

```
<?php
```

```
/*
```

```
* This file is part of EC-CUBE
```

```
*
```

```
* Copyright(c) EC-CUBE CO.,LTD. All Rights Reserved.
```

```
*
```

```
* http://www.ec-cube.co.jp/
```

```
*
```

```
* For the full copyright and license information, please view the LICENSE
```

```
* file that was distributed with this source code.
```

```
*/
```

```
namespace Plugin\HogePlugin\Service;
```

```
use Eccube\Common\EccubeConfig;
```

```
use Eccube\Entity\BaseInfo;
```

```
use Eccube\Event\EccubeEvents;
```

```
use Eccube\Event\EventArgs;
```

```
use Eccube\Repository\BaseInfoRepository;
```

```
use Eccube\Repository\MailTemplateRepository;
```

```
use Symfony\Component\EventDispatcher\EventDispatcher;
```

```
use Symfony\Component\EventDispatcher\EventDispatcherInterface;
```

```
use Plugin\HogePlugin\Repository\ConfigRepository;
```

```
class MailService
```

```
{
```

```
/**
```

```
 * @var \Swift_Mailer
```

```
*/
```

```
protected $mailer;
```

```
/**
```

```
 * @var MailTemplateRepository
```

```
*/
```

```
protected $mailTemplateRepository;
```

```
/**
```

```
 * @var EventDispatcher
```

```
*/
```

```
protected $eventDispatcher;
```

```
/**
```

```

* @var BaselInfo
*/
protected $BaselInfo;

/**
* @var EccubeConfig
*/
protected $eccubeConfig;

/**
* @var \Twig_Environment
*/
protected $twig;

/**
* @var ConfigRepository
*/
protected $configRepository;

/**
* MailService constructor.
*
* @param \Swift_Mailer $mailer
* @param MailTemplateRepository $mailTemplateRepository
* @param BaselInfoRepository $baselInfoRepository
* @param EventDispatcherInterface $eventDispatcher
* @param \Twig_Environment $twig
* @param EccubeConfig $eccubeConfig
* @param ConfigRepository $configRepository
*/
public function __construct(
    \Swift_Mailer $mailer,
    MailTemplateRepository $mailTemplateRepository,
    BaselInfoRepository $baselInfoRepository,
    EventDispatcherInterface $eventDispatcher,
    \Twig_Environment $twig,
    EccubeConfig $eccubeConfig,
    ConfigRepository $configRepository
) {
    $this->mailer = $mailer;
    $this->mailTemplateRepository = $mailTemplateRepository;
    $this->BaselInfo = $baselInfoRepository->get();
    $this->eventDispatcher = $eventDispatcher;
}

```

```

$this->eccubeConfig = $eccubeConfig;
$this->twig = $twig;
$this->configRepository = $configRepository;
}

/**
 * Send contact mail.
 *
 * @param $formData お問い合わせ内容
 */
public function sendContactMail($formData)
{
    log_info('お問い合わせ受付メール送信開始');

    $config = $this->configRepository->findOneBy(["kensaku" => 'search_mail_template']);

    $mail_template_id = $config->getMailTemplateId();

    $MailTemplate = $this->mailTemplateRepository->find($mail_template_id);

    $body = $this->twig->render($MailTemplate->getFileName(), [
        'data' => $formData,
        'BaseInfo' => $this->BaseInfo,
    ]);

    // 問い合わせ者にメール送信
    $message = (new \Swift_Message())
        ->setSubject(['.$this->BaseInfo->getShopName().'] '.$MailTemplate->getMailSubject())
        ->setFrom([$this->BaseInfo->getEmail02() => $this->BaseInfo->getShopName()])
        ->setTo([$formData['email']])
        ->setBcc($this->BaseInfo->getEmail02())
        ->setReplyTo($this->BaseInfo->getEmail02())
        ->setReturnPath($this->BaseInfo->getEmail04());

    // HTMLテンプレートが存在する場合
    $htmlFileName = $this->getHtmlTemplate($MailTemplate->getFileName());
    if (!is_null($htmlFileName)) {
        $htmlBody = $this->twig->render($htmlFileName, [
            'data' => $formData,
            'BaseInfo' => $this->BaseInfo,
        ]);
    }

    $message

```

```

        ->setContentType('text/plain; charset=UTF-8')
        ->setBody($body, 'text/plain')
        ->addPart($htmlBody, 'text/html');
    } else {
        $message->setBody($body);
    }

    $event = new EventArgs(
        [
            'message' => $message,
            'formData' => $formData,
            'BaseInfo' => $this->BaseInfo,
        ],
        null
    );
    $this->eventDispatcher->dispatch(EccubeEvents::MAIL_CONTACT, $event);

    $count = $this->mailer->send($message);

    log_info('お問い合わせ受付メール送信完了', ['count' => $count]);

    return $count;
}

/**
 * [getHtmlTemplate description]
 *
 * @param string $templateName プレーンテキストメールのファイル名
 *
 * @return string|null 存在する場合はファイル名を返す
 */
public function getHtmlTemplate($templateName)
{
    // メールテンプレート名からHTMLメール用テンプレート名を生成
    $fileName = explode('.', $templateName);
    $suffix = '.html';
    $htmlFileName = $fileName[0].$suffix.'.'.$fileName[1];

    // HTMLメール用テンプレートの存在チェック
    if ($this->twig->getLoader()->exists($htmlFileName)) {
        return $htmlFileName;
    } else {
        return null;
    }
}

```

}  
}  
}