

Input Validation on Fitness Equipments Information System Based on Struts2

Kai Su

Teaching and Research Institute of Sports, Dalian Nationalities University, Dalian, P.R. China

sukai123456789@163.com

Keywords: Struts2; verification framework; fitness equipments; input validation

Abstract. Due to the openness of the Web application, and the data collected by the input page is very complex, to prevent these illegal importations to enter the system, therefore, input validation is a question that must be solved by all Web applications, this paper studies based on Struts2 validation framework. Firstly, studies Struts2 framework and workflow; then, the logical structure design of the database; and finally, using required, requiredstrin, int, double, date, email, url, stringlength and others provided by Validator verification framework in Struts2 to research input validation file design. Content of this paper is the basic work in the software system development, and has an important role for improving the quality and efficiency in the fitness equipment system development.

Introduction

Due to the openness of the Web application, and the data collected by the input page is very complex, not only contains the input errors for normal user, but also contains input errors come from malicious user. A robust application system must prevent illegal import from these applications, and prevent illegal input entering the system, so as to ensure the system is not affected. Therefore, input validation is a question that must be solved by all Web applications. Struts2 is the next generation of the Struts products, is a new framework merged based on the Struts1 and WebWork. WebWork is the core of the Struts2 which uses the interceptor mechanism to handle user request and makes the business logic controller separate from ServletAPI completely [1]. Input verification generally be divided into client validation and server validation, the most common way is to use the JavaScript scripting language to verify the data inputted by user in the client validation, server validation is to use the program code or profile to verify the data entered by the user on the side of server. Relying solely on traditional thinking, traditional methods cannot meet the needs of modern management. Using modern computer information technology research and develop a information system about fitness equipment management, which can effectively manage the equipment to make fitness equipment management work in conjunction with the system more efficient and convenient. This paper studies the input validation of the fitness equipment management system based on the Struts2 validation framework.

Struts2 Framework Structure

Struts2 framework can be divided into three parts: the core controller FilterDispatcher, business controller Action and user-implemented business logic components. In which, the core controller FilterDispatcher is the basis of Struts2 framework, including the control flow and processing mechanism inside the framework. Business controller Action and business logic components require the user to achieve. When the user develops Action and business logic components, they also need to write the relevant configuration file for the core controller FilterDispatcher using; Business controller Action is written and achieved by the developers, Action class can be a simple Java class, and completely separate from Servlet API. Action generally have an execute () method, and can also define other business control methods; business model components can implement business logic modules, which can be EJB, POJO or JavaBean, in the actual development, the difference and definition of business model components is very vague, and also beyond the scope of

Struts2 framework. Different developers or teams have their own way to implement business logic modules; the purpose of Struts2 framework is to use Action to invoke business logic modules. Struts2 framework structure is shown in Fig. 1 [2].

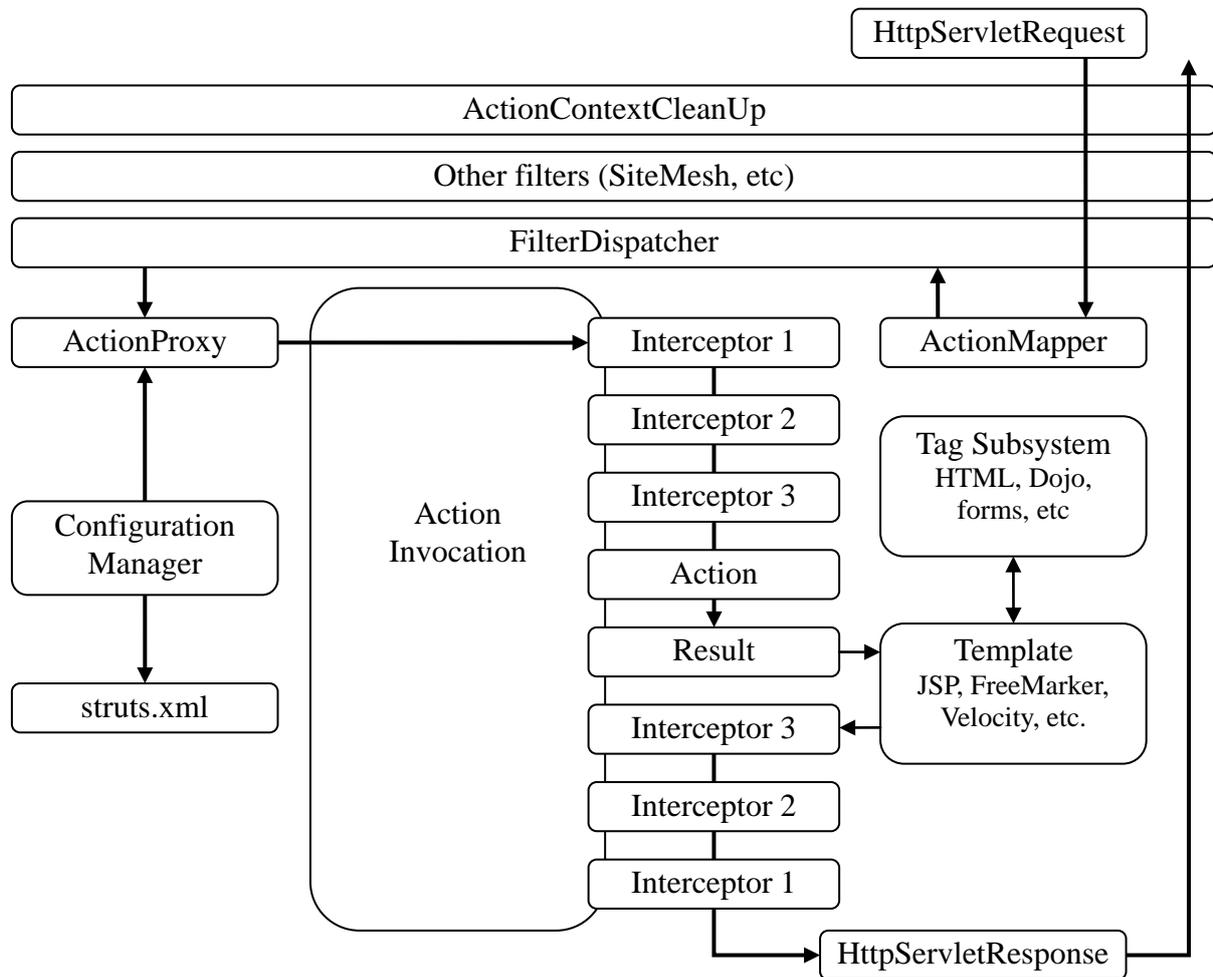


Fig. 1. Struts2 Framework Structure

Struts2 framework workflow is as follows: The client submits a HttpServletRequest request, action or JSP pages; the request is submitted to a series of Filter filter, ActionCleanUp and FilterDispatcher etc.; FilterDispatcher is the core of Struts2 controller, usually is a final filter in the filter chain; after the request is sent to the FilterDispatcher, when FilterDispatcher asked ActionMapper need to call a Action to deal with the Request; if ActionMapper decides to need to call a Action, FilterDispatcher sends the request ActionProxy for processing; ActionProxy asks configuration files struts.xml in the framework by Configuration Manager, and find the Action class called; ActionProxy creates a ActionInvocation instance, and calls Action by proxy mode; after Action completed returns a result string, then pass the Interceptor by reverse order; finally, ActionInvocation instance is responsible for finding the corresponding result according to the result elements deployed in struts.xml and deciding the further output.

Database Logical Structure Design

Database design is usually based on the needs analysis, conceptual structure design, logical structure design, physical structure design and security design. The core content of this article is input validation, input validation only relates to the logical structure of the database. Therefore, this article only studies logical structure design. Logical structure reflects the logical relationships among components and associates with the database management system (DBMS). The database logical structure design based on Microsoft SQL Server 2008 database management system is shown in Table 1.

Table 1. Fitness equipments information table

No	FieldName	DataType	Width	No	FieldName	DataType	Width
1	EquipmentCode	varchar	10	11	Address	varchar	200
2	EquipmentName	varchar	100	12	PostalCode	char	6
3	ClassificationCode	varchar	2	13	E-mail	varchar	50
4	ClassificationName	varchar	50	14	Homepage	varchar	200
5	ProductionDate	datetime	8	15	PurchasingDate	datetime	8
6	StorageLocation	varchar	200	16	PurchasingQuantity	int	2
7	SupplierCode	varchar	5	17	PurchasingPrice	money	8
8	SupplierName	varchar	100	18	Specification	varchar	100
9	ManufacturersCode	varchar	5	19	MeasurementUnit	varchar	10
10	ManufacturersName	varchar	100	20	Introduction	varchar	500

Validation File Design

Each Action separately correspond a validation file in the Struts2, named as <Action-name>-validation.xml, and stored in the same package with the Action, verification can be realized only need to configure a validation file. After classified the 20 fields in Table 1, need to use 7 kinds of validator. Validation files design as follows:

<pre><? Xml version="1.0" encoding="UTF-8"?> <!DOCTYPE validators PUBLIC "-//OpenSymphony Group//XWork Validator 1.0/EN" http://www.opensymphony.com/xwork/xwork- alidator-1.0.2.dtd" > <validators > <!-- Verify "non-empty" sample --> <field name="EquipmentCode"> <field-validator type="required"> <message> Equipment code Not be empty</message> </field-validator> </field> <!-- Verify "string length" sample --> <field name="PostalCode"> <field-validator type="stringlength"> <param name="minlength">6</param> <param name="maxlength">6</param> <message> Postal code Must be 6 bite </message> </field-validator> </field> <!-- Verify "integer" sample --> <field name="PurchasingQuantity"> <field-validator type="int"> <param name="min">0</param> <param name="max">999</param> <message> Purchasing quantity Must be between 0 and 999 </message> </field-validator> </field></pre>	<pre><!-- Verify "double-precision float number" sample --> <field name="PurchasingPrice"> <field-validator type="double"> <param name="min">0</param> <param name="max">9999.99</param> <message> Purchasing price Must be between 0 and 9999.99</message> </field-validator> </field> <!-- Verify "date" sample --> <field name="ProductionDate"> <field-validator type="edate"> <param name="min">1949-01-01</param> <param name="max">2049-12-31</param> <message> Production date Must be between \${min} and \${max} </message> </field-validator> </field> <!-- Verify "E-mail address" sample --> <field name="E-mail"> <field-validator type="email"> <message>E-mail Address format is incorrect</message> </field-validator> </field> <!-- Verify "Web site" sample --> <field name="Homepage"> <field-validator type="url"> <message> Homepage format is incorrect </message> </field-validator> </field></validators></pre>
---	--

Description of the validation files as follows [3-5]:

(1) Verify non-empty. When entering the data, may be some data is not collected or uncertain, it can be

temporarily empty. Validator type is "required", in this paper, "Equipment code" field as an example.

(2) Verify the string length. All data is the field in the type of varchar and char, need to verify string length. Validator type is "stringlength", in this paper, "Postal code" field as an example.

(3) Verify integer. "PurchasingQuantity" field requires verifying integer. It requires the value of field within the specified range. Validator type is "int".

(4) Verify double-precision float number. "PurchasingPrice" fields need to verify double-precision float number, which requires the values within the specified range, and the verified Action property is double-precision float number. Validator type is "double".

(5) Verify the date. "ProductionDate, PurchaseDate" two fields need to verify the date format that requires date values must be within the specified range. Validator type is "date". In this paper, "ProductionDate" field as an example.

(6) Verify E-mail address. "E-mail" field needs to verify E-mail address. If the character of this field is non-empty, it must be a lawful E-mail address. Validator type is "email".

(7) Verify URL. "Homepage" field needs to verify the site. If the character of this field is non-empty, it must be a lawful URL. Validator type is "url".

Conclusion

Scientific management methods can improve using efficiency and fitness equipment and ease the shortage of fitness equipment, strengthening fitness equipment management helps extend the life of sports and fitness equipment, and paying attention on fitness equipment management helps organize effective physical teaching [6, 7]. Fitness equipment management system based on Web technology has important theoretical and practical significance. Input validation is an important task of the system development, this paper studies the input validation by using required, requiredstrin, int, double, date, email, url, stringlength etc. provided Validator verification framework and Struts2, and has an important role for improving the quality and efficiency in the fitness equipment system development.

Acknowledgement

This work is supported by youth project on freedom scientific research fund of central university (20150436).

References

- [1] Z. G. Zhang, "Development and Application on Struts2 Framework Technology," Computer CD Software and Applications, vol. 16, no. 23, pp. 119-120, 2013.
- [2] huamengxing blog, "Struts2 architecture diagram," <http://www.blogjava.net/huamengxing/archive/2009/10/21/299153.html>, 2014-12-15.
- [3] X. M. Li, "Based on the Input Validation Functions Struts2 Framework Implementation and Application," Computer Knowledge and Technology, vol. 9, no. 35, pp. 8201-8202, 2013.
- [4] Luob blog, "Struts2 Use the Validation Framework to Validate the Data (thirty-seven)," <http://takeme.iteye.com/blog/1667716>, 2014-12-15.
- [5] D. C. Zhou, F. J. J, D. H. Guo, "Implementation of validation data and integration regular expression based on Struts2 framework," Computer and Modernization, vol. 27, no. 1, pp. 85-87, 2011.
- [6] J. Q. Sun, "Design and Implementation of Sports Equipment Management System," Master's Degree of Inner Mongolia University, 2013.
- [7] M. Wang, Y. Q. Meng, Z. Zhang, "An Analysis and Research on the Use of Fitness Apparatus by Beijing Residents for Exercise," Journal of Capital Institute of Physical Education, vol. 19, no. 3, pp. 58-61, 2007.