在表单设置数据之前定义一个事件

```
    public function buildForm(FormBuilderInterface $builder, array
$options)
    {
        $fileClass = $options['file_class'];
//        $builder->addEventListener(FormEvents::SUBMIT,[$this,
'onFormSubmit']);

        $builder->addEventListener(FormEvents::SUBMIT, function (FormEvent
$event) use ($fileClass) {
            /**@var UploadedFile $data * */
            $file = $event->getData();
            $originName = $file->getClientOriginalName();
            $fileName = pathinfo(htmlspecialchars($originName),
PATHINFO_FILENAME) . '-' . $file->getFilename() . '.' . $file-
>getClientOriginalExtension();
//        $uploadPath = $this->parameterBag-
>get('teebb.upload_dir');//$this->getParameter('base_path');

            $mimeType = $file->getMimeType();
            $filesize = $file->getSize();

//        $file->move($uploadPath, $fileName);
            $file->move($this->uploadDir, $fileName);

            $uploadedFile = new $fileClass();
            $uploadedFile->setName($fileName);
            $uploadedFile->setMimeType($mimeType);
            $uploadedFile->setOriginName($originName);
            $uploadedFile->setSize($filesize);

            // 创建事件
            $fileUploadEvent = new AfterFileUploadEvent($file,
$uploadedFile);
            $this->eventDispatcher->dispatch($fileUploadEvent);

            $event->setData($fileUploadEvent->getFileObject());
        });

    }
```

```
class AfterFileUploadEvent extends Event
{
    /**
     * @var UploadedFile
     */
    private $uploadedFile;
    /**
     * @var File
     */
    private $fileObject;

    public function __construct(UploadedFile $uploadedFile, File
$fileObject)
    {
        $this->uploadedFile = $uploadedFile;
        $this->fileObject = $fileObject;
    }

    /**
     * @return UploadedFile
     */
    public function getUploadedFile(): UploadedFile
    {
        return $this->uploadedFile;
    }

    /**
     * @return File
     */
    public function getFileObject(): File
    {
        return $this->fileObject;
    }

}
```

在`FileManagedType`中我们注入了`EventDispatcherInterface`对象，需要修改xml配置文件，添加构造参数。可以在Teebblog项目中为xml添加参数

再到项目中为SimpleFile添加`path`属性，并且执行数据库的修改。创建事件订阅器，扩展path属性。

```
//teebblog SimpleFileSubscriber

class SimpleFileSubscriber implements EventSubscriberInterface
{
    public function onAfterFileUploadEvent(AfterFileUploadEvent $event)
    {
        /**@var SimpleFile $fileObject**/
        $fileObject = $event->getFileObject();
        $fileObject->setPath('teebb_upload' . DIRECTORY_SEPARATOR .
$fileObject->getName());
    }

    public static function getSubscribedEvents()
    {
        return [
            AfterFileUploadEvent::class => 'onAfterFileUploadEvent',
        ];
    }
}
```