

Internet Data Validation

Loredana MOCEAN
 Babes - Bolyai University of Cluj – Napoca
 Business Information Systems Department
 mloredana@econ.ubcluj.ro

In Business Information Systems, data validation is the process of ensuring that a program operates on clean, correct and useful data. It uses routines, often called validation rules, that check for correctness or meaningfulness of data that are input to the system.

This paper presents the main aspects in data validation. In the paper is exemplified some functions and methods for data validation in JavaScript. It is exemplified the usage of the theoretical tools by applying them on a few examples chosen from real applications.

Keywords. *Data Validation, XML document, JavaScript*

Introduction.

Through data validation is apprehended the operation of checking the correctness of the data. A data is considered correct if respects a series of arithmetic and logical preestablished conditions. The validation of the date supposes the specification of next elements:

- The checking conditions;
- The action which must be executed when the data is or not correct;
- The way of point out the errors and structure of the messages;
- The correction way of errors;
- The way of input the records after the correction.

The data validation can be done to the following levels:

- *level of field* (a field respects its own conditions);
- *level of record* (the observance of relations between the fields of the same record);
- *level of more records* (the observance of relations between the fields of different records, completeness on packages of documents, the correctness on totals on packages);
- *file level* (completeness, correctness on totals).

The most important data validations to field's level of a record are referring to:

- Existence;
- Natura;
- Length;

- Sign;
- The belong of a set or of a values list;
- The observance of arithmetic and logical correlations between fields.

If a data doesn't respect all the established conditions, it will be generate an error and it will be need to re-enter it.

Enter it can be repeated, until is obtained a correct data or until is exceeded a preestablished number of reinsertions. When a field is correct, is passed to the introduction and the validation of the next field. When the record is correct, is written the in the externally support and is passed to next record.

Statement of the problem

Many of the XML applications contain different forms of searching, authentication, editing. Big majority of us we confronted with problem of creation such an application. But the application we have created is safe? It's an important question. One of the important thing to remember each time we write an application is not trusting in the input of the user. Some important rules are described as follow:

- We must try to validate the entire input of the user. That's why is necessary a good comprehension of the parameters which come from the client (hide fields, cookies etc.). The validation of the data must be done before any another operation;
- We must create and use validation functions, such as *isValidEmail()* which

returns a logical value (true or false) depending by the data introduced by the user;

➤ We must always verify the format and the domain of the data. The domain refers especially on numeric data (for example, in an invoice, de price can't be negative);

➤ Data validation must be done on the client side but not only the client side in JavaScript. The browsers permit deactivation of executing JavaScript code. If the part of code from server is based just on the validation did on the part of customer, the application is one total uncertain;

We must have big care to data on which we send them by dint of the fields "hidden"; Even if these fields haven't seen in the webpage of the user they are not sure. On the contrary, a hidden field is easy of modified as visible. Its not very well to send data within this fields. A method much safe is the use of sessions, because these are stored on servers, the customers don't have the access to them.

Results and discussions

XML data are validated within a scheme in order to assure us that all the XML imported or exported data from one or more cells, from a mapped zone from a worksheet, correspond XML schema from the XML source panel of activity.

One of the most important solution for XML data validating is JavaScript. JavaScript is a programming language which permits to offer our documents a dynamic and interactive character. It is easy to learn and use. Javascript is dedicated to those who are not programmers and want to improve the appearance and the functionality of the Web Pages.

A script written in JavaScript may be placed in the web document, it is executed by the

The code for the previous form is shown bellow:

```
function test1(form) {
if (form.text1.value == "")
alert("Introduceti un sir de caractere!")
else {
alert("Buna ziua, "+form.text1.value+"! Introducere corecta!");
}
}
function test2(form) {
```

browser with the document which contain it. The Internet Explorer browser correctly execute the scripts.

Javascript is interpreted language as Perl, but in this case isn't due to you to distress in installation of an interpreter because this is figured in browser. The source code can be included in the document or can be putted into a file which is referred from the document.

As in the case of the CGI scripts, it isn't necessary to know the JavaScript language for inserting and using with big success these scripts.

There are an impressive number of scripts for utilization, which can be easily integrated in the code source of the documents.

The validation of the data entered in the form

The forms are objects very frequent meet on Internet. Data entered in the form are, as a rule, sended backward to server by electronic messagery to an email account. But we must assure that input data from user are correct.

First, we will create a simple script. The document will contain two text elements. The user introduces the *Company Name* and the *Fiscal Registration Code*. He can test the introduced values through auctioning the two buttons. Also the user can try to enter nothing and to press buttons.

Denumirea firmei:

Codul unic de inregistrare:

Fig.1. The first validating form

```

if (form.text2.value == "")
alert("Introduceti un sir de caractere!")
else {
alert("Buna ziua, "+form.text2.value+"! Introducere corecta!");
}
}

<body>
<form name="first">
Denumirea firmei:<br>
<input type="text" name="text1">
<input type="button" name="button1" value="Test Introducere"
onClick="test1(this.form)">
<P>
Codul unic de inregistrare:<br>
<input type="text" name="text2">
<input type="button" name="button2" value="Test Introducere"
onClick="test2(this.form)">
</body>

```

In the first case, the entered element shall returned a prompt of error if is none entered. Of course, this kind of check will allow to user to introduce a false name. The browser accepts also digits. This is not the best mod to verify.

In the case of e-mail address, the criteria accepted entered as e-mail address has the character @. One single @ character is

```

function test3(form) {
  if (form.text3.value == "" ||
      form.text3.value.indexOf('@', 0) == -1)
    alert("Adresa de mail invalida!");
  else alert("OK!");
}
<body>
<form name=" a treia">
Introduceti adresa de mail:<br>
<input type="text" name="text3">
<input type="button" name="button3" value="Test verificare"
onClick="test3(this.form)">
</body>

```

Sometimes we want to restrict the string entered to some characters or numbers. We may think at a telephone number, which contains only digits, but in some situations it is written with some symbols which are characters not numbers (for example 0745/0990... or 0745-12222...). If the number is written correctly, only with digits we can verify if the input data don't contain

enough but is not representative. All the e-mail addresses contain @ so, its obviously to search @ in address.

Introduceti adresa de mail:

Fig.2. The second validation form

other characters. We must extend the script to check out the digits and other symbols.

Telefon:

Fig.3. The third validation form

The source code for this script is presented in the following rows:

```

function test4(input) {
  var ok = true;
  for (var i = 0; i < input.length; i++) {
    var chr = input.charAt(i);
    var found = false;
    for (var j = 1; j < text4.length; j++) {
      if (chr == test4[j]) found = true;
    }
    if (!found) ok = false;
  }
  return ok;
}
function test(input) {
  if (!test4(input, "1", "2", "3", "4",
    "5", "6", "7", "8", "9", "0", "/", "-", " ")) {
    alert("Introducere gresita");
  }
  else {
    alert("Introducere ok!");
  }
}
<body>
<form>
Telefon:<br>
<input type="text" name="telefon" value="">
<input type="button" value="Verificare"
  onClick="test(this.form.telefon.value)">
</form>
</body>

```

Conclusions

XML is a very important technology which permits a risen interoperability of information between various systems. XML is the technology within data can be transferred on most diverse platforms.

Data validation strategies are often heavily influenced by the architecture for the application. Java/XML-based data validation approach separates the implementation of common data validation reasoning code from the business rules and criteria data used to validate user input. The validation reasoning code is implemented in Java, while the business-specific rules and data are specified in XML. This approach provides a powerful and flexible way for application developers to specify data validation in a manner that is secure and easy to manage, while decoupling validation rules from the server-side business logic implementation.

These were the reason for which we wanted to present the XML technology and a few

procedures of data validation using JavaScript.

References

- [Buraga04] Buraga, S., *Semantic Web * fundamente și aplicații*, Matrix Rom, Bucuresti, 2004
- [Koch07] Koch, S., *Voodoo's Introduction To JavaScript*, white paper, 2007
- [Mindruta05] Mindruta, C., *Arhitecturi, tehnologii si programare WEB*, Matrix Rom, Bucuresti, 2005
- <http://www.pcworld.ro>
- <http://www.devx.com/Java/Article/16407/1763>
- http://www.softpageinternet.ro/manual_html
- <http://www.afaceri-online.net>