跟踪symfony3记录

1. 自动加载

自动加载完成后，ClassLoader：：loadClass 还有 swift里也有一个自动加载得函数。

1. Debug

调用Debug::enable();

ExceptionHandler::register();

set\_exception\_handler(array($handler, 'handle'))

ErrorHandler::register(new ErrorHandler(new BufferingLogger()));

$this->loggedErrors;是记录得错误，$this->loggers;是记录得方式

register\_shutdown\_function(\_\_CLASS\_\_.'::handleFatalError');

set\_error\_handler(array($handler, 'handleError'), $handler->thrownErrors | $handler->loggedErrors);

set\_exception\_handler(array($handler, 'handleException'))

public function setExceptionHandler(callable $handler = null)

{

$prev = $this->exceptionHandler;

$this->exceptionHandler = $handler;

return $prev;

}

DebugClassLoader::enable();

总结：注册错误处理函数，开启调试，把自动加载接入调试里。

3、

class AppKernel extends Kernel

abstract class Kernel implements KernelInterface, RebootableInterface, TerminableInterface

interface KernelInterface extends HttpKernelInterface, \Serializable

interface HttpKernelInterface

$this->checkClass($class, $file);

$deprecations = $this->checkAnnotations($refl, $name);

$parentAndOwnInterfaces = $this->getOwnInterfaces($class, $parent);

1. $kernel = new AppKernel('dev', true);

public static function createFromGlobals()

$request = self::createRequestFromFactory($\_GET, $\_POST, array(), $\_COOKIE, $\_FILES, $server);

$response = $kernel->handle($request);

$this->initializeBundles();

public function registerBundles()

实例化bundle经过的类：

new Symfony\Bundle\FrameworkBundle\FrameworkBundle(),

class FrameworkBundle extends Bundle

abstract class Bundle implements BundleInterface

use Symfony\Component\DependencyInjection\ContainerAwareTrait;

trait ContainerAwareTrait

interface BundleInterface extends ContainerAwareInterface

interface ContainerAwareInterface

new Symfony\Bundle\SecurityBundle\SecurityBundle(),

class SecurityBundle extends Bundle

new Symfony\Bundle\TwigBundle\TwigBundle(),

$this->initializeContainer();

$cache = new ConfigCache($cacheDir.'/'.$class.'.php', $this->debug);

{

$checkers = array(new SelfCheckingResourceChecker());

parent::\_\_construct($file, $checkers);

}

class ConfigCache extends ResourceCheckerConfigCache

{

$this->resourceCheckers = $resourceCheckers;

}

class ResourceCheckerConfigCache implements ConfigCacheInterface

interface ConfigCacheInterface

class SelfCheckingResourceChecker implements ResourceCheckerInterface

interface ResourceCheckerInterface