

Introduction to MVC Framework

Database Laboratory 1/2017

Titipat <ttpsvb@gmail.com>

Recap: ORM

- Use OOP to mapping data relations.
- CRUD
 - HTTP POST -> Create
 - HTTP GET -> Read
 - HTTP UPDATE -> Update
 - HTTP DELETE -> Delete

CRUD: Create

```
<?php
```

```
$post = new Post();
```

```
$post->title = 'Today is a good day';
```

```
$post->user_id = MySession::currentUser()->id;
```

```
$post->save();
```

CRUD: Read

```
<?php
```

```
$post = Post::where('title', 'Today is a good day');  
// render with template engine  
render('post.view', $post);
```

CRUD: Update

```
<?php
```

```
$post = Post::where('title', 'Today is a good day');  
$post->title = 'Today is a very good day';
```

```
$post->save();
```

CRUD: Delete

```
<?php
```

```
$post = Post::where('title', 'Today is a very good day');
```

```
$post->delete();
```

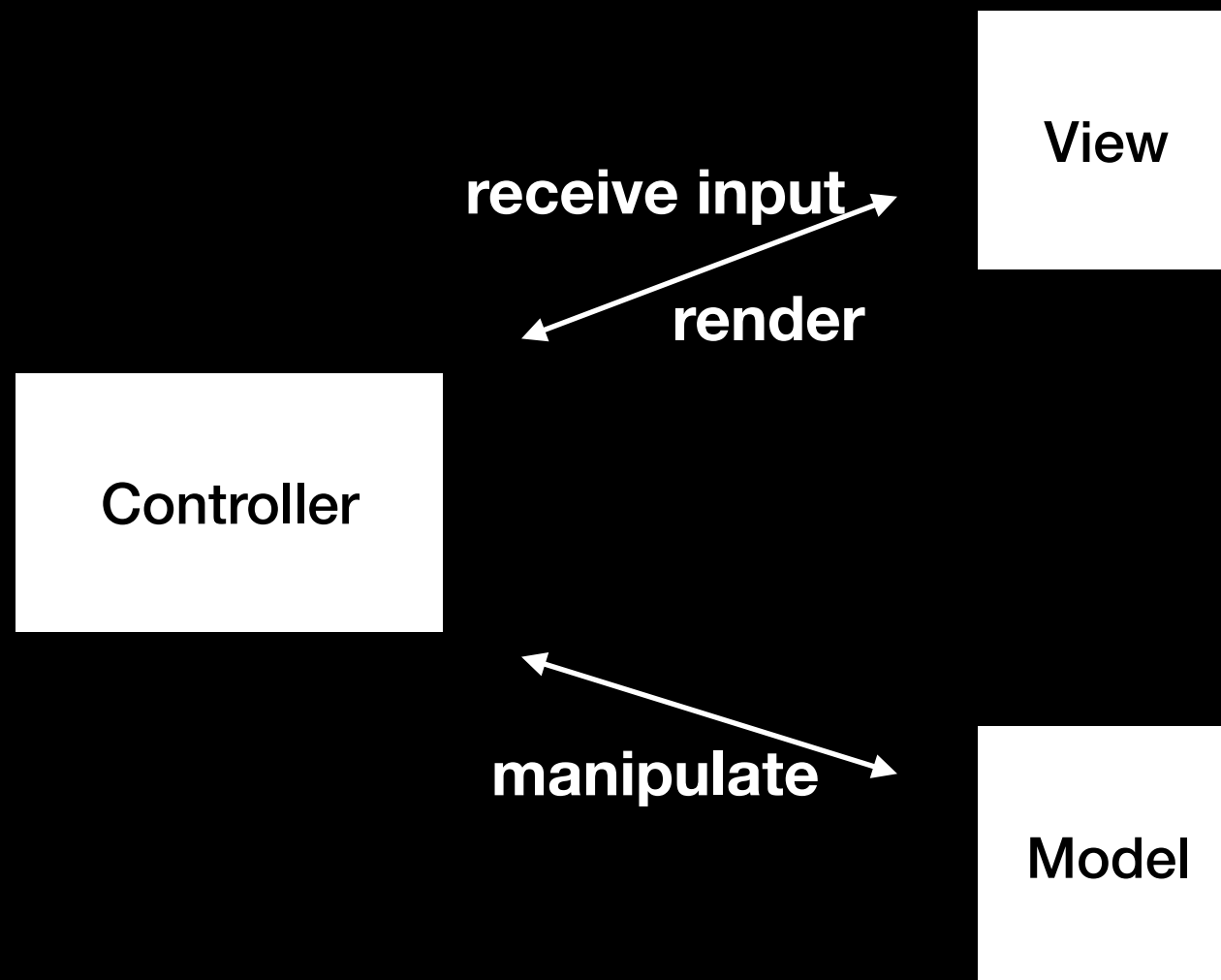
MVC

- It a software architecture design pattern.
- It divides the whole system into 3 components:
 1. Models main component, represented from problem domain
 2. Views interact with users and send input to the controller.
 3. Controllers process tasks when its triggered by any events.

MVC Benefits

- Simultaneous development.
- Can be focus on exact points.
- Maintainable code.
- Code reusable (DRY: Don't repeat yourself)
 - Authentication, authorization middleware.
 - Inheritable template.
 - Model; no more WET (Write everything twice) SQL.

MVC Diagram



Router

- It's an application router. Don't confuse with layer 3-4 router.
- The router will handle every request sent to the framework.
- Router compares REQUEST METHOD and REQUEST PATH with existed ROUTES.
- It's a sequential comparison, like access control list.

Example routing.

- Considering this following routes
 - ANY * - This will catch any routes.
 - GET /posts - will catch <http://localhost/posts> and should return an array of posts
 - GET /posts/1 - will catch a request for post which has id equal to 1
 - POST /posts
 - UPDATE /posts/2
 - DELETE /posts/3

References

- The MVC Loosens and Refocuses Programming - Learning PHP Design Patterns by William Sanders