

Phalcon\Cache\Exception: Cache file /home/dev.crusit.bg/public_html/data/cache/queries/settings-all could not be opened

/home/dev.crusit.bg/public_html/common/models/Settings.php (20)

- [Backtrace](#)
- [Request](#)
- [Server](#)
- [Included Files](#)
- [Memory](#)

#0 [Phalcon\Cache\Backend\File](#)->get(settings-all, 3600)

#1 [Phalcon\Mvc\Model\Query](#)->execute()

/home/dev.crusit.bg/public_html/common/models/Settings.php (20)

```
<?php
/**
 * @file Settings.php
 * @package Framework v1.0
 * @author Nikolay Mihaylov <n.mihayloff@gmail.com>
 */
namespace Models;

class Settings extends BaseModel
{
    public function getAll()
    {
        $cacheFile = 'settings-all';
        $builder = $this->modelsManager->createBuilder();
        $builder->columns([
            'keyword',
            'content',
        ]);
        $builder->from(['main' => 'Models\Settings']);
        $items = $builder->getQuery()->cache(['key' => $cacheFile])->execute();

        $collection = [];
        foreach ($items as $item) {
            $collection[$item->keyword] = $item->content;
        }
        return $collection;
    }
}
```

#2 [Models\Settings](#)->getAll()

/home/dev.crusit.bg/public_html/apps/frontend/controllers/BaseController.php (41)

```
<?php
/**
 * @file BaseController.php
 * @package Framework v1.0
 * @author Nikolay Mihaylov <n.mihayloff@gmail.com>
 */
namespace Frontend\Controllers;

class BaseController extends \Phalcon\Mvc\Controller
{
    public $additionalAssets;

    public function initialize()
    {
        // Restricted access to dev site
        if ($this->config->isWebsiteIpRestricted AND !in_array($this->request->getServer('REMOTE_ADDR'), (array) $this->config->cms->ipRestriction)) {
            //die('Forbidden');
        }

        // Detect language
        $this->detectLanguage();

        // Load global cache files
        $this->loadCache();

        // Cookie specific
        $this->cookieSpecific();

        // Current route name
        $this->view->currentRouteName = "";
        if ($this->router->getMatchedRoute()) {
            $this->view->currentRouteName = $this->router->getMatchedRoute()->getName();
        }

        // Others..
    }
}
```

```

$this->view->assetsVersion = $this->config->assetsVersion;

// Update: 2021-10-15: temporary disable live chat
// $this->view->isChatOperatorAvailable = date('G') >= 9 && date('G') < 19 && !in_array(date('l'), ['Saturday', 'Sunday']);
$this->view->isChatOperatorAvailable = false;
$this->view->globalSettings = (new \Models\Settings)->getAll();

// Countdown bar
if (
    $this->config->site->countdownBar->isActive === true AND
    time() >= strtotime($this->config->site->countdownBar->from) AND
    time() <= strtotime($this->config->site->countdownBar->to)
){
    $this->view->countdownBar = $this->config->site->countdownBar;
    $this->additionalAssets('js', 'countdown-timer');
    $this->additionalAssets('css', 'countdown-timer');
}

}

// Cache
private function loadCache()
{
    // Translations
    $file = 'public-' . $this->session->language;
    $this->loadLanguageFile('public', $file);

    // Supermenu pages and Static menu links
    $pages = new \Models\Pages();
    $this->view->supermenu = $pages->getForSupermenu();
    $this->view->staticMenuLinks = $pages->getStaticLinks();

    // Form filters
    $this->view->formFilters = \Helpers\Cruises::getFormFilters();
    $this->view->lastCruiseDate = (new \Models\CruisePricesCache()->getLastCruiseDate());
}

// Cookie specific actions
private function cookieSpecific()
{
    // Newsletter popup - First time visiting Crusit from Usit
    if (
        !$this->cookies->has('firstVisitFromUsit') AND
        strpos($this->request->getServer('HTTP_REFERER'), 'usitcolours') !== false
    ){
        $this->additionalAssets('script', 'first-visit-from-usit-popup');
        $this->cookies->set('firstVisitFromUsit', 1, strtotime('+7 days'));
    }

    // Newsletter popup - First visit for a month
    if (!$this->cookies->has('firstVisitThisMonth')) {
        $this->additionalAssets('script', 'trigger-newsletter-popup');
    }

    // Special popup
    // if (!$this->session->has('isSpecialPopupDisplayed')) {
    //     $this->session->isSpecialPopupDisplayed = 1;
    //     $this->additionalAssets('script', 'trigger-special-popup');
    // }

    // Cookies Assets
    $this->additionalAssets('js', 'cookies-v2');
    $this->additionalAssets('css', 'cookies-v2');

    // Cookie consent settings
    $this->view->cookiesSettings = \Helpers\CookiesConsent::get();
    $this->view->cookiesSettingsAcceptedEverything = true;

    // Facebook
    if (isset($this->view->cookiesSettings['marketingFacebook']) AND $this->view->cookiesSettings['marketingFacebook'] == 1) {
        $this->view->cookiesSettingsAllowFacebook = 1;
    } else {
        $this->view->cookiesSettingsAcceptedEverything = false;
    }

    // Google
    if (isset($this->view->cookiesSettings['marketingGoogle']) AND $this->view->cookiesSettings['marketingGoogle'] == 1) {
        $this->view->cookiesSettingsAllowGoogle = 1;
    } else {
        $this->view->cookiesSettingsAcceptedEverything = false;
    }

    // OpenX
    if (isset($this->view->cookiesSettings['marketingOtherAdvertisers']) AND $this->view->cookiesSettings['marketingOtherAdvertisers'] == 1) {
        $this->view->cookiesSettingsAllowBannerSystem = 1;
    } else {
        $this->view->cookiesSettingsAcceptedEverything = false;
    }

    // Statistics
    if (isset($this->view->cookiesSettings['statisticsGoogleAnalytics']) AND $this->view->cookiesSettings['statisticsGoogleAnalytics'] == 1) {
        $this->view->cookiesSettingsAllowStatistics = 1;
    } else {

```

```

} else {
    $this->view->cookiesSettingsAcceptedEverything = false;
}

// LiveChat (requires Google statistics too)
if (
    isset($this->view->cookiesSettings['preferencesLiveChat']) AND
    $this->view->cookiesSettings['preferencesLiveChat'] == 1 AND
    isset($this->view->cookiesSettings['statisticsGoogleAnalytics']) AND
    $this->view->cookiesSettings['statisticsGoogleAnalytics'] == 1
) {
    $this->view->cookiesSettingsAllowLiveChat = 1;
} else {
    $this->view->cookiesSettingsAcceptedEverything = false;
}

// Display Disclaimer?
if (!isset($this->view->cookiesSettings['displayDisclaimer']) OR $this->view->cookiesSettings['displayDisclaimer'] == 1) {
    $this->view->cookiesDisplayDisclaimer = 1;
}

// REMOVE THIS LATER
// $this->view->gdprTesting = \in_array($_SERVER['REMOTE_ADDR'], ['94.155.221.220', '178.169.252.245', '178.169.255.197', '85.196.180.242']);
// $this->view->gdprTesting = false; // Going LIVE for now. Remove checks from templates later!
}

// Flight search form - Autocomplete airports
public function airportAutocompleteAction()
{
    $result = \Helpers\Airport::search($this->request->getPost('title'), $this->session->language, true);
    $this->response->setContentType('application/json', 'UTF-8');
    $this->response->setJsonContent($result, JSON_UNESCAPED_UNICODE);
    return $this->response->send();
}

// Flight search form - Process submit
public function processFlightSearchFormAction()
{
    $params = [
        'fs' => 1,
        'type' => 'O',
        'adults' => $this->request->getPost('adults', 'int', 1),
        'children' => $this->request->getPost('children', 'int', 0),
    ];

    // From
    $parsed = \Helpers\Airport::extractParams($this->request->getPost('airportFromClean', 'string'));
    $params['from'] = $parsed['type'] . $parsed['airportCode'];

    // To
    $parsed = \Helpers\Airport::extractParams($this->request->getPost('airportToClean', 'string'));
    $params['to'] = $parsed['type'] . $parsed['airportCode'];

    // Date
    $params['departure'] = date('Y-m-d', strtotime($this->request->getPost('departureDate', 'string')));
    if (empty($this->request->getPost('returnDate'))) {
        $params['return'] = date('Y-m-d', strtotime($this->request->getPost('returnDate', 'string')));
        $params['type'] = 'R';
    }

    // Multiple destinations?
    if (
        $this->request->getPost('searchType') == 'advanced' AND
        empty($this->request->getPost('airportFromTwoClean')) AND
        empty($this->request->getPost('airportToTwoClean'))
    ) {
        unset($params['return']);
        $params['type'] = 'M';

        // From
        $parsed = \Helpers\Airport::extractParams($this->request->getPost('airportFromTwoClean', 'string'));
        $params['from1'] = 'C'. $parsed['airportCode']; // Always search by city

        // To
        $parsed = \Helpers\Airport::extractParams($this->request->getPost('airportToTwoClean', 'string'));
        $params['to1'] = 'C'. $parsed['airportCode']; // Always search by city

        // Date
        $params['departure1'] = date('Y-m-d', strtotime($this->request->getPost('departureDateTwo', 'string')));
    }

    // Redirect to Amadeus
    $url = 'https://flights.usitcolours.bg/?';
    $url.= http_build_query($params);
    return $this->response->redirect($url, true);
}

// Subscribe popup
public function subscribePopupAction()
{
    $post = $this->request->getJsonRawBody();
    $this->view->setRenderLevel(\Phalcon\Mvc\View::LEVEL_NO_RENDER);
}

```

```

$viewParams = [];
$html = $this->view->getRender('_layouts/parts', 'subscribe-popup', $viewParams, function($view) {
    $view->setRenderLevel(\Phalcon\Mvc\View::LEVEL_ACTION_VIEW);
});

// Response
if ($this->request->isAjax()) {
    return $this->response->setJsonContent(['html' => $html]);
} else {
    return $this->response->setContent($html);
}
}

// 404
public function error404Action()
{
    // Forbid indexing this page!
    $this->view->searchIndexStatus = 'noindex';

    $this->response->setStatusCode(404, 'Not Found');
    $this->view->pick('_layouts/error-404');
    $this->response->send();
}

/**
 * Returns md5 hash for the current page.
 *
 * @param string $token
 * @return string
 */
public function getCurrentShareToken($token)
{
    return md5($token);
}

public function getUsitOffices($visibility = 'all')
{
    return \Helpers\Common::getUsitOffices($this->session->language, $visibility);
}

// Update share counters
public function updateShareCounterAction()
{
    $result = [];
    $token = $this->request->getPost('token');
    $action = $this->request->getPost('action');
    $shared = $this->session->shared;

    if (!$shared OR !isset($shared[$action][$token])) {
        // Check for sneaky users

        if (in_array($action, ['facebook', 'twitter', 'google', 'print', 'email', 'viber', 'whatsapp', 'fbMessenger'])) {
            return false;
        }

        $check = \Models\ShareCounters::findFirstByToken($token);
        if ($check === false) {
            $check = new \Models\ShareCounters();
            $check->token = $token;
        }
        $check->{$action} = ($check->{$action} ?? 0) + 1;
        $check->save();

        // Update session
        $shared[$action][$token] = 1;
        $this->session->shared = $shared;

        $result = [
            'action' => $action,
            'counter' => $check->{$action}
        ];
    }
    $this->response->setJsonContent($result);
    return $this->response->send();
}

// Detect Language
private function detectLanguage()
{
    // Single language => use default locale
    $language = $this->config->site->defaultLanguage;

    // Multi language site
    if (count($this->config->site->languages) > 1 AND $this->dispatcher->getParam('language')) {
        if (in_array($this->dispatcher->getParam('language'), (array) $this->config->site->languages)) {
            // default language should not be in the url. Redirect in case someone wrote it manually
            if ($this->dispatcher->getParam('language') == $language) {
                $needle = '/' . $language . '/'; // Example: /bg/
                $currentUrl = getCurrentUrl();
                // Inner page (/bg/...) or Home (/bg)
                if (strpos($currentUrl, $needle) !== false) {
                    $redirect = str_replace($needle, '/', $currentUrl);
                }
            }
        }
    }
}

```

```

        } else {
            $redirect = str_replace('/', $language, '', $currentUrl);
        }
        return $this->response->redirect($redirect, true, 301)->send();
    } else {
        $language = $this->dispatcher->getParam('language');
    }
} else {
    // Locale is in URL but its not a valid config value. Redirect to home
    return $this->response->redirect($this->config->site->url, true, 301)->send();
}
}
$this->session->language = $language;
}

/**
 * Add additional scripts to header/footer templates
 *
 * @param string $type Available values: css|js|script
 * @param string || array $fileNames Filenames to be included. File extension is added automatically
 * @return void
 */
public function additionalAssets($type, $fileNames)
{
    foreach ((array) $fileNames as $name) {
        $this->additionalAssets[$type][] = $name;
    }
    $this->view->additionalAssets = $this->additionalAssets;
}

/**
 * Check if user is logged.
 * Redirect to login if not
 *
 * @return void
 */
/* public function checkLogged()
{
    $exceptions = [
        'profile/login',

        'profile/logout',
    ];
    $needle = $this->router->getControllerName() . '/' . $this->router->getActionName();
    if (!$this->session->has('profile') AND !in_array($needle, $exceptions)) {
        $this->flash->error($this->translations->public->general->loggedOnly);
        return $this->response->redirect(
            url(['for' => 'default', 'controller' => 'profile', 'action' => 'login'], ['return' => $this->request->getUri()]),
            true
        )->send();
    }
}
*/

public function loadLanguageFile($key, $file)
{
    $obj = new \Models\Translations();
    $translations = $obj->getCache($file);
    if ($this->di->has('translations')) {
        $temp = $this->translations;
    } else {
        $temp = new \stdClass();
    }
    $temp->{$key} = $translations;
    $this->di->setShared('translations', $temp);
}
}

```

#3 Frontend\Controllers\BaseController->initialize()

#4 [Phalcon\Dispatcher](#)->dispatch()

#5 [Phalcon\Mvc\Application](#)->handle()

/home/dev.crusit.bg/public_html/apps/Bootstrap.php (182)

<?php

```
require_once __DIR__ . '/../common/lib/global-functions.php';
```

```
class Bootstrap
```

```
{
```

```
    private $app;
```

```
    public function __construct()
```

```
    {
```

```
        // Dependency Injector
```

```
        $di = new \Phalcon\DI\FactoryDefault();
```

```
        // Configuration
```

```
        $config = loadConfigFile('config');
```

```
        $databaseCredentials = $config->sensitive->database;
```

```
        unset($config->sensitive);
```

```
        $di->setShared('config', $config);
```

```
        // 301 Redirects from CMS - must be before routes
```

```
        $currentUrl = getCurrentUrl();
```

```

$redirectsCacheFile = $config->site->path->cache . 'site/redirects';
if ($config->cms->functionalities->redirects === true AND is_file($redirectsCacheFile)) {
    $redirects = json_decode(
        file_get_contents($redirectsCacheFile),
        true
    );
    $url = str_replace($config->site->url, "", $currentUrl);
    $url = strtok(rawurldecode($url), "?");
    if (isset($redirects[$url])) {
        return $di->getResponse()->redirect($redirects[$url]['url'], true, $redirects[$url]['code']->send());
    }
}

// 301 Redirect if /public/index.php
if (strpos($currentUrl, 'public/index.php') !== false) {
    return $di->getResponse()->redirect($config->site->url, true, 301)->send();
}

// 301 Redirect if /public/ - Replace only the first occurrence of /public/
$currentPath = str_replace($config->site->url, "", $currentUrl);
if (substr($currentPath, 0, 6) === 'public') {
    $needle = '/public/';
    $redirect = substr_replace($currentUrl, '/', strpos($currentUrl, $needle), strlen($needle));
    return $di->getResponse()->redirect($redirect, true, 301)->send();
}

// Router
$di->setShared('router', function() use ($config) {
    $router = new \Phalcon\Mvc\Router();
    $router->setDefaultModule('frontend');

    $router->removeExtraSlashes(true);
    require_once $config->site->path->configs . 'routes.php';
    // loadConfigFile('routes', true); // Why this was here? Am I getting too old?
    return $router;
});

// Crypt
$di->setShared('crypt', function () use ($config) {
    $crypt = new \Phalcon\Crypt();
    $crypt->setKey('$4^&/.%2@k%0ROQ<@{(e=*!<2u|rl~4');
    return $crypt;
});

// URL component
$di->setShared('url', function() use ($config) {
    $url = new \Phalcon\Mvc\Url();
    $url->setBaseUri($config->site->url);
    $url->setStaticBaseUri($config->site->staticUrl);
    return $url;
});

// Session
$di->setShared('session', function() {
    $session = new \Phalcon\Session\Adapter\Files();
    $session->start();
    return $session;
});

// Cache
$di->setShared('cache', function() use ($config) {
    return new \Phalcon\Cache\Backend\File(
        new \Phalcon\Cache\Frontend\Json(['lifetime' => 172800]), // 2d
        ['cacheDir' => $config->site->path->cache . 'site/']
    );
});

// Database connection. Use set() if Transactions are needed
$di->setShared('db', function() use ($databaseCredentials) {
    return new \Phalcon\Db\Adapter\Pdo\Mysql([
        'host' => $databaseCredentials->host,
        'username' => $databaseCredentials->username,
        'password' => $databaseCredentials->password,
        'dbname' => $databaseCredentials->name,
        'charset' => 'utf8',
        'options' => [
            // attr syntax "SET query1, query2, query 3...." (Only 1 set and rest are comma separated).
            // Disable latest mysql ONLY_FULL_GROUP_BY mode.
            // \PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES utf8, GLOBAL sql_mode=(SELECT REPLACE(@@sql_mode,'ONLY_FULL_GROUP_BY',''))";
            // \PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES utf8, CHARACTER SET utf8";
            \PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES utf8";
            \PDO::ATTR_PERSISTENT => true,
            \PDO::ATTR_STRINGIFY_FETCHES => false
        ]
    ]);
});

// Cache Model meta data only in production
if ($config->debug === false) {
    $di->setShared('modelsMetadata', function() use ($config) {
        return new \Phalcon\Mvc\Model\MetaData\Files([
            'lifetime' => 86400,

```

```

        'metaDataDir' => $config->site->path->cache . 'models-metadata/'
    });
});
}

// Models cache
$di->setShared('modelsCache', function() use ($config) {
    $frontCache = new \Phalcon\Cache\Frontend\Data(['lifetime' => 7200]); // 2h
    $cache = new \Phalcon\Cache\Backend\File($frontCache, [
        'cacheDir' => $config->site->path->cache . 'queries/'
    ]);
    if ($config->debug) {
        $cache->flush();
    }
    return $cache;
});

$this->app = new \Phalcon\Mvc\Application();
$this->app->setDI($di);
}

public function run()
{
    $modules = [
        'frontend' => [
            'className' => 'Frontend\Module',
            'path' => __DIR__ . '/frontend/Module.php'
        ],
        'backend' => [
            'className' => 'Backend\Module',
            'path' => __DIR__ . '/backend/Module.php'
        ]
    ];

    // Is there an API?
    $uri = $this->app->router->getRewriteUri();
    if ($this->app->config->dev->hasApi === true) {
        if (substr($uri, 0, 5) == '/api/') {
            // Extract api version from url: /api/v1/robots...
            $parts = array_filter(explode('/', $uri));

            // Save API version
            $this->app->config->apiVersion = $parts[2];

            // Add to modules
            $modules['api'] = [
                'className' => 'Api\Module',
                'path' => __DIR__ . '/api/'. $this->app->config->apiVersion . '/Module.php'
            ];
        }
    }

    // Whitelabel app?
    if (substr($uri, 0, 12) == '/whitelabel/') {
        // Extract version from url: /whitelabel/v1/robots...
        $parts = array_filter(explode('/', $uri));

        // Save version
        $this->app->config->whitelabelVersion = $parts[2];

        // Add to modules
        $modules['whitelabel'] = [
            'className' => 'Whitelabel\Module',
            'path' => __DIR__ . '/whitelabel/'. $this->app->config->whitelabelVersion . '/Module.php'
        ];
    }

    // Register application modules
    $this->app->registerModules($modules);
    echo $this->app->handle()->getContent();
}

public function getApp()
{
    return $this->app;
}
}

```

#6 Bootstrap->run()
/home/dev.crusit.bg/public_html/public/index.php (7)

```
<?php
try {
    define('DEV', in_array($_SERVER['REMOTE_ADDR'], ['77.70.108.105', '::1']));
    $root = str_replace('\\', '/', dirname(dirname(__FILE__))) . '/';
    require_once $root . 'apps/Bootstrap.php';
    $bootstrap = new Bootstrap();
    $bootstrap->run();
} catch (\Exception $e) {
    if ($bootstrap->getApp()->config->debug === true) {
        $debug = new \Phalcon\Debug();
        die($debug->listen()->onUncaughtException($e));
    } else {
        // Log the error
        // $logFile = $bootstrap->getApp()->config->site->path->logs . 'errors/error-500_' . date("YmdHis") . '_' . mt_rand(1, 1000000) . '.txt';
        // file_put_contents($logFile, var_export($e, true));
        error_log($e);

        // Output for user
        header("HTTP/1.1 500 Internal Server Error");
        die(file_get_contents('../apps/frontend/views/_layouts/error-500.phtml'));
    }
}
```

| Key | Value |
|-----------------------|--|
| _url | /cruises/viewpdf |
| slug | 7-dni-yuzhni-karibi-ot-puerto-riko-prma07cnnsjusju |
| period | 0 |
| loadAutoPrefixedStyle | |

| Key | Value |
|----------------------|--|
| USER | dev.crusit.bg |
| HOME | /home/dev.crusit.bg |
| HTTP_HOST | dev.crusit.bg |
| HTTP_ACCEPT_LANGUAGE | en,* |
| HTTP_ACCEPT_ENCODING | gzip |
| HTTP_CONNECTION | Keep-Alive |
| HTTP_ACCEPT | text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 |
| HTTP_USER_AGENT | Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/534.34 (KHTML, like Gecko) wkhtmltopdf Safari/534.34 |
| HTTPS | on |
| PATH_INFO | |
| SERVER_NAME | dev.crusit.bg |
| SERVER_PORT | 443 |
| SERVER_ADDR | 185.55.229.202 |
| REMOTE_PORT | 55930 |
| REMOTE_ADDR | 185.55.229.202 |
| SERVER_PROTOCOL | HTTP/1.1 |
| DOCUMENT_ROOT | /home/dev.crusit.bg/public_html/public |
| DOCUMENT_URI | /index.php |
| REQUEST_URI | /cruises/viewpdf?slug=7-dni-yuzhni-karibi-ot-puerto-riko-prma07cnnsjusju&period=0&loadAutoPrefixedStyle |
| SCRIPT_NAME | /index.php |
| SCRIPT_FILENAME | /home/dev.crusit.bg/public_html/public/index.php |
| CONTENT_LENGTH | |
| CONTENT_TYPE | |
| REQUEST_METHOD | GET |
| QUERY_STRING | _url=/cruises/viewpdf&slug=7-dni-yuzhni-karibi-ot-puerto-riko-prma07cnnsjusju&period=0&loadAutoPrefixedStyle |
| SERVER_SOFTWARE | nginx |
| GATEWAY_INTERFACE | CGI/1.1 |
| FCGI_ROLE | RESPONDER |
| PHP_SELF | /index.php |
| REQUEST_TIME_FLOAT | 1787654765.8096 |
| REQUEST_TIME | 1787654765 |

| # | Path |
|----|--|
| 0 | /home/dev.crusit.bg/public_html/public/index.php |
| 1 | /home/dev.crusit.bg/public_html/apps/Bootstrap.php |
| 2 | /home/dev.crusit.bg/public_html/common/lib/global-functions.php |
| 3 | /home/dev.crusit.bg/public_html/common/config/config.php |
| 4 | /home/dev.crusit.bg/public_html/common/config/routes.php |
| 5 | /home/dev.crusit.bg/public_html/common/config/routes-api.php |
| 6 | /home/dev.crusit.bg/public_html/common/config/routes-whitelabel.php |
| 7 | /home/dev.crusit.bg/public_html/apps/frontend/Module.php |
| 8 | /home/dev.crusit.bg/public_html/common/lib/Fabfuel/Prophiler/Profiler.php |
| 9 | /home/dev.crusit.bg/public_html/common/lib/Fabfuel/Prophiler/ProfilerInterface.php |
| 10 | /home/dev.crusit.bg/public_html/common/lib/Fabfuel/Prophiler/Aggregator/Database/QueryAggregator.php |

11 /home/dev.crusit.bg/public_html/common/lib/Fabfuel/Prophiler/Aggregator/AbstractAggregator.php
12 /home/dev.crusit.bg/public_html/common/lib/Fabfuel/Prophiler/AggregatorInterface.php
13 /home/dev.crusit.bg/public_html/common/lib/Fabfuel/Prophiler/Aggregator/Cache/CacheAggregator.php
14 /home/dev.crusit.bg/public_html/common/lib/Fabfuel/Prophiler/Plugin/Manager/Phalcon.php
15 /home/dev.crusit.bg/public_html/common/lib/Fabfuel/Prophiler/Plugin/Phalcon/Db/AdapterPlugin.php
16 /home/dev.crusit.bg/public_html/common/lib/Fabfuel/Prophiler/Plugin/PluginAbstract.php
17 /home/dev.crusit.bg/public_html/common/lib/Fabfuel/Prophiler/Plugin/Phalcon/Mvc/DispatcherPlugin.php
18 /home/dev.crusit.bg/public_html/common/lib/Fabfuel/Prophiler/Plugin/Phalcon/Mvc/ViewPlugin.php
19 /home/dev.crusit.bg/public_html/common/lib/Fabfuel/Prophiler/Plugin/Phalcon/Mvc/ViewPluginInterface.php
20 /home/dev.crusit.bg/public_html/common/lib/Fabfuel/Prophiler/Toolbar.php
21 /home/dev.crusit.bg/public_html/common/lib/Fabfuel/Prophiler/Adapter/Psr/Log/Logger.php

22 /home/dev.crusit.bg/public_html/common/lib/Fabfuel/Prophiler/Adapter/AdapterAbstract.php
23 /home/dev.crusit.bg/public_html/common/lib/Fabfuel/Prophiler/DataCollector/Request.php
24 /home/dev.crusit.bg/public_html/common/lib/Fabfuel/Prophiler/DataCollectorInterface.php
25 /home/dev.crusit.bg/public_html/common/lib/Fabfuel/Prophiler/Benchmark/BenchmarkFactory.php
26 /home/dev.crusit.bg/public_html/common/lib/Fabfuel/Prophiler/Benchmark/Benchmark.php
27 /home/dev.crusit.bg/public_html/common/lib/Fabfuel/Prophiler/Benchmark/BenchmarkInterface.php
28 /home/dev.crusit.bg/public_html/apps/frontend/controllers/CruisesController.php
29 /home/dev.crusit.bg/public_html/apps/frontend/controllers/BaseController.php
30 /home/dev.crusit.bg/public_html/common/models/Translations.php
31 /home/dev.crusit.bg/public_html/common/models/BaseModel.php
32 /home/dev.crusit.bg/public_html/common/models/Pages.php
33 /home/dev.crusit.bg/public_html/common/models/Pages18n.php
34 /home/dev.crusit.bg/public_html/common/models/Uploads.php
35 /home/dev.crusit.bg/public_html/common/models/Uploads18n.php
36 /home/dev.crusit.bg/public_html/common/lib/Fabfuel/Prophiler/Aggregator/Aggregation.php
37 /home/dev.crusit.bg/public_html/common/lib/Fabfuel/Prophiler/Aggregator/AggregationInterface.php
38 /home/dev.crusit.bg/public_html/common/helpers/Cruises.php
39 /home/dev.crusit.bg/public_html/common/models/CruisePricesCache.php
40 /home/dev.crusit.bg/public_html/common/helpers/CookiesConsent.php
41 /home/dev.crusit.bg/public_html/common/helpers/Common.php
42 /home/dev.crusit.bg/public_html/common/models/Settings.php

Memory

Usage 4194304