



## 自定义登录失败处理器

### 源码分析

failureForwardUrl()内部调用的是 failureHandler() 方法

```
public FormLoginConfigurer<H> failureForwardUrl(String forwardUrl) {  
    this.failureHandler(new ForwardAuthenticationFailureHandler(forwardUrl));  
    return this;  
}
```

ForwardAuthenticationFailureHandler 中也是一个请求转发，并在request 作用域中设置 `SPRING_SECURITY_LAST_EXCEPTION` 的 key，内容为异常对象。

```
public class ForwardAuthenticationFailureHandler implements AuthenticationFailureHandler {  
    private final String forwardUrl;  
  
    public ForwardAuthenticationFailureHandler(String forwardUrl) {  
        Assert.isTrue(UrlUtils.isValidRedirectUrl(forwardUrl), () -> {  
            return "" + forwardUrl + " is not a valid forward URL";  
        });  
        this.forwardUrl = forwardUrl;  
    }  
  
    public void onAuthenticationFailure(HttpServletRequest request, HttpServletResponse response, AuthenticationException exception) throws IOException, ServletException {  
        request.setAttribute(S:"SPRING_SECURITY_LAST_EXCEPTION", exception);  
        request.getRequestDispatcher(this.forwardUrl).forward(request, response);  
    }  
}
```

### 代码实现

#### 新建控制器

新建 com.xxxx.handler.MyForwardAuthenticationFailureHandler 实现 AuthenticationFailureHandler。在方法中添加重定向语句

```
package com.xxxx.springsecuritydemo.handler;  
  
import org.springframework.security.core.AuthenticationException;  
import org.springframework.security.web.authentication.AuthenticationFailureHandler;  
  
import javax.servlet.ServletException;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import java.io.IOException;  
  
/**  
 * @author zhoubin  
 * @since 1.0.0  
 */  
public class MyForwardAuthenticationFailureHandler implements AuthenticationFailureHandler {  
  
    private String url;
```



```
public MyForwardAuthenticationFailureHandler(String url) {
    this.url = url;
}

@Override
public void onAuthenticationFailure(HttpServletRequest request, HttpServletResponse
response, AuthenticationException exception) throws IOException, ServletException {
    response.sendRedirect(url);
}
}
```

## 修改配置类

修改配置类中表单登录部分。设置失败时交给失败处理器进行操作。 `failureForwardUrl` 和 `failureHandler` 不可共存

```
//表单提交
http.formLogin()
    //自定义登录页面
    .loginPage("/login.html")
    //当发现/login时认为是登录，必须和表单提交的地址一样。去执行UserServiceImpl
    .loginProcessingUrl("/login")
    //登录成功后跳转页面，POST请求
    // .successForwardUrl("/toMain")
    //和successForwardUrl不能共存
    .successHandler(new MyAuthenticationSuccessHandler("http://www.baidu.com"))
    //登录失败后跳转页面，POST请求
    // .failureForwardUrl("/toError")
    .failureHandler(new MyForwardAuthenticationFailureHandler("/error.html"))
    .usernameParameter("myusername")
    .passwordParameter("mypassword");
```