

首先我们删除上节课所下的断点，在Phpstorm底部我们打开debug选项卡。点击这个按钮展开所有的PHP断点，选中之后点击这个删除，然后我们关闭xdebug监听。

回到浏览器刷新页面，当我们的浏览器访问一个地址时，通常会发送一个GET请求，我们可以在浏览器的访问地址上，使用'?'加上'参数名'='参数值'的方式来添加请求参数。

当我们敲击回车访问时，浏览器会将这个地址和page参数一块发送到我们的服务器端，那么服务器端该如何获取到这个page参数呢？

对于HTTP请求在访问地址后面添加的参数通常叫做query参数。

回到项目，我们在action函数中可以添加参数。因为Symfony实现了PSR标准规范的容器接口，可以通过使用依赖注入的方式来解耦合代码。

我们可以在index方法中直接注入Request对象，这个Request我们要选择HttpFoundation这个命名空间下的类。Symfony为我们提供了dump方法，我们可以用dump方法来查看一下Request对象。我们在这里我们输入die方法，来暂停后面代码的执行。

再次刷新页面，Symfony将我们的请求和请求参数封装成了Request对象。我们来看一下我们的query属性，page参数就在query属性中。另外还有request属性，files属性，cookies属性，headers等等属性。request属性通常用于POST请求参数的获取，files通常用于文件的接收。那么我们该如何获取到page的值呢？Request对象为我们封装了很多方法，获取query，然后get()，传入'page'，我们再次查看一下\$page。

现在我们已经获取到了page参数(口误)。这里加了两个引号，这是一个字符串类型。如果我们想获取到整数类型该怎么办呢？Request仍然给我们做了封装，我们使用getInt()。

再次刷新，这样就变成了一个整数类型。我们按着command键把鼠标移动到getInt方法中，能看到getInt有两个参数。第一个参数是query参数的键，第二个参数是默认值。如果我们的访问地址中没有设置这个键，它将返回一个默认值。

我们在后面输入default值为1，然后回到浏览器，我们去掉page参数，再次访问页面，我们看到默认值为1。

下面我们来发送一个POST请求，我们看Symfony如何处理POST请求中的参数。我们打开postman，我们新建一个请求，请求的方式，我们选择POST方式，地址输入127.0.0.1:8000/test路径。POST请求我们需要在Body中填入表单的值，这次我们输入name等于hello。

回到项目，我们再次使用dump方法来查看一下Request对象。发送请求，我们点击这个preview选项卡。在Request对象的request属性中，我们看到了name参数。我们可以用Request对象的request属性来获取到对应的参数。

再次发送请求，我们看到我们已经获取到了name参数。Symfony为我们提供了另外一个方法，还可以省略十七行十八行的代码，输入dd()方法来查看变量的值。

按着command点击dd()，dd()方法是对dump方法和exit方法的一个封装，删掉18行19行代码，再次发送请求，显示的效果是一样的。

继续修改代码，我们希望在/test路径后面添加一个name参数，当我们访问时，让页面显示我们的名字。这里我们修改一下，name这里我们要使用query属性，修改hello word为我们\$name变量。

再次刷新页面，我们的名字已经显示了，这就是我们第一个最简单的Symfony页面。

现在我们来深入学习一下，我们看index方法中的Request对象是如何注入到方法中的。我们回到上一节课，在流程图中当Kernel对象接收到Request对象后，会首先找到那个controller方法，再根据controller方法和请求对象来解析它的所有参数。

现在我们来深入学习一下，回到项目，还记得上节课的handleRaw方法吗？我们来双击shift键，输入HttpKernel。如果你的编辑器没有找到这个文件的话，你要勾选这项。

现在我们找到handleRaw方法，在handleRaw方法中，我们下个断点跟踪这段代码。监听xdebug请求，刷新第一个页面。

这时断点就断在了handleRaw方法中，我们看现在的controller方法，它就是TestController的index方法。在这一步它会获取到index方法的所有参数，就是Request这个参数。

我们进入这段代码，我们点击进入，我们看50行51行代码，这是一个双重循环，Symfony会首先根据我们的controller方法获取到所有参数的Metadata，获取完之后进行一项遍历，因为controller方法的参数可以有多种类型，Symfony也针对不同的参数类型提供了不同的解析器。

点击左边Controller文件夹，文件夹下有个ArgumentResolver文件夹。点击，下面提供了很多种的resolver解析器，我们当前controller(口误)方法的参数只有一个Request对象 它会使用RequestValueResolver对象来处理这个参数。

我们在56行下断点，点击左下角执行，它断到了56行，我们再看这个\$resolver变量，\$resolver变量它指向的就是RequestValueResolver类。它是这个类的一个实例，我们进入这个类查看一下代码如何执行。

找到RequestValueResolver类，我们看一下supports方法。我们参数的类型是Request类型，它就往下执行，在resolve方法中，直接返回了Symfony已经封装好的Request对象。所以我们可以直接在action方法中使用这个Request对象。

在handleRaw第151行下断点，我们看一下最后的\$arguments参数，直接点击运行。断点停在了151行，我们看\$arguments变量，\$arguments变量只有一个，它是我们的Request对象。所以我们可以直接在action方法中直接使用这个Request对象来获取所有的请求参数。

我们看第12行代码，这个写法是PHP8.0版本提供的注解的写法，第一个参数是路由指向的路径，第二个参数是路由的名称。

在下一节课，我们将深入的学习一下路径中的占位符如何使用。