在Namer目录中新增一个HashNamer，根据上传文件的哈稀值来重命名文件：（这里快速过一遍代码）

```php
//这里新定义一个ConfigurableInterface接口，用于给Namer类添加设置
class HashNamer implements NamerInterface, ConfigurableInterface
{
    /** @var string */
    private $algorithm = 'sha1';

    /** @var int */
    private $length;

    /**
     * @param array $options Options for this namer. The following options are accepted:
     *                        - algorithm: wich hash algorithm to use.
     *                        - length: limit file name length
     */
    public function configure(array $options): void
    {
        $options = \array_merge(['algorithm' => $this->algorithm, 'length' => $this->length], $options);

        $this->algorithm = $options['algorithm'];
        $this->length = $options['length'];
    }

    public function rename(UploadedFile $file): string
    {
        $name = \hash($this->algorithm, $this->getRandomString());
        if (null !== $this->length) {
            $name = \substr($name, 0, $this->length);
        }

        if ($extension = $file->getClientOriginalExtension()) {
            $name = \sprintf('%s.%s', $name, $extension);
        }

        return $name;
    }

    protected function getRandomString(): string
    {
        return \microtime(true).\random_int(0, 9999999);
    }

}
```

修改Extension类，改用xml把Namer、Storage服务类添加到容器里。（这里也快速修改一下，不水时长）

```
<!-- namer.xml -->
<?xml version="1.0" encoding="utf-8" ?>
<container xmlns="http://symfony.com/schema/dic/services"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
           xsi:schemaLocation="http://symfony.com/schema/dic/services
http://symfony.com/schema/dic/services/services-1.0.xsd">
    <services>
        <service id="teebb.upload.namer.php_namer"
class="Teebb\UploadBundle\Namer\PhpNamer" public="true"/>
        <service id="teebb.upload.namer.hash_namer"
class="Teebb\UploadBundle\Namer\HashNamer" public="true"/>
    </services>
</container>

<!-- storage.xml -->
<?xml version="1.0" encoding="utf-8" ?>
<container xmlns="http://symfony.com/schema/dic/services"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
           xsi:schemaLocation="http://symfony.com/schema/dic/services
http://symfony.com/schema/dic/services/services-1.0.xsd">
    <services>
        <service id="teebb.upload.storage.file_system_storage"
class="\Teebb\UploadBundle\Storage\FileSystemStorage" public="true"/>
    </services>
</container>
```

```
//修改Extension类加载配置文件，修改`Handler`类的构造参数
  public function load(array $configs, ContainerBuilder $container)
    {
        $processor = new Processor();
        $configuration = new Configuration();
        $config = $processor->processConfiguration($configuration,
$configs);

        $loader = new XmlFileLoader($container, new
FileLocator(__DIR__.'/../Resources/config'));

// 这里修改
        $resources = [
            'form',
            'namer',
            'storage',
        ];

        foreach ($resources as $resource) {
            $loader->load($resource . '.xml');
        }

        $container->setParameter('teebb.upload_dir',
$config['upload_dir']);
```

```
        $fileManagedTypeDefinition = $container-
>getDefinition('teebb.upload.form.file_managed_type');
        $fileManagedTypeDefinition->setArgument(0, $config['upload_dir']);

        //改用xml文件
//        $this->registerStorages($container);
//        $this->registerNamers($container);

        //最后把Handler添加到容器，Handler有构造参数，所以先定义再添加到容器
        $handlerDefinition = new Definition(UploadHandler::class);
        $handlerDefinition->setArgument(0, new
Reference('teebb.upload.storage.file_system_storage'));
        $handlerDefinition->setArgument(1, $config['upload_dir']);
        $handlerDefinition->setArgument(2, new
Reference('teebb.upload.namer.php_namer'));

        $container->setDefinition(UploadHandler::class,
$handlerDefinition);

    }
```

代码修改后，我们看到Handler服务类的参数写死了，我们可以为Bundle添加配置项，然后动态的设置Handler的构造参数。