我已经完成了上节课剩余的代码，首先我定义了一个Transition类。在类中定义了一些常量，对应着工作流的转换。回到PostCrudController类，在`configureActions()`方法中我已经定义了剩下的三个按钮，并且将按钮添加到文章列表页，并且每个按钮对应的action方法我也已经定义好了。在每个按钮的action方法中，我们只处理文章状态的转换，不进行其他的逻辑处理。

```
#src/Utils/Transition.php

class Transition
{
    public const REVIEW_REQUEST = 'review_request';
    public const EDITOR_REVIEW = 'editor_review';
    public const CHECKER_CHECK = 'checker_check';
    public const PUBLISH = 'publish';
}
```

代码我进行了一些优化，首先我们定一个applyTransition()方法，这个方法用来对文章的状态进行修改，然后我们定义了一个获取页面路径的方法，通过这个方法我们来返回到文章列表页。

```
#PostCrudController中代码的更改，请看这里
https://github.com/teebbstudios/teebblog/commit/8ba36e88e0fe69e9b31092207cde92c7c846dc8d
```

回到浏览器刷新文章列表，我们看第一篇文章。第一篇文章当前是草稿状态，当我们点击`review_request`按钮之后，文章的状态会变成`wait_for_review`和`wait_for_check`。然后我们点击`checker_check`按钮，现在文章的状态就进行了修改。我们再次点击`editor_review`按钮，现在文章的状态又进行了修改，同时后面添加了`publish`按钮。我们点击`publish`按钮，现在文章的状态就变为已发布状态。

看下一篇文章，我们点击review_request按钮，当点击完这个按钮之后，我们来进行一下用户的切换。`_switch_user`，首先我们切换为editor用户，现在我们是editor用户，对于editor用户我们希望只显示`editor_review`按钮，对于checker用户我们只显示`checker_check`按钮，这里我们就需要添加权限了。

回到项目我们来修改按钮的方法，首先是$editorReviewAction，在displayIf方法中我们添加另外一个条件判断，输入`$this->isGranted()` 我们检查`ROLE_EDITOR`，只有`ROLE_EDITOR`角色的用户才会显示editor_review按钮。

同样的我们修改checker_check按钮，这里使用`$this->isGranted('ROLE_CHECKER');`，回到浏览器刷新。现在我们是editor用户，我们看到现在只显示editor_review按钮。再次切换用户`_switch_user=checker`，现在用户切换为checker用户，按钮也只显示checker_check按钮了。

回到项目，对应的我们需要按钮的action方法中也添加权限的判断。修改代码，在editorReviewAction()方法中，我们使用`$this->denyAccessUnlessGranted()`，这里输入`ROLE_EDITOR`。同样的修改checkerCheckAction()，这里修改为`ROLE_CHECKER`，这样我们的权限配置就完成了。

回到浏览器刷新，点击checker_check，文章的状态进行了修改，我们再次切换用户，点击editor_review，现在这一篇文章就可以进行发布了。点击publish，文章就变更为published状态。

回到项目，除了这种添加权限的方法外，我们还可以在工作流的配置文件中添加权限。打开控制台，我们输入`symfony console config:dump`，打开workflow.yaml配置文件。工作流可用的配置项在framework配置下，后面我添加参数framework，再添加workflows，我们查看一下当前的工作流，所有可用的配置。

我们定义自己的工作流名称，然后指定工作流的类型等等等等。在transitions配置下，我们可以配置`guard`选项，`guard`选项可以对当前的transition进行权限的配置。这里有段示例，我们可以使用一些表达式方法，对权限进行认证。修改workflow.yaml配置文件，在editor_review 配置下，我们添加`guard`，这里使用单引号添加`is_granted`表达式方法。这里我们输入`ROLE_EDITOR`，然后呢我们复制一下当前行，粘贴。这里修改为CHECKER。

```
#config/packages/workflow.yaml

framework:
    workflows:
        blog_publishing:
            type: 'workflow' # or 'state_machine'
            audit_trail:
                enabled: true
            marking_store:
                type: 'method'
                property: 'status'
            supports:
                - App\Entity\Post
            initial_marking: draft
            places:
                - draft
                - wait_for_review
                - wait_for_check
                - approved_by_editor
                - approved_by_checker
                - published
            transitions:
                review_request:
                    from: draft
                    to: [wait_for_review, wait_for_check]
                editor_review:
                    guard: 'is_granted(''ROLE_EDITOR'')'
                    from: wait_for_review
                    to: approved_by_editor
                checker_check:
                    guard: 'is_granted(''ROLE_CHECKER'')'
                    from: wait_for_check
                    to: approved_by_checker
                publish:
                    from: [approved_by_editor, approved_by_checker]
                    to: published
```

回到PostCrudController，我们注释权限的认证。回到浏览器刷新，出错了，这里提示我们需要安装表达式组件。复制命令行，打开控制台粘贴。表达式组件安装完成后，我们再次刷新浏览器，现在我们的用户是

editor用户。我们点击第三篇文章的review_request按钮。

现在第三篇文章就只显示了editor_review按钮，我们点击editor_review，再次切换用户，_switch_user=checker，现在checker用户也只显示checker_check按钮，我们点击按钮。现在第三篇文章就可以进入到发布状态了，这就是工作流中权限的使用。

我们已经学习了工作流，我们还可以定义一个状态机，在下节课我们来定义评论的工作流。