

之前课程把Handler作为FileUploadSubscriber的构造参数注入到了服务类里。上节课, 我们把不同文件Entity类对应的Handler类id作为参数添加到了容器。在构造参数里, 我们需要2个参数, 一个ParameterBagInterface类型的对象, 用于获取teebb.upload.handlers参数。另一个使用ContainerInterface对象, 用于获取容器中对应的Handler类对象。

重构后的代码:

```
class FileUploadSubscriber implements EventSubscriberInterface
{
    /**
     * @var UploadHandler
     */
    private $uploadHandler;

    /**
     * public function __construct(UploadHandler $uploadHandler)
     */
    {
        $this->uploadHandler = $uploadHandler;
    }

    /**
     * @var ParameterBagInterface
     */
    private $parameterBag;

    /**
     * @var ContainerInterface
     */
    private $container;

    /**
     * @var array
     */
    private $handlers;

    public function __construct(ParameterBagInterface $parameterBag,
        ContainerInterface $container)
    {
        $this->parameterBag = $parameterBag;
        $this->container = $container;
        $this->handlers = $parameterBag->get('teebb.upload.handlers');
    }

    public function getSubscribedEvents(): array
    {
        return [
            Events::prePersist,
            Events::preUpdate,
            Events::postPersist,
            Events::postUpdate,
        ];
    }
}
```

```
public function prePersist(LifecycleEventArgs $args): void
{
    $object = $args->getEntity();
    $this->setFileObjectName($object);
}

public function preUpdate(PreUpdateEventArgs $args): void
{
    $object = $args->getEntity();
    $this->setFileObjectName($object);
}

public function postPersist(LifecycleEventArgs $args): void
{
    $object = $args->getEntity();
    $this->uploadFile($object);
}

public function postUpdate(LifecycleEventArgs $args): void
{
    $object = $args->getEntity();
    $this->uploadFile($object);
}

private function setFileObjectName($object)
{
    $className = get_class($object);
    if (key_exists($className, $this->handlers))
    {
        $handler = $this->container->get($this->handlers[$className]);

        if ($uploadedFile = $object->getUploadedFile()) {
            $fileName = $handler->getFileName($uploadedFile);
            $object->setName($fileName);
        }
    }

    // if ($object instanceof SimpleFile) {
    //     //这里要添加判断
    //     if ($uploadedFile = $object->getUploadedFile()) {
    //         $fileName = $this->uploadHandler-
    >getFileName($uploadedFile);
    //         $object->setName($fileName);
    //     }
    // }

    private function uploadFile($object)
    {
        $className = get_class($object);
        if (key_exists($className, $this->handlers))
        {
            $handler = $this->container->get($this->handlers[$className]);

            if ($uploadedFile = $object->getUploadedFile()) {
```

```
        $handler->upload($uploadedFile, $object->getName());
    }
}

//      if ($object instanceof SimpleFile) {
//          //这里要添加判断
//          if ($uploadedFile = $object->getUploadedFile()) {
//              $this->uploadHandler->upload($uploadedFile, $object-
>getName());
//          }
//      }
}

}
```