

HTTPEXPLORER: Exploring The Hypertext Transfer Protocol

Volker Turau^{*}
Technische Universität Hamburg-Harburg
Schwarzenbergstraße 95
Hamburg, Germany
turau@tuhh.de

ABSTRACT

This paper presents HTTPEXPLORER, an interactive tool to explore the Hypertext Transfer Protocol. The intention is to use the tool in a course on web-based applications to support the learning of HTTP, the most significant protocol used on the internet today. A web-based user-interface allows students to contact any HTTP-Server connected to the internet, to issue a request and to make the data flow between a client and the server completely visible. The tool can be used by novices to get first experience with HTTP and by advanced users to experiment with more complex features. We also report about some initial experience gained in the usage of HTTPEXPLORER in a real course.

Categories and Subject Descriptors

H.5.2 [User Interfaces]: Training, help, and documentation; C.2.2 [Network Protocols]: Applications—*HTTP*; K.3.1 [Computer Uses in Education]: Computer-assisted instruction (CAI)

General Terms

Algorithms, experimentation, design

Keywords

HTTP, simulator, WWW, e-learning

1. INTRODUCTION

A wide range of new, complex distributed applications is emerging in the Web environment because of the popularity and ubiquity of the Web itself and the nature of its features. The web provides an information representation that supports interlinking of all kinds of content from different sources and easy access for end-users with widely

^{*}This work was supported by the German Ministry of Education and Research (BMBF).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ITICSE'03, June 30–July 2, 2003, Thessaloniki, Greece.
Copyright 2003 ACM 1-58113-672-2/03/0006 ...\$5.00.

available tools. The successful development, deployment, operation and maintenance of such applications requires a deep understanding of the underlying protocol HTTP. Misconceptions about cache control and transport encoding result in applications with poor responsiveness. Acquaintance with content and language negotiation, chunked-encoding and byte-ranges will lead to a smarter usage of the web. Familiarity with cookies is a prerequisite for personalized and session-aware applications, which are ubiquitous in commercial web-based systems.

To facilitate the acquirement of knowledge about HTTP an interactive tool called HTTPEXPLORER was developed. The tool allows to experiment with HTTP using any web-server connected to the internet. It makes the data flow between a client and a server completely visible. Compared to a HTTP-Simulator using an implementation which is fully compliant with HTTP as defined in RFC 2616, this approach has the advantage that it provides experience with real implementations of HTTP and that it allows to see the differences between individual server implementations [1]. The tool also enables users to see the reaction of HTTP-Servers when confronted with syntactically or semantically ill formed requests or incomplete requests. Two types of usage are supported:

- requests can be provided using plain text in an input form and are henceforth submitted without any changes
- requests can be assembled using helper menus based on previous requests and responses. This allows the quick assembly of complex requests and reduces the possibility of syntactic errors

The two modes of usage are suitable for naive respectively advanced users. The tool is intended as a supplementary aid to a course on HTTP, i.e. it is not a self contained learning unit on HTTP. It comes with a suite of exercises to be used in conjunction.

The presented work is part of the research project Wissenswerkstatt Rechensysteme (Knowledge Factory for Computing Systems), a joint project of twelve German universities [4, 3]. The tool integrates well with the learning material developed in this project. This is achieved by hyperlinks combining entities of HTTPEXPLORER with corresponding sections of the learning material about HTTP.

The following sections describe the features and implementation of the tool and its integration into a learning environment.

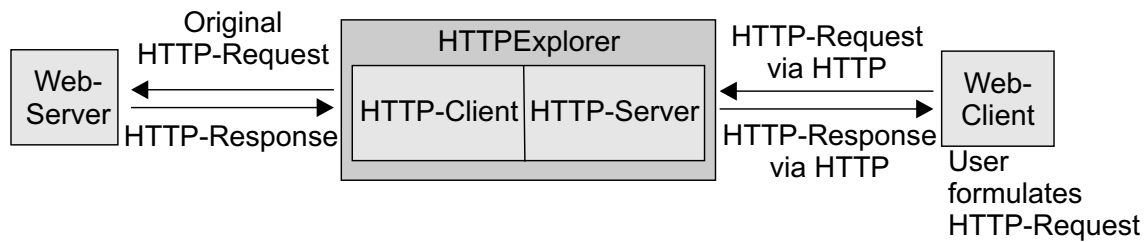


Figure 1: The architecture of HTTPExplorer

2. THE TOOL

HTTPExplorer has a web-front end which allows the assembly of HTTP-requests in several ways. When a user submits a request to HTTPExplorer, the assembled HTTP request is transported via the HTTP method POST (in a piggyback style). Figure 1 depicts the overall architecture of HTTPExplorer and the data flow. The tool consists of two components: towards the client it works like a web-server and towards the internet it has the role of a web-client. The server component receives a request and constructs a real HTTP-request from the content body. The request is handed over to the client component which submits it to the target server. After receiving a response the HTTP headers and the content body are forwarded to the server component. Then an answer document with this information is generated and finally forwarded to the user. Users behind a firewall can configure HTTPExplorer to use a proxy. The usage of a proxy is completely transparent.

The user interface consists of a set of window-like structures in a single web page. These windows can be minimized and maximized at any time (similar to windows in graphical user interfaces). Users should close windows which are currently not of importance in order to avoid scrolling. The most important windows are:

- **Main Window:**
This is the central data entry window of the application. It allows to formulate HTTP requests manually. This window has input fields for the url, the HTTP-method and the HTTP-version. Based on these values HTTPExplorer can generate a minimal request (with respect to the selected version of HTTP) to retrieve the document. Users can add additional request headers manually and finally execute the request. There is a text-field for the display of error-messages with respect to the general operation of the tool (e.g. time-out of a network connection).

This window also contains sections for displaying the request and the response. There is a raw and a parsed view available for each of these sections. The raw view displays the request or response as communicated by HTTP without any changes. The parsed view mainly includes hyper-links into the learning material (see section 4). And finally this window allows to see the requested document in three different views:

- The *raw view* displays the content as transported by HTTP.
- The *plain view* shows the content after the transfer encoding (such as *chunked*) has been decoded.

- The *live view* shows the content after the content encoding (such as *zip*) has been decoded.

Figure 2 shows a screenshot of the main window.

- **Cookie Window:**
This window displays a list with the content of all cookies received in the current session in tabular form. Each cookie can be added to the current request.
- **Quick Header:**
A menu with all headers defined in RFC 2616 is available, users must enter manually the value for a header. Neither the syntax nor the semantics of the values are checked.
- **Header assembler:**
This window supports the creation of individual HTTP headers. The headers are grouped into seven categories according to the format of their values. The intention of this window to reduce the possibility of syntactical errors in header values. Whenever possible, menus with all legal values are provided.

Other windows include proxy settings, authentication information and a request history.

3. SUPPORT FOR DIFFERENT USER LEVELS

HTTPExplorer can be tailored for different user groups. The configuration of the tool allows to specify

- the available windows and their states
- the range of HTTP-methods which can be executed
- details about proxy servers and authorization

The configuration is done by an administrator and not by the learner. The information is stored on the server. This flexibility makes it possible to have a rather simple and limited version for beginners. This makes the first encounter with the tool very easy and students are not overwhelmed by a complex user interface and do not have to configure the tool. At this stage learners basically enter requests manually. To avoid typing errors at that point a history with all previously submitted requests can be made available. Students can select any previously formulated query as a basis for a new query. After changing, deleting or adding new headers the query can be submitted.

For more advanced users there are several helper applications available:

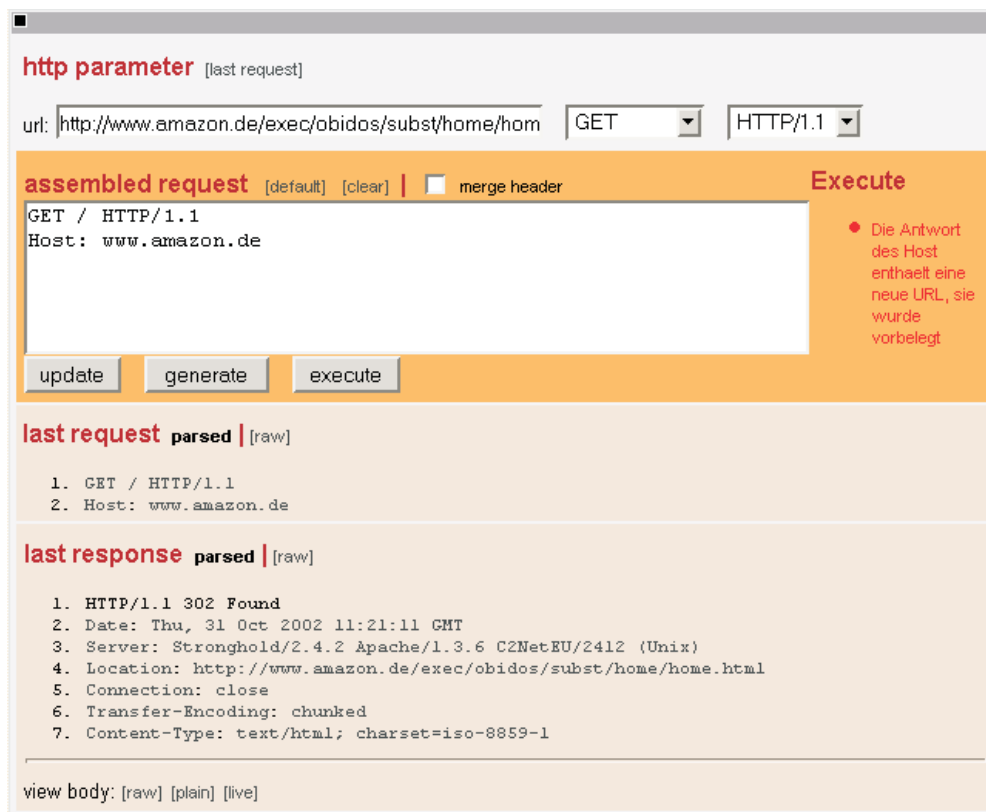


Figure 2: The main window of HTTPExplorer

- menus with all HTTP-methods and headers as defined in RFC 2616
- header-dependent menus to generate legal values (see Figure 4 for an example of headers with media type and date values)
- a list with all cookies received so far
- fields for request parameters

Using these applications learners can quickly experiment with a specific feature of HTTP without wasting time with correcting syntactical errors.

4. INTEGRATION INTO A LEARNING ENVIRONMENT

HTTPExplorer can be used as a standalone tool, but it also can be integrated into the learning environment developed in the WWR project. The main benefit of this integration are the hyperlinks from the tool into the text material. These links are provided at several places: error-messages, response and request messages etc. These hyper-links form a bridge between the tool and the text-material. In order to review the material about a term appearing in a response of HTTPExplorer users must only select the corresponding link. For this purpose, the tool provides two views of the response of a HTTP request. The raw view displays the response in the original format, as transported by HTTP. The parsed view shows the response enriched with hyperlinks into the defining sections of the text material. Figure 4

shows the parsed view of a response, the user has selected the value of the header *Date*. The corresponding hyperlink points to a section about date formats that can be used in HTTP.

This tight integration of the tool and the text material proved to be very valuable. To achieve a high degree of portability the implementation uses only logical destinations in the links. A mapping of these logical destinations to real destinations in the learning material is defined in a configuration file. This separation allows the usage of HTTPExplorer in combination with different web-based learning environments.

5. EXPERIENCE

HTTPExplorer has been used in a course on web-based applications. After one initial lecture on HTTP and a short introduction to the tool itself the students had the opportunity to use the tool to see the protocol in action. Without the set of accompanying exercises students aimlessly played with the tool, exploring mainly the features of the tool itself. Only when asked to solve the exercises the tool proved to be very valuable. Students quickly became acquainted with the tool. Within a short period of time, students gained a deep insight into the workings and features of HTTP. This knowledge enabled them to quickly learn the Servlet API and to build efficient and sophisticated web-applications. At a later stage students used the tool to test their own web-based applications.

In order to allow the usage of HTTPExplorer independently of a connection to the internet, we provide a distri-

Figure 3: Menu to generate values for fields with media types and date type according to RFC 1123

```

last response parsed | [raw]
1. HTTP/1.1 200 OK
2. Content-Type: text/html; charset=ISO-8859-1
3. Set-Cookie: JSESSIONID=F11B666CA5B3DCD541CA2FA16FB42225; Path=/
4. Transfer-Encoding: chunked
5. Date: Fri, 25 Oct 2002 09:36:15 GMT
6. Server: Apache Coyote/1.0

```

Figure 4: User has selected a link leading to a section about date formats

bution of HTTP Explorer including the Servlet-engine Tomcat together with a sample application. This application includes amongst others static and dynamically generated pages. Learners have to install the server on their machine, the only requirement is a Java environment. They can run HTTP Explorer using Tomcat and at the same time explore the HTTP-implementation of Tomcat with the help of the sample application. They can also use any locally available web-server. The performance of the tool is satisfactory.

6. IMPLEMENTATION

HTTP Explorer is implemented using Servlets and Java Server pages. The implementation follows the Model-View-Controller design pattern. To support the generation of the views a tag library with more than 25 different tags was developed. The implementation also uses tag libraries developed in the Struts project [2]. Support for localization to various languages is provided, currently German and English versions are available.

The core of HTTP Explorer is the HTTPClient component. It is based on the package *HTTPClient* from the Jakarta Commons project [2]. In order to use this package for our purpose significant changes had to be made. The main reason is that we want to allow ill-formed requests. Also we had to prevent many automatic mechanisms which are useful for normal HTTP clients (such as automatically sending cookies and authentication information or automatically following redirections). The complete tool is distributed as a single WAR-file which can easily be deployed on any Servlet-engine supporting Servlet version 2.3 (e.g. using drag and drop).

7. EXPLORING OTHER PROTOCOLS

After the successful usage of HTTP Explorer we plan to apply the main idea to other protocols and applications as well. There are basically two requirements for a protocol or an application, it must

- be text-based and
- have a request-response characteristic.

These requirements have their roots in the characteristics of HTTP. Possible candidates for protocols include WebDAV

and SOAP, but it is also possible to access applications such as compilers or simulators. In this case the communication between the tool and the target application is either based on an API directly used by our tool or based on a network protocol. If there is no such protocol available a simple wrapper based on TCP/IP can provide this service. Our goal is to define an API, such that all learning tools can use a common kernel. Also we want to develop a common testing environment.

8. CONCLUSION

In this paper we have presented HTTP Explorer, a tool to support the learning of the Hypertext Transfer Protocol as defined in RFC 2616. The tool has a web-interface and can be tailored to different user groups. Embedded hyperlinks into corresponding learning materials allow the usage of HTTP Explorer with a learning environment. The tool has been successfully used in a course on web-based applications.

9. ACKNOWLEDGMENTS

The author would like to thank Krister Saleck for implementing the main core of HTTP Explorer. The source code of HTTP Explorer will be available on demand.

10. REFERENCES

- [1] Fielding, R. et. al, Hypertext Transfer Protocol – HTTP/1.1, Request for Comments: 2616, Internet Engineering Task Force.
- [2] The Jakarta Project, <http://jakarta.apache.org/>.
- [3] Kalfa, W., Kröger, R. and Köhler, F., Integrating Simulators and Real-Life Experiments into an XML-based Teaching and Learning Platform, Educational Multimedia, Denver/Colorado, June 2002.
- [4] Lucke, U. and Tavangarian, D., Turning a Current Trend into a Valuable Instrument: Multidimensional Educational Multimedia based on XML, Educational Multimedia, Denver/Colorado, June 2002.