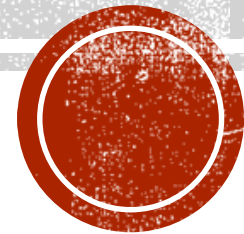
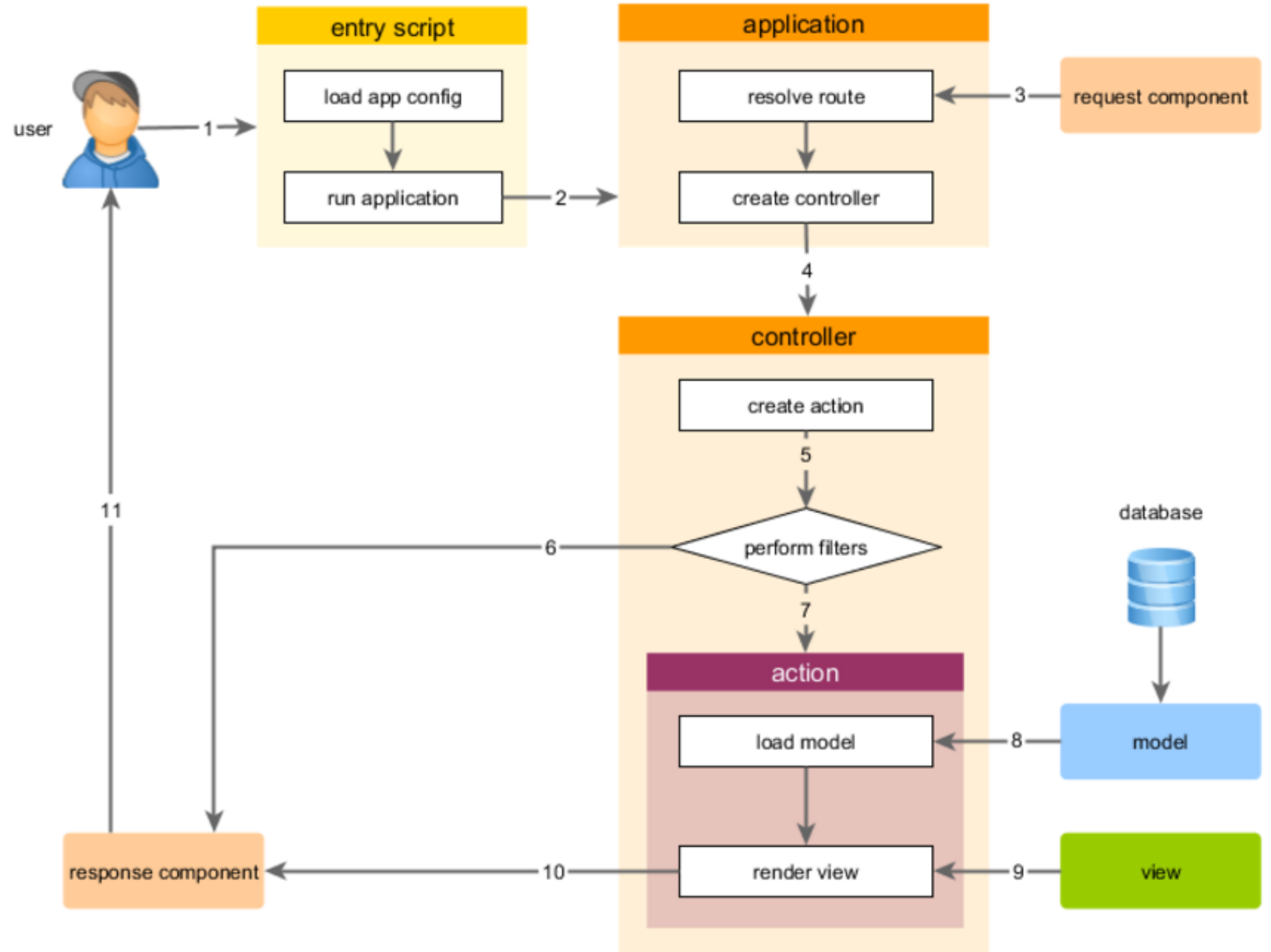


# Yii2: Handling Requests

Oleh: Ahmad Syauqi Ahsan



# Overview



# Proses Request

Setiap kali aplikasi Yii2 menerima request akan melalui proses-proses berikut ini:

1. User membuat request ke *entry script* web/index.php
2. *Entry script* memuat konfigurasi (config.php) dan membuat sebuah *application instance*
3. Aplikasi akan menetapkan *route* dengan bantuan komponen aplikasi yang bernama *request*
4. Aplikasi membuat *controller instance* untuk menangani request
5. Controller membuat *action instance* dan menjalankan *filters* untuk *action* tersebut

# Proses Request (2)

6. Jika ada *filter* yang tidak terpenuhi, maka *action* akan dibatalkan
7. Jika semua *filter* terpenuhi, maka *action* akan dijalankan
8. *Action* akan memuat data model, kemungkinan besar dari database
9. *Action* akan me-render *view*, dengan menyertakan data model
10. Hasil dari *view* yang telah di-render akan dikembalikan ke komponen aplikasi bernama *response*
11. Komponen *response* akan mengirimkan hasil proses render ke browser pengguna (dalam bentuk html, css, dan/atau *client-side* script lainnya)

# Bootstrapping

- Bootstrapping adalah proses mempersiapkan *environment* sebelum aplikasi mulai menerima dan memproses *request* yang datang
- Bootstrapping dilakukan pada dua tempat: *entry script* dan *aplikasi*
- Pada *entry script*, dua kelas autoloader untuk library2 berbeda didaftarkan: Composer autoloader melalui `autoload.php` dan Yii autoloader melalui file class dari Yii.
- Pada *constructor* dari aplikasi, beberapa hal ini akan dikerjakan:
  - Memanggil `yii\base\Application::preInit()` -> melakukan konfigurasi property2 penting seperti property *basePath*
  - Meregister `yii\base\Application::errorHandler`
  - Menginisialisasi property2 dari aplikasi menggunakan file konfigurasi
  - Memanggil `yii\base\Application::init()` yang nantinya akan memanggil `yii\base\Application::bootstrap` untuk menjalankan komponen bootstrapping
  - Meng-include-kan file extension manifest di `vendor/yiisoft/extensions.php`
  - Membuat dan menjalankan komponen aplikasi dan/atau module2 yang dideklarasikan pada property `bootstrap` dari aplikasi.

# Bootstrapping (2)

- Karena bootstrapping dilakukan sebelum menangani request, pastikan hanya komponen yang penting saja yang dimasukkan dalam proses bootstrapping ini.
- Contoh: pada mode production, anda mungkin perlu meng-enable-kan APT cache pada proses *bootstrapping*.

# Routing

- Ketika metode `yii\web\Application::run()` dipanggil oleh *entry script*, hal pertama yang dilakukan adalah *me-resolve* request yang datang kemudian menjalankan controller action untuk menangani request tersebut. Proses ini disebut *Routing*
- Proses *resolve* ditangani oleh metode `yii\web\Request::resolve()`, yang akan menghasilkan:
  - Route valid -> controller action yang sesuai dipanggil
  - Route tidak valid -> dimunculkan `yii\web\NotFoundHttpException`
  - Tidak ada route -> controller action pada property *defaultRoute* akan dipanggil
- *catchAll Route* adalah kondisi spesial dimana semua request akan diarahkan ke sebuah controller action (misal aplikasi sedang dalam fase *maintenance*). Hal ini dapat dilakukan dengan mengisi property *catchAll* di file konfigurasi (`web.php`) dengan controller action yang dikehendaki.

```
$config = [  
    'name' => 'SI Jurusan',  
    'id' => 'sijur',  
    'catchAll' => ['site/offline'],  
    'basePath' => dirname(__DIR__),  
    'bootstrap' => ['log'],  
    'components' => [  
        'urlManager' => [  
            'enablePrettyPath' => true,  
            'enableStrictParsing' => true,  
        ],  
    ],  
];
```

# Requests

- Request direpresentasikan dalam bentuk objek `yii\web\Request`, yang didalamnya terdapat informasi seperti: parameter dari request, HTTP headers, cookies, dll.
- Parameter dari request bisa didapat dengan memanggil metode `yii\web\Request::get()` dan `yii\web\Request::post()` yang berturut-turut akan mengembalikan nilai `$_GET` dan `$_POST`. Contoh:
  - `$get = Yii::$app->request->get();`
  - `$id = Yii::$app->request->get('id');`
  - `$post = Yii::$app->request->post();`
  - `$name = Yii::$app->request->post('name');`



# Requests (2)

- Anda dapat melihat metode HTTP yang digunakan oleh sebuah Request melalui fungsi2 seperti berikut ini:
  - `if (Yii::$app->request->isAjax)`
  - `if (Yii::$app->request->isGet)`
  - `if (Yii::$app->request->isPost)`
- Informasi HTTP header dari sebuah request dapat diakses melalui property `yii\web\Request::headers`. Contoh:
  - `$accept = Yii::$app->request->headers->get('Accept')`
  - `if (Yii::$app->request->headers->has('User-Agent'))`
- Anda juga dapat melihat hostname dan alamat IP dari computer pengguna melalui:
  - `$userHostname = Yii::$app->request->userHost;`
  - `$userIP = Yii::$app->request->userIP;`

# Request URLs

Anda dapat melihat bagian dari request URL melalui atribut berikut ini: (contoh: URL yang diminta adalah <http://example.com/admin/index.php/product?id=100>)

- **url**: returns /admin/index.php/product?id=100, which is the URL without the host info part.
- **absoluteUrl**: returns http://example.com/admin/index.php/product?id=100, which is the whole URL including the host info part.
- **hostInfo**: returns http://example.com, which is the host info part of the URL.
- **pathInfo**: returns /product, which is the part after the entry script and before the question mark (query string).
- **queryString**: returns id=100, which is the part after the question mark.
- **baseUrl**: returns /admin, which is the part after the host info and before the entry script name.
- **scriptUrl**: returns /admin/index.php, which is the URL without path info and query string.
- **serverName**: returns example.com, which is the host name in the URL.
- **serverPort**: returns 80, which is the port used by the Web server.

# Responses

- Setelah aplikasi selesai menangani sebuah request, objek `yii\web\Response` akan dibuat dan dikirimkan ke pengguna.
- Objek response ini berisi informasi seperti HTTP status code, HTTP headers dan body.
- Untuk mengirimkan HTTP header dapat anda lakukan dengan memanipulasi property `yii\web\Response::headers`. Contoh:
  - `$headers = Yii::$app->response->headers;`
  - `// add a Pragma header. Existing Pragma headers will NOT be overwritten.`  
`$headers->add('Pragma', 'no-cache');`
  - `// set a Pragma header. Any existing Pragma headers will be discarded.`  
`$headers->set('Pragma', 'no-cache');`
- Pada kasus-kasus sederhana, kita tidak perlu melakukan perubahan/penambahan pada objek response ini. Aplikasi Yii sudah membuat objek ini secara otomatis melalui *return value* dari controller action.

```
public function actionIndex()  
{  
    return $this->render('index');  
}
```

# Sessions dan Cookies

- Pada PHP murni, sessions dan cookies dapat diakses melalui variable global `$_SESSION` dan `$_COOKIE`
- Pada Yii, session dan cookies dikelola dalam sebuah objek (class) yang dapat diakses melalui variable:
  - Untuk sessions dapat diakses melalui `Yii::$app->session`
  - Untuk cookies dapat diakses melalui `Yii::$app->request->cookies` dan `Yii::$app->response->cookies`

```
$session = Yii::$app->session;
```

```
// get a session variable. The following usages  
are equivalent:
```

```
$language = $session->get('language');  
$language = $session['language'];
```

```
// set a session variable. The following usages  
are equivalent:
```

```
$session->set('language', 'en-US');  
$session['language'] = 'en-US';  
$_SESSION['language'] = 'en-US';
```

```
// get the cookie collection  
(yii\web\CookieCollection) from the "request"  
component
```

```
$cookies = Yii::$app->request->cookies;
```

```
// get the "language" cookie value. If the cookie  
does not exist, return "en" as the default value.  
$language = $cookies->getValue('language', 'en');
```

```
// add a new cookie to the response to be sent  
$cookies = Yii::$app->response->cookies;  
$cookies->add(new \yii\web\Cookie([  
    'name' => 'language',  
    'value' => 'zh-CN',  
]));
```

# Latihan

- Buat table "statistic" di database yang memiliki kolom seperti pada gambar disamping!
- Buat Model di aplikasi anda untuk table "statistic" tersebut.
- Tambahkan proses untuk mencatat informasi (waktu akses, IP pengguna, nama computer pengguna, path info, dan query string) dari request yang dilakukan oleh pengguna pada route "item/index" dan "item/view" dari aplikasi anda!



```
my_basic_statistic
id : int(11)
access_time : datetime
user_ip : varchar(20)
user_host : varchar(50)
path_info : varchar(50)
query_string : varchar(50)
```



**Terima Kasih**