

06. 数据库与模型

学习要点：

1. 连接数据库
2. 模型定义

本节课我们来开启数据库连接搭配创建模型的方式来获取数据库里的数据。

一. 连接数据库

1. ThinkPHP 采用内置抽象层将不同的数据库操作进行封装处理；
2. 数据抽象层基于 PDO 模式，无须针对不同的数据库编写相应的代码；
3. 使用数据库的第一步，就是连接你的数据库；
4. 在根目录的 `config` 下的 `database.php` 可以设置数据库连接信息；
5. 大部分系统已经给了默认值，你只需要修改和填写需要的值即可；

```
// 数据库类型
'type'           => 'mysql',
// 服务器地址
'hostname'       => '127.0.0.1',
// 数据库名
'database'       => 'grade',
// 用户名
'username'       => 'root',
// 密码
'password'       => '123456',
// 端口
'hostport'       => '3306',
// 编码
'charset'        => 'utf8',
// 数据库表前缀
'prefix'         => 'tp_',
```

6. `type` 属性默认支持的数据库有：`mysql`、`sqlite`、`pgsql`、`sqlsrv`；
7. 还有其它很多连接的细节和方式，需要在具体问题中或项目才能更好理解；
8. 比如：字符串连接 `dsn`、多模块、动态连接等，这里都暂略；
9. 配置完数据库，我们使用如下代码，在控制器端输出 `mysql` 里的数据；

```
public function getNoModelData()
{
    // $data = Db::table('tp_user')->select();
    $data = Db::name('user')->select();
    return json($data);
}
```

二. 模型定义

1. 在 MVC 中, 我们已经使用过 Controller(C), View(V), 剩下一个就是 Model(M);
2. Model 即模型, 就是处理和配置数据库的相关信息;
3. 在项目应用根目录创建 model 文件夹, 并且创建 User.php;

```
namespace app\model;\nuse think\\Model;\n\nclass User extends Model\n{\n\n}
```

4. 当创建了 User 模型了, 控制器端可以这么写:

```
public function getModelData()\n{\n    $data = User::select();\n    return json($data);\n}
```

5. 而此时, 命名空间会自动导入 User 模型: use app\\model\\User;
6. 很多时候, 我们需要调试 SQL 是否正确, 建议打开 Trace, 可以查看原生 SQL;

```
// 应用 Trace\n'app_trace' => true,
```