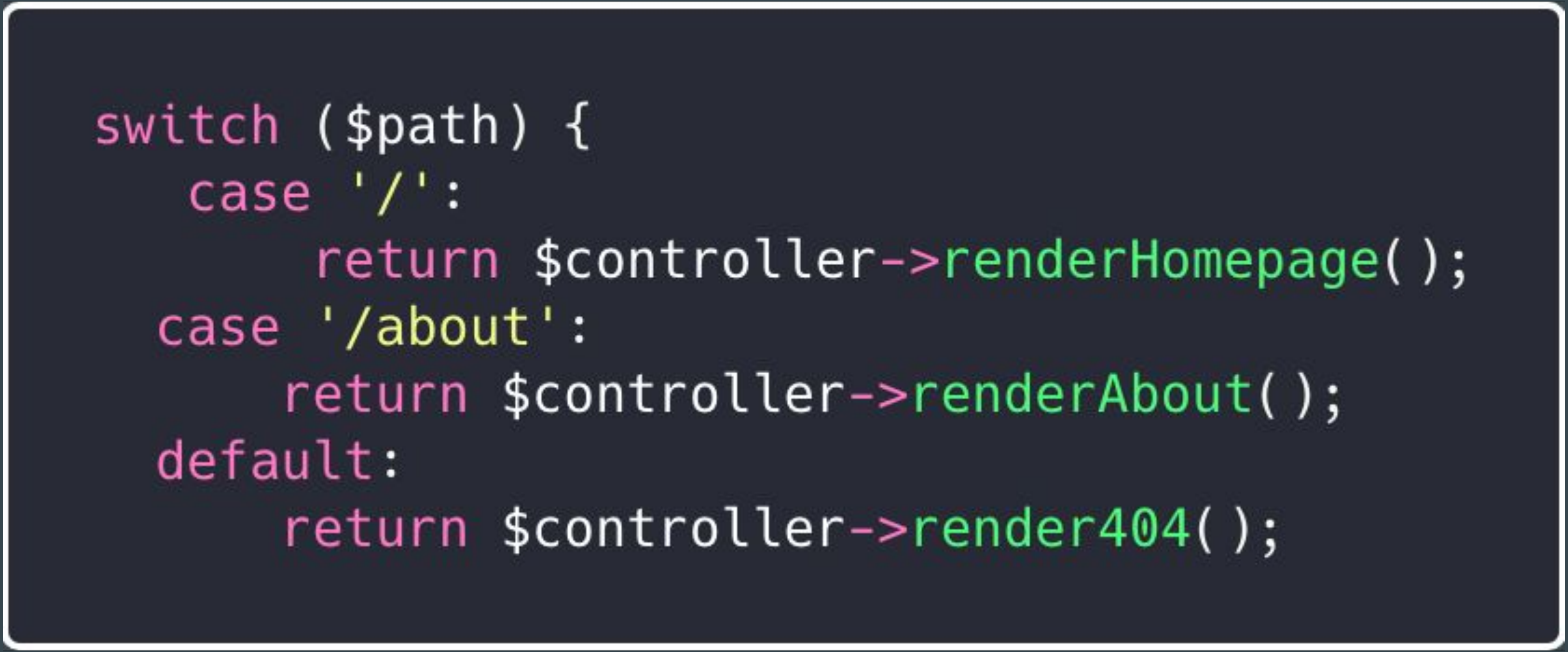
Routing & Controllers

Routing component

Transform Request object to array with routing information



```
switch ($path) {
   case '/':
       return $controller->renderHomepage();
  case '/about':
      return $controller->renderAbout();
  default:
      return $controller->render404();
```

Routing formats

- Yaml
- Annotation
- PHP
- XML

All formats are compiled to PHP code

Routing configuration

- Route Annotation
  - Usage
  - Loading
- Loading priority is important

Symfony Router

- RouteCollection
- UrlMatcherInterface
- UrlGeneratorInterface
- RouterInterface
- Router

Optimization

- Compile routes to PHP
- Dump cached matcher, single class
- Group similar routes
- Prefer strpos, use regex only when needed
- Possessive quantifiers in regex
- RouteCollection performance issue

Controller

Handle Request to return the Response

```php
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\HttpFoundation\Response;

$request = Request::createFromGlobals();
$path = $request->getPathInfo();

if (in_array($path, array('', '/')))  {
   $response = new Response('Home page.');
} elseif ('/about' === $path) {
   $response = new Response(’About me’);
} else {
   $response = new Response('Page not found.', Response::HTTP_NOT_FOUND);
}
$response->send();
```

Controller arguments

Resolved using Reflection

- Request object
- Route placeholder name match argument name
- ParamConverter and match by type
- Autowiring bindings
- Optional parameters from query params

Controller Dependencies

- type-hint an argument

```php
/**
* @Route("/lucky/number/{max}")
*/
public function numberAction($max, LoggerInterface $logger) {
}
```

- bind the argument by its name

```yaml
# explicitly configure the service
AppBundle\Controller\LuckyController:
  public: true
  bind:
      # for any $logger argument, pass this specific service
      $logger: '@monolog.logger.doctrine'
```

Controller Dependencies

- Extend Controller
- Extend AbstractController
- ControllerTrait
- Dependency Injection without Extend

Override Controller

- Don’t load the original routing
- Bundle inheritance
- Register route before the original one with the new controller
- Override service if controller defined as a service

Best Practice

Symfony follows the philosophy of "thin controllers and fat models".

- **>= 5 constants**
- **>= 10 actions**
- **>= 20 lines of code in each action**

If you have more - think about services.

Debugging

- debug:router category_show
- router:match /category/403
- Symfony profiler tab
- debug:autowiring