post())) {
        $model->msgTime=date("Y-m-d H:i:s");
        $model->save();
        //return $this->redirect(['view', 'id' => $model->msgId]);
        return $this->redirect(['index']);
    } else {
        return $this->render('create', [
            'model' => $model,
        ]);
    }
}
```

View

新建留言：

```
<?php $form = ActiveForm::begin(); ?>
<?php echo $showtime=date("Y-m-d H:i:s");?>
<?= $form->field($model, 'msgText')->textarea(['rows' => 6]) ?>
<?= $form->field($model, 'msgWriter')->textarea(['rows' => 1]) ?>
<div class="form-group">
    <?= Html::submitButton($model->isNewRecord ? 'Create' : 'Update', ['class' =>
$model->isNewRecord ? 'btn btn-success' : 'btn btn-primary']) ?>
</div>
```

显示留言：

```
<?= GridView::widget([
    'dataProvider' => $dataProvider,
    'filterModel' => $searchModel,
    'columns' => [
        'msgId',
        'msgTime',
        'msgText:ntext',
        'msgWriter:ntext',
    ],
]); ?>
```

团队展示

Model

```
public static function findByReceiver()
{
    $qry=new query();
    $qry->select('stuld,
                stuName,
                stuGender,
                stuGrade,
                teamName')
        ->from('info_student,info_team')
        ->where('info_team.teamId=666 and info_student.teamId=info_team.teamId')
        ->orderBy('stuld');
    return $qry->all();
}
```

Controller

```
public function actionTeamshow()
{
    $msgList=Message::findByReceiver();
    return $this->render('teamshow',
        ['msgList'=>$msgList]
    );
}
```

View

展示队员的个人信息

```
<div class="col-md-4 fh5co-staff">
    
    <h3><?=$msgList[2]['stuName']?></h3>
    <h4>学号：<?=$msgList[2]['stuld']?></h4>
    <p>性别：<?=$msgList[2]['stuGender']?>
        <br/>年级：<?=$msgList[2]['stuGrade']?>
        <br/>队伍：<?=$msgList[2]['teamName']?>
    </p>
    <a href="index.php?r=site/wwhindex" target="_blanket">个人主页
    </a>>
</div>
```

个人展示

个人展示是做了四份的，就放上来一个。【其实我觉得其实就做一个就可以】

Model

不能用作别的队伍，有一些静态的思想在这里，但是只作为我们队的展示还是可以的

```
public static function findByReceiver()
{
    $qry=new query();
    $qry->select('stuld,
                stuName,
                stuGender,
                stuGrade,
                teamName')
        ->from('info_student,info_team')
        ->where('stuName="王文豪" and info_student.teamId=info_team.teamId')
        ->orderBy('stuld');
    return $qry->all();
}
```

Controller

```
public function actionWwhindex()
{
```

```

/* $msgList=Gufei::findByReceiver();
return $this->render('gfindex',
    ['msgList'=>$msgList]
); */
$msgList=Gufei3::findByReceiver();
$msgListB=GufeiB3::findByReceiver();
$msgListM=GufeiM3::findByReceiver();
$msgListG=GufeiG3::findByReceiver();
return $this->render('wwhindex',
    ['msgList'=>$msgList,
    'msgListB'=>$msgListB,
    'msgListM'=>$msgListM,
    'msgListG'=>$msgListG]
);
}

```

View

个人信息部分

```

<div class=" col-lg-7 col-md-7 col-sm-7 col-xs-12 pull-left media">
    <h5>姓名：<?=$msgList[0]['stuName']?><br></h5>
    <h4>学号：<?=$msgList[0]['stuld']?><br></h4>
    <h4>性别：<?=$msgList[0]['stuGender']?><br></h4>
    <h4>年级：<?=$msgList[0]['stuGrade']?><br></h4>
</div>

```

显示爱好部分

```

<div class="row">
    <div class="col-md-12">
        <div class="heading">
            <h2>MOVIE</h2>
            <h3>It is impossible to love and to be wise--Francis Bacon </h3>
        </div>
    </div>
    <div class="col-md-12">
        <h4>影名：<?=$msgListM[0]['movieName']?><br></h4>
        <h4>导演：<?=$msgListM[0]['movieDirector']?><br></h4>
        <h4>类别：<?=$msgListM[0]['movieType']?><br></h4>
    </div>
</div>
</div>

```

后台

因为后台基本不需要其他功能，所以 Model 层基本都是用 gii 生成的默认的，就不贴了。
创建/修改队伍和队员基本一样，就以队伍为例了

Controller

这是查看队伍信息的函数

```
public function actionViewt($id)
{
    return $this->render('viewt', [
        'modelit' => $this->findModelit($id),
        'modelt' => $this->findModelt($id),
    ]);
}
```

这是创建队伍的函数

```
public function actionCreatet()
{
    $modelit = new InfoTeam();
    $modelt = new ShowTeam();
    if ($modelit->load(Yii::$app->request->post()) &&
    $modelt->load(Yii::$app->request->post()))
    {
        $modelit->memberNum=0;
        $modelit->save();
        $modelt->teamId=$modelit->teamId;
        $modelt->save();
        return $this->redirect(['viewt', 'id' => $modelit->teamId]);
    } else {
        return $this->render('createt', [
            'modelit' => $modelit,
            'modelt' => $modelt,
        ]);
    }
}
```

这是修改队伍的函数

```
public function actionUpdatet($id)
{
    $modelit = $this->findModelit($id);
    $modelt = $this->findModelt($id);
    if ($modelit->load(Yii::$app->request->post()) &&
```



```

$modelt->load(Yii::$app->request->post()))
{
    $modelit->save();
    $modelt->teamId=$modelit->teamId;
    $modelt->save();
    return $this->redirect(['view', 'id' => $modelit->teamId]);
} else {
    return $this->render('updatet', [
        'modelit' => $modelit,
        'modelt' => $modelt,
    ]);
}
}

```

View

查看队伍信息 (viewt.php)

就是用了两个 widget 来显示信息

```

<?= DetailView::widget([
    'model' => $modelit,
    'attributes' => ['teamId','teamName:text','memberNum'],
])
?>
<?= DetailView::widget([
    'model' => $modelt,
    'attributes' => ['picture:text','slogan:text'],
])
?>

```

创建(createt.php)

```

<?php $form = ActiveForm::begin(); ?>

<?= $form->field($modelit, 'teamName')->textarea(['rows' => 1]) ?>
<?= $form->field($modelt, 'picture')->textarea(['rows' => 1]) ?>
<?= $form->field($modelt, 'slogan')->textarea(['rows' => 6]) ?>
<div class="form-group">
    <?= Html::submitButton($modelit->isNewRecord ? 'Create' : 'Update', ['class' =>
    $modelit->isNewRecord ? 'btn btn-success' : 'btn btn-primary']) ?>
</div>

```

修改信息(updatet.php)

其实和创建是一样的，只是传进来的 modelit 一个是 new 出来的，一个是根据 id 找的

```

<?php $form = ActiveForm::begin(); ?>

```

```

<?= $form->field($modelit, 'teamName')->textarea(['rows' => 1]) ?>
<?= $form->field($modelit, 'picture')->textarea(['rows' => 1]) ?>
<?= $form->field($modelit, 'slogan')->textarea(['rows' => 6]) ?>
<div class="form-group">
    <?= Html::submitButton($modelit->isNewRecord ? 'Create' : 'Update', ['class'
=> $modelit->isNewRecord ? 'btn btn-success' : 'btn btn-primary']) ?>
</div>

```

项目目录

只写出了存在改动并且在最终的项目中用到的目录：

advance2	项目文件夹
├─backend	后台文件夹，前后台是分离的
│ └─controllers	后台的 controller 层，包括了 SiteController 和 TtController
│ └─models	后台的 model 层，每个表都对应 1~2 个 model
│ └─views	后台的 view 层
│ └─layouts	后台的布局用的是这个目录下的 main_layout.php
│ └─site	基本没有变化
│ └─tt	自己写的几个 view 都在这里
│ └─web	用到一些 js 和 css 等
├─common	
│ └─config	只修改 main-local 的数据库
├─data	
│ └─document	需求、设计、实现等各种文档
│ └─personal	个人作业
│ └─advanced.sql	数据库文件
│ └─pic	用到的图片
└─frontend	前台文件夹
└─controllers	留言功能新建了 MsgMessageController ; 其他没有新建 Controller，修改了 SiteController
└─models	新建的 model 都在这里
└─views	
└─layouts	布局文件用的是 main_layout（和后台的不一样）
└─msg-message	留言功能的 view 层
└─site	其他的 view 层，写的点乱
└─web	用到一些 js 和 css 等