我们已经为管理端的用户定义了一些角色，现在我们为各个角色的用户分配一下工作。首先是编辑角色，编辑角色的用户可以发布文章，发布完之后需要对自己发布的文章进行一次检查。然后是ROLE_CHECKER角色，ROLE_CHECKER角色用于审核文章，对于文章的拼写错别字进行检查，当两个角色的用户工作完成之后。就可以将文章进行发布了。

ROLE_ADMIN用户拥有较高的权限，它可以完成上面两个角色的所有工作。为了使不同的角色完成不同的工作，我们可以定义工作流。

Symfony提供了工作流组件，在学习工作流之前，我们先来查看一个工作流的案例。回到浏览器，这个网站是Symfony官方提供的一个工作流案例

```
#工作流案例
https://symfony-workflow-demo.herokuapp.com/
```

工作流组件提供了两种形式的工作流，一种是状态机，一种是工作流。两者最大的区别就是状态机，同时只允许有一个状态。而工作流可以同时有多个状态。

我们先来学习一下工作流，点击Article，我们创建一个文章。标题叫做Article 476。点击创建现在我们的文章就已经创建好了，当前工作流的状态是空的数组。默认情况下，它在草稿状态。

我们登录Alice用户，Alice用户可以对工作流进行一个状态转换，我们点击这个状态转换。点击之后目前文章的状态是等待记者的确认和等待拼写检查。我们登录记者的用户，记者用户只可以发送记者状态的转化，这是记者用户的一个工作，我们点击转换。点击转换之后，当前的文章状态是记者用户已经认证了，我们需要等待拼接检查用户进行检查，退出当前用户。我们登录拼写检查用户，现在拼写检查用户可以对拼写检查工作进行确认，当他确认之后，当前文章的状态就变成了拼写检查用户已确认。当文章处于这两个状态时，文章就可以进行发布了，我们点击publish按钮，现在文章就已经成功的变成发布状态了。这就是工作流的一个流程。

我们回到网站的首页，查看一下状态机。我们创建一个任务，状态机同时只有一个状态。这里有个新的任务，我们点击开始处理按钮，现在任务的状态就是正在处理中。当当前的任务出现临时错误时，它就会变换到积压状态。我们点击这个按钮，在积压状态下的任务，它可以稍后再进行重试。我们点击重试，点击重试之后，它又回到处理状态。当出现永久性的错误时，当前的任务就会失败，失败之后，那么任务就不会再进行执行了。我们点击永久性错误，现在任务就到了失败的状态，当然任务处理过程中没有错误的话，就会变成处理完成的状态了。这就是状态机的一个使用案例。

我们回想一下现实中的案例，我们通常在下单时订单有订单的状态。有待付款，已付款，待发货已收货。订单的状态我们可以使用状态机来进行配置。然后是我们的请假流程，当我们在工作中需要请假时，首先我们要写请假条，然后由我们的部门主任进行签字，然后再由经理签字，当两个上级签字完成之后再交给人力部门。我们就可以完成请假的流程了。请假的流程我们可以定义为一个工作流。

在下节课，我们对管理端的文章添加工作流。