//概述

HttpServletRequestWrapper是一個java類別

可以繼承它以後用不同方式override或增加 request相關的各項功能

ex 新增add方法來修改用戶上傳的信息

//應用: 一般用於防範資安漏洞，轉換用戶上傳信息

//1. 建立RequestWrapper類別並繼承HttpServletRequestWrapper

**public** **class** RequestWrapper **extends** HttpServletRequestWrapper{

//新增paramMap變數，複製request中所有參數來改造

Map<String,String[]> paramMap=**new** HashMap<String,String[]>();

//constructor

**public** RequestWrapper(HttpServletRequest request) {

**super**(request);

//構造函數--將所有request中的參數加入自己的變數

//this.paramMap=request.getParameterMap();

//copy過來一份，不指向相同記憶體，否則會發生mapLock

**this**.paramMap.putAll(request.getParameterMap());

}

//新增addParameter方法來增加參數

**public** **void** addParameter(String name , Object value) {

**if**(value != **null**) {

**if**(value **instanceof** String[]) {

paramMap.put(name , (String[])value);

}**else** **if**(value **instanceof** String) {

paramMap.put(name , **new** String[] {(String)value});

}**else** {

paramMap.put(name , **new** String[] {String.*valueOf*(value)});

}

}

}

//重寫取得參數方式(要從自己的paramMap中取得)

**public** String getParameter(String name) {

String[]values = paramMap.get(name);

**if**(values == **null** || values.length == 0) {

**return** **null**;

}

**return** values[0];

}

}

//2. 建立filter調用

**public** **void** doFilter(ServletRequest request, ServletResponse response, FilterChain chain) **throws** IOException, ServletException {

RequestWrapper rw=**new** RequestWrapper((HttpServletRequest) request);

//新增一筆request參數

rw.addParameter("testMapKey", "我是requestWrapper進去的Value");

rw.addParameter("username", "哈哈，帳號被我改掉了");//改User輸入參數(帳號)

rw.addParameter("pwd", "哈哈，密碼被我改掉了");//改User輸入參數(密碼)

chain.doFilter(rw, response);

}

//3. 在servlet中調用測試

System.***out***.println(request.getParameter("username"));

System.***out***.println(request.getParameter("pwd"));

System.***out***.println("requestWrapper : "+request.getParameter("testMapKey"));

request.getRequestDispatcher("Test.jsp").forward(request, response);;