修改Teebblog项目的Comment类，

Comment类的files属性现在对应的Entity类修改为SimpleFile类

```
    /**
     * @ORM\ManyToMany(targetEntity=SimpleFile::class, cascade=
{"persist"})
     * @ORM\JoinTable(name="comments_files",
     *      joinColumns={@ORM\JoinColumn(name="comment_id",
referencedColumnName="id")},
     *      inverseJoinColumns={@ORM\JoinColumn(name="file_id",
referencedColumnName="id", unique=true)}
     *      )
     *
     */
    private $files;
```

我们需要把文件上传表单提交的数据转换为SimpleFile对象。在`FileManagedType`类中，修改代码，文件移动成功后创建Entity类对象并赋值:

```
    public function onFormSubmit(FormEvent $event){

        /**@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 SimpleFile();
        $uploadedFile->setName($fileName);
        $uploadedFile->setMimeType($mimeType);
        $uploadedFile->setOriginName($originName);
        $uploadedFile->setSize($filesize);

        $event->setData($uploadedFile);

//        $fileManaged = new FileManaged();
//          $fileManaged->setOriginName($originName);
//          $fileManaged->setFileName($fileName);
//          $fileManaged->setMimeType($mimeType);
//          $fileManaged->setPath($uploadPath . '/' . $fileName);
//          $fileManaged->setFileSize($filesize);
//
//          $event->setData($fileManaged);
    }
```

还要修改FileManaged验证器的设置，让文件验证可以通过。

```
class FileManagedValidator extends ConstraintValidator
{
    public function validate($value, Constraint $constraint)
    {
//          dd($value, $constraint);
        /* @var $constraint \App\Validator\FileManaged */

        if (null === $value || '' === $value) {
            return;
        }

        // 修改这里让文件验证生效
        if ($value instanceof File){
            $mimeType = $value->getMimeType();

            foreach ($constraint->mimeTypes as $allowMimeType){
                if ($allowMimeType === $mimeType){
                    return;
                }

                if ($discrete = strstr($allowMimeType, '/*', true)) {
                    if (strstr($mimeType, '/', true) === $discrete) {
                        return;
                    }
                }

            }
        }

        // TODO: implement the validation here
        $this->context->buildViolation($constraint->message)
            ->setParameter('{{ name }}', $value->getOriginName())
            ->addViolation();
    }
}
```

评论表单提交后，SimpleFile类中并没有path属性，会报错误。要解决这个问题有很多种方法，可以设置twig方法`asset()`添加packageName，

```
#config/package/assets.yaml

framework:
    assets:
        json_manifest_path:
'%kernel.project_dir%/public/build/manifest.json'

        packages:
            # Prototype
            simple_file_upload:
                base_path: 'teebb_upload'
```

```
{# macro.html.twig #}
<div>
    {% for file in comment.files %}
        <a href="{{ asset(file.name, 'simple_file_upload') }}" data-
lightbox="{{ comment.id }}">
            <img src="{{ asset(file.name,
'simple_file_upload')|imagine_filter('my_thumb') }}" alt="{{
file.originName }}"
                    class="img-thumbnail">
        </a>
    {% endfor %}
</div>
```

我们自己的文件验证器可以也加入到Bundle中，方便以后复用。下节课，我们把FileManaged验证器加入到Bundle中。