WALT PLATFORM FOR WEB APPLICATION DEVELOPMENT

V.V. Korenkov, S.V. Kuniaev, S.V. Semashko, I.A. Sokolov [a]

Joint Institute for Nuclear Research, 6 Joliot-Curie St., Dubna, 141980, Russia

E-mail: [a] isokolov@jinr.ru

The article describes the Web Application Lego Toolkit (WALT), which is a template-oriented platform designed for the development of web applications of various degrees of complexity. Unlike some other platforms, which represent a "magic black box", the main idea of WALT is to provide transparent, extensible, and modifiable tools for solving some specific problems that arise when developing web applications. WALT is primarily intended for use by individual full-stack developers or small groups of developers. The article depicts the WALT architecture and provides examples of its use for the development of JINR corporate web applications.

Keywords: web application, service, template, platform, java, request

Vladimir Korenkov, Sergey Kuniaev, Sergey Semashko, Ivan Sokolov

1. Introduction

Currently, there is a great variety of platforms for the development of web applications: Django, ASP.NET Core, Express, Angular, etc. [1, 2, 3, 4]. These platforms provide developers with a rich set of tools and libraries, which allows them to create advanced web applications. As a rule, a relatively large team of developers works on a project in large or medium-sized IT companies. At the same time, there is a division within the team according to the nature of the work: design, frontend web development, backend development, testing. Listed above platforms are well suited for such a web development organization.

However, in the scientific field and in our real life, very often only one or two people are engaged in the development of a web application, they are essentially full-stack developers. In addition, often the same people simultaneously work on other projects and support applications which are already in operation. Thus, the development process follows the principles of Agile Programming [5].

The WALT platform presented in the article is very effective for such organization of work and can be successfully used for the development of web applications of various complexity by a small team. It also allows expanding and modifying the functionality of the application during its operation. Web applications developed using WALT are characterized by high performance and humble server resource requirements. The platform is easy to learn. To start using it, it is enough for a developer to have basic layout skills (HTML + CSS), be able to administer a DBMS, master the SQL query language, learn the WALT template language, and get acquainted with the existing set of WALT services.

2. Project overview

WALT is implemented as a Java^{tm} class library [6], but it does not require Java proficiency from the developer to perform typical tasks. WALT is a template-oriented platform. A simple language and corresponding interpreter were developed to use templates. This language allows to customize the output HTML code, SQL queries to the database, as well as to modify the processing route in run-time depending on the particular user request and intermediate results. In other words, template says to the Java-program what to do using a very laconic language and the Java-program does it. So, templates define the main part of the web application logic and behavior.

WALT makes it possible to combine the functionality of frontend and backend parts within a module and freely move the functionality from the frontend to the backend and vice versa. It is compatible with modern frontend tools like JQuery, Bootstrap, etc. This allows the developer to choose the optimal distribution of functionality between the frontend and backend right during the project implementation.

Unlike many others, WALT is a transparent and easily extensible platform.

2.1 WALT architecture

The WALT architecture is shown in Fig. 1. It is based on Java servlet technology [7]. Some kind of a servlet container (Servlet engine) is needed to run it. Apache Tomcat [8] is well suited and free. The Servlet engine receives an HTTP request from the client and passes it to the WALT application servlet. In the servlet, the request goes through several general initial stages of processing: creating and launching a separate thread to execute the request, extracting input data from the request and forming a list of parameter values, connecting to the database, authenticating the user, etc. After going through the general steps, based on the obtained information, servlet selects a specific module with templates and launches the corresponding service. The service extracts necessary information from the module templates and performs necessary actions based on the client’s request, namely, exchanging data with the database, preparing a response to the client, and other actions on the server-side. At the end of processing, the servlet sends the generated response to the client.

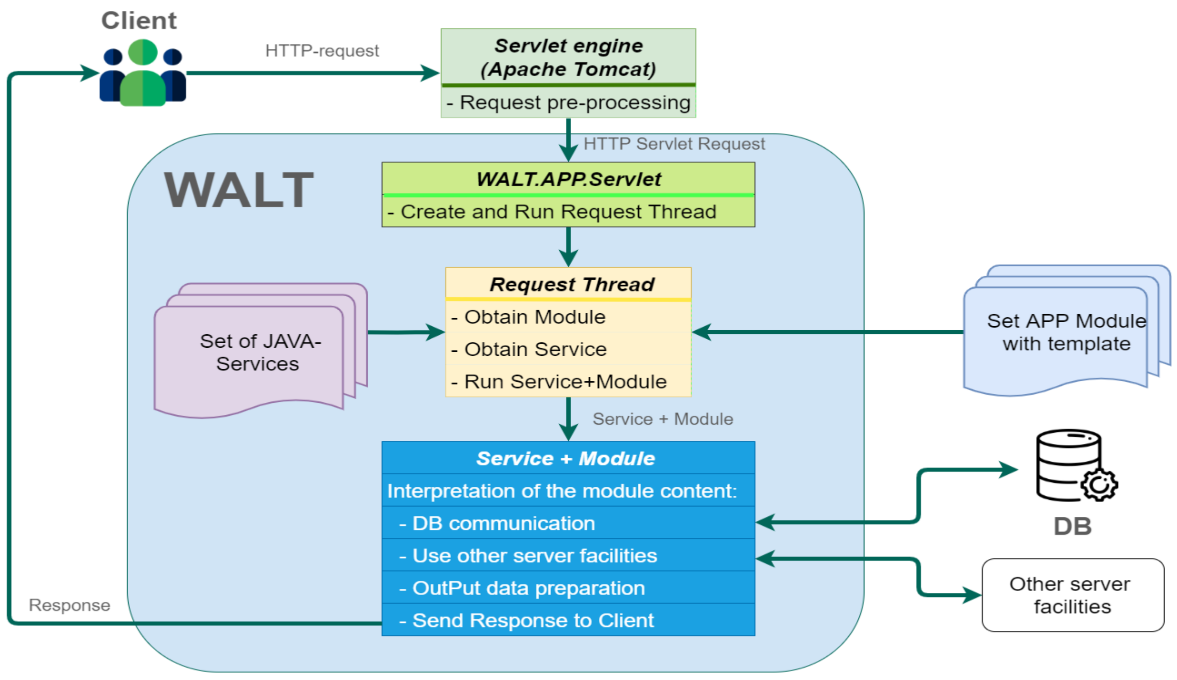


Figure 1. WALT architecture

2.2 Main WALT services

The WALT core provides the developer with 21 services. A service is a class written in Java, designed to perform a specific kind of tasks (working with data from a database, file operations, sending emails, image processing, etc.). The basic root service is the Java-class dubna.walt.service.Service. It is used as a parent class for all other services. Also Service itself together with corresponding modules can process many common tasks like execute a set of SQL-queries to database, prepare output HTML code, make some file operations etc. Other most frequently used services are:

- TableServiceSimple, TableServiceSortable, TableServiceComplex, TableServiceSpecial – different kinds of table-oriented services that can execute an SQL query to the DB and process the obtained recordset (typically, output results to the client, but other operations like update data can also be performed)
- CrossTabService – executes an SQL query and creates a cross-tab report similar to EXCEL's Pivot table
- CommandExecutor – executes a native host command and obtains the result
- ServiceUploadFile – uploads a file to the server
- ServiceBinaryData – sends data to the client in a binary form for downloading data as a file or for some other processing
- SendMailService – sends an email
- ServiceImportData, ServiceCopyData – copies data from a database to another one

Usually, the core services allow to solve at least 95% of the tasks within a web application. In rare cases, when the existing services cannot cope with some very specific tasks, or it is necessary to modify the processing, WALT makes it possible to expand the functionality. To do this, it is necessary to implement a new service based on the existing one. However, it requires some Java programming skills from the developer.

2.3 WALT template syntax and directives

Templates are grouped into modules, which are plain text files with a specific structure. Each module is designed to perform some specific task. It is made up of sections containing a code fragment that performs some part of the whole task. The use of sections allows following the principle "divide and conquer". Sections can be used both inside the same module and in other modules, which enables code reuse.

In practice, most of the tasks in web applications are related to database record processing. WALT enables solving such tasks by writing a simple module with information about necessary queries to the database and about the output HTML (or any other) code. The rest of the work is done by a core WALT service.

An important basic element of the template syntax is the "Parameters Substitution" construction. It is denoted in the code as #name#, where "name" is the name of the parameter whose value we want to use at this point of the code. In the following sample, the value of a parameter named "age" is inserted into the code instead of #age#:

```
Age: #age#
```

Another important element of the syntax is the "Condition Check" construction. It checks the fulfillment of the condition after the double question mark, and, if the result is false, the code line is ignored. In the following sample, the string "Some_line_of_code" is inserted into the code only if the value of the parameter "ID" is greater than 5 or the value of the parameter "Name" is equal to "Ivan", and the value of the parameter "age" is greater than 30:

```
Some_line_of_code    ??ID>5|Name=Ivan&age>30
```

"Parameters Substitution" and "Condition Check" allow controlling user request processing at runtime. For instance, they can be used to customize the SQL query template according to the user's request or to modify the output HTML code depending on the data obtained from the DB.

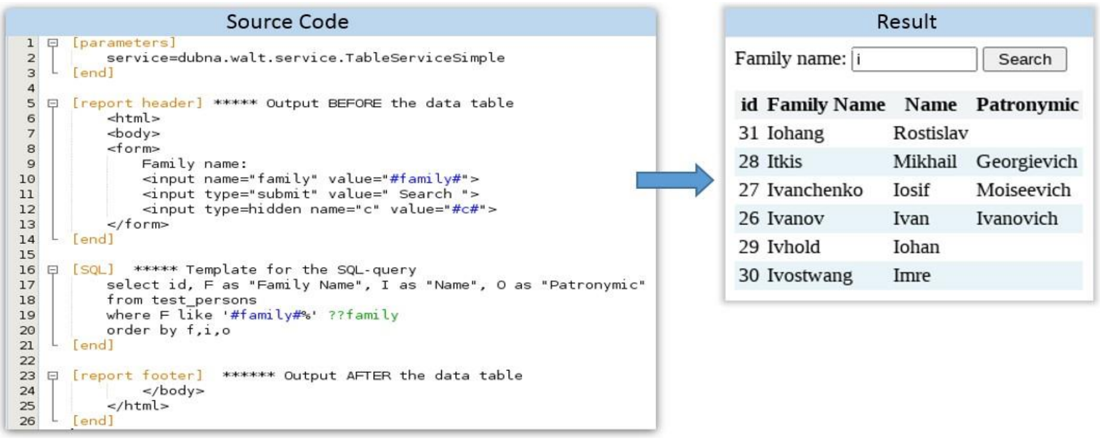
Another important element of the syntax is the directive. Directives make it possible to perform some separated completed actions on the backend. Directives extend the Services functionality and add flexibility. In the template, the directive call starts with the "$" sign. 26 directives are implemented in WALT:

- $INCLUDE – insert some section from this or another module into the code. It enables code reuse and makes it more compact and readable
- $GET_DATA – get necessary data by executing some SQL script or by other actions (e.g. read data file)
- $CALL_SERVICE – execute another module inside the parent one. It allows one to divide a complex task into several simple subtasks, as well as to reuse the code. It gives an opportunity to implement recursive calls.
- $JS, $JS_BEGIN - $JS_END – use the server-side JavaScript code. It often protects the developer from having to develop a new Java service.
- $READ_FILE, $COPY_FILE, $MOVE_FILE, $DELETE_FILE, $GET_FILE_SIZE – common file operations
- $GET_URL – open the specified URL and obtain its response
- And others such as $EXECUTE_LOOP, $IF - $ELSE - $ENDIF, $WAIT, $USE_DB, $CLOSE_DB, $STORE_PARAMETERS - $RESTORE_PARAMETERS, $GET_AUTH_URL, $BREAK, $PRINT, $LOG, etc.

The WALT template language is capacious and concise, which enables the creation of a compact module to solve complex tasks by decomposing it into simple parts and reusing the code.

2.4 Simplest usage sample

The code of a simple module and the result of its work are presented in Fig. 2. This module selects information about people from the database, as well as allows searching for people by the beginning of the family name. It contains an indication of the service to be used, templates for the SQL query to the database and the output HTML code. The rest of the task on fetching data, forming the HTML table and output to the client is carried out by the specified TableServiceSimple. Thus, the compact code performs some completed task.



```
[parameters]
    service=dubna.walt.service.TableServiceSimple
[end]

[report header] ***** Output BEFORE the data table
    <html>
    <body>
    <form>
        Family name:
        <input name="family" value="#family#">
        <input type="submit" value=" Search ">
        <input type=hidden name="c" value="#c#">
    </form>
[end]

[SQL]  ***** Template for the SQL-query
    select id, F as "Family Name", I as "Name", O as "Patronymic"
    from test_persons
    where F like '#family#%' ??family
    order by f,i,o
[end]

[report footer]  ****** Output AFTER the data table
        </body>
    </html>
[end]
```

Family name: i [Search]

id	Family Name	Name	Patronymic
31	Iohang	Rostislav	
28	Itkis	Mikhail	Georgievich
27	Ivanchenko	Iosif	Moiseevich
26	Ivanov	Ivan	Ivanovich
29	Ivhold	Iohan	
30	Ivostwang	Imre	

Figure 2. Simplest module

2.5 JINR corporate web applications developed using WALT

WALT has been used to develop many JINR corporate web applications of various levels of complexity. The table below provides a list of them and approximate estimated labour costs in man-months for the development of the first working version. All these applications are currently in operation, some of them are evolving.

Table 1. JINR corporate web applications developed using WALT

Application	**Appointment**	**HR (man x months)**
ADB2	JINR's management accounting	4
PIN	Staff information in various aspects [9]	6
EDMS "Documents DB"	Electronic storage for administrative activity documents	6
NICA EVM	Project structure, workplans, expenses, Costbook, reports, etc.	4
EDMS "Dubna"	JINR electronic document management system [10]	12
HR JINR	Staff information in various aspects	4
Map JINR	Basic map of JINR's sites. Accessible for everyone online without restrictions	2
Gateway	Universal gateway for data exchange between various systems	2
ISSC	Scientific attestation support system	3
CERN DB	JINR's staff at CERN: trips, accommodation, reports, etc.	4
eco.jinr.ru	Structured list of JINR's corporate applications with distributed administration	0.5
EDMS "Advance reports"	Data preparation for business trip accounting reports	5
Checkpoint lists	Lists for access to JINR's sites in a limited access mode	0.25

It took only one week to twelve man-months to develop and launch a trial of the 1st version of the application. In addition to the above listed, more than a dozen applications have been developed using WALT outside of JINR. This confirms the effectiveness of WALT usage.

3. Conclusion

The presented platform with its very small code volume (~650 KB) has many advantages and can facilitate the development of sophisticated web applications by a small group of developers. Thanks to the number of available services and the concise and capacious template language, the developer can promptly perform complex tasks with a minimum amount of code. When very specific tasks arise, WALT allows the developer to expand or modify the functionality. WALT is compatible with client tools such as jQuery, AJAX technology, etc., enabling to use them for frontend development. The effectiveness of the platform is proved by many applications developed using it.

References

[1] Django. Available at: https://www.djangoproject.com/ (accessed 09.09.2021)

[2] What is ASP.NET Core. Available at: https://dotnet.microsoft.com/learn/aspnet/what-is-aspnet-core (accessed 09.09.2021)

[3] Express. Available at: http://expressjs.com/ (accessed 09.09.2021)

[4] Angular: The modern web developer's platform. Available at: https://angular.io/ (accessed 09.09.2021)

[5] Beck K., Grenning J., Martin R. C., Beedl M., Highsmith J., Mellor S., Bennekum A., Hunt A., Schwaber K., Cockburn A., Jeffries R., Sutherland J., Cunningham W, Kern J., Thomas D., Fowler M., Marick B. "Manifesto for Agile Software Development" // Agile Alliance. Retrieved 14 June 2010. Available at: https://agilemanifesto.org/

[6] Java. Available at: https://www.java.com/ (accessed 10.09.2021)

[7] Java Servlet Technology. Available at: https://www.oracle.com/java/technologies/servlet-technology.html (accessed 21.09.2021)

[8] Apache Tomcat. Available at: https://tomcat.apache.org/ (accessed 10.09.2021)

[9] Borisovsky V.F., Korenkov V.V., Kuniaev S.V., Lenskaja, N.A., Musulmanbekov G.G., Nikonov E.G., Filozova I.A. Scientific and organizational digital resources of the Joint Institute for Nuclear Research // Digital Libraries: Advanced Methods and Technologies, Proceedings of the RCDL 2008. P. 227. Available at: http://rcdl2008.jinr.ru/pdf/277_283_paper33.pdf (accessed 21.09.2021)

[10] Alexandrov I.N., Belyakova O.V., Korenkov V.V., Kuniaev S.V., Pechnikova L.N., Plyashkevich M.S., Semashko S.V., Trubnikov G.V., Ustenko P.V., Chikhalina S.N., Yakovlev A. Development and implementation of electronic document management system "EDMS Dubna" at JINR // CEUR Workshop Proceedings. 2016.Vol. 1787. P. 92