**简介**

**AbpController基类**

这是一个简单的继承自AbpController的控制器：

public class HomeController : **AbpController**

{

public ActionResult Index()

{

return View();

}

}

**本地化**

ABP定义了L方法，使[本地化](http://www.cnblogs.com/kid1412/p/6014030.html" \t "http://www.cnblogs.com/kid1412/p/_blank)更加容易，例如：

public class HomeController : AbpController

{

public HomeController()

{

**LocalizationSourceName = "MySourceName";**

}

public ActionResult Index()

{

var helloWorldText = **L("HelloWorld");**

return View();

}

}

要先设置LocalizationSourceName，才能让L方法正常工作，你可以在你的控制器基类里设置，这样就不用为每个控制器重复设置。

**过滤**

**异常处理和结果包装**

所有的异常都被自动处理、日志并返回一个适应的响应给客户端，更多信息查看[异常处理](http://www.cnblogs.com/kid1412/p/6014030.html" \t "http://www.cnblogs.com/kid1412/p/_blank)。

如果Action的返回类型是JsonResult（或异步的Task<JsonResult>），ABP也会默认地去包装这个结果。

**审计日志**

AbpMvcAuditFilter用来集成到[审计日志系统](http://www.cnblogs.com/kid1412/p/6009881.html" \t "http://www.cnblogs.com/kid1412/p/_blank)，你可以用Audited和DisableAuditing特性控制Action和控制器的审计日志。

**验证**

AbpMvcValidationFilter自动检查ModelState.IsValid并在检测到非法时阻止Action执行。实现输入DTO的验证在[验证文档](http://www.cnblogs.com/kid1412/p/6006293.html" \t "http://www.cnblogs.com/kid1412/p/_blank)里进行了描述。

**授权**

你可以为你的api控制器或Action使用AbpApiAuthorize特性，阻止未授权用户访问它们，例如：

public class HomeController : AbpController

{

**[AbpMvcAuthorize("MyPermissionName")]**

public ActionResult Index()

{

return View();

}

}

**工作单元**

AbpMvcUowFilter用来集成到[工作单元](http://www.cnblogs.com/kid1412/p/5996459.html" \t "http://www.cnblogs.com/kid1412/p/_blank)系统，在一个Action执行前自动开始一个工作单元，并在Action执行结束后完成工作单元（如果没有异常抛出）。

**防伪造**

AbpAntiForgeryMvcFilter自动保护Mvc的Action，阻止来自CSRF/XSRF的POST、PUT和DELETE请求攻击。更多信息查看[CSRF文档](http://www.cnblogs.com/kid1412/p/6014030.html" \t "http://www.cnblogs.com/kid1412/p/_blank)。