



UNIVERSITÀ DI PARMA

Relazione progetto telematica
Sistema di diagnostica HTTP

Vincenzo Fraello

August 4, 2021

Contents

1	Tecnologie e strumenti utilizzati	3
1.1	Versione JAVA	3
1.2	Versione Web-Application	3
1.3	Strumenti comuni alle due versioni	6
2	Configurazione server	8
3	Interazione Client-Server	9
4	Risultati delle simulazioni	10
	Bibliografia	11

Obiettivi del progetto

L'obiettivo del progetto è quello di realizzare uno strumento di diagnostica per generare richieste personalizzate in modo da esaminare le risposte di un certo server HTTP. I messaggi HTTP devono potere essere costruiti in modo preciso, agendo su ogni possibile header field.

Dopo aver raggiunto lo scopo principale, il progetto è stato esteso aggiungendo le seguenti funzionalità:

- i. I messaggi HTTP possono essere costruiti per intero (non solo headers): status-line, header, body;
- ii. Configurazione di Apache HTTP Server tramite direttive nei files *.conf* e *.htaccess*;
- iii. Creazione di pagine web dinamiche e Web-Applications.

1 Tecnologie e strumenti utilizzati

Lo sviluppo del sistema di diagnostica è avvenuto su un PC con sistema operativo *Windows 10*.

Il sistema è stato realizzato in due versioni: (i) programma JAVA; (ii) web-application.

La duplice realizzazione del progetto, è dovuta al fatto che l'*API* che consente di realizzare il sistema come web-application, impone delle limitazioni sulla manipolazione di alcuni headers. L'*API* JAVA non prevede alcuna limitazione. Per contro, la realizzazione della *GUI* di una web-application è più rapida.

1.1 Versione JAVA

Le principali tecnologie e strumenti di sviluppo:

- L'*API HttpClient* è stato aggiunto in Java 11. Può essere utilizzato per richiedere risorse HTTP sulla rete. Supporta HTTP/1.1 e HTTP/2, modelli di programmazione sia sincroni che asincroni, gestisce i corpi di richiesta e risposta [1]. Un oggetto *HttpClient* permette di inviare richieste ai server personalizzando gli header e il body dei messaggi HTTP;
- Ambiente di sviluppo: Eclipse;

1.2 Versione Web-Application

Le principali tecnologie e strumenti di sviluppo:

- L'*API XMLHttpRequest* consente di creare oggetti (XHR) che vengono utilizzati per interagire con i server. Si possono recuperare i dati da un URL senza dover aggiornare l'intera pagina. Ciò consente a una pagina Web di aggiornare solo una parte di una pagina senza interrompere ciò che sta facendo l'utente. *XMLHttpRequest* è usato pesantemente

nella programmazione AJAX [2]. Tra le altre funzionalità, consente di manipolare alcuni header e il contenuto del body dei messaggi HTTP;

- Ambiente di sviluppo: Visual Studio Code;
- In conformità con l'architettura *MVC* adottata nel caso di Web-Application, le tecnologie utilizzate sono state:
 - *Model*: JSON, XML;
 - *View*: Framework Bootstrap, HTML, CSS;
 - *Controller*: PHP, JavaScript, JQuery;
- JSON è un formato leggero adatto all'interscambio di dati fra applicazioni client-server [3]. I dati possono essere rappresentati in maniera più compatta ed efficiente in termini di spreco di banda e memoria;
- L'eXtensible Markup Language (XML), è un metalinguaggio per la definizione di linguaggi di markup, ovvero un linguaggio basato su un meccanismo sintattico che consente di definire e controllare il significato degli elementi contenuti in un documento o in un testo [4]. Questo linguaggio può essere usato per la rappresentazione e scambio di dati;
- L'*HyperText Markup Language*, comunemente noto con l'acronimo HTML è un linguaggio di markup. Nato per la formattazione e impaginazione di documenti ipertestuali [5]. Questo linguaggio è lo standard utilizzato per la definizione della struttura di una web-application;
- Il *Cascading Style Sheets* (CSS) è un linguaggio usato per definire la formattazione di documenti HTML [6];
- Bootstrap è una raccolta di strumenti liberi per la creazione di siti e applicazioni per il Web [...]. Bootstrap è stato sviluppato da Mark Otto e Jacob Thornton presso Twitter come un framework che uniformasse i vari componenti che ne realizzavano l'interfaccia web, dato che la

presenza di diverse librerie aveva portato ad incoerenze ed elevati oneri di manutenzione [7];

- JavaScript è un linguaggio di programmazione orientato agli oggetti e agli eventi, comunemente utilizzato nella programmazione Web lato client [8]. Questo linguaggio è lo standard utilizzato per realizzare il Controller delle web-application;
- JQuery è una libreria JavaScript per applicazioni web [9]. "jQuery is a lightweight, 'write less, do more', JavaScript library. The purpose of jQuery is to make it much easier to use JavaScript." – W3Schools;
- Google Chrome è il browser utilizzato per l'esecuzione e test della web-application.

Limitazioni *XMLHttpRequest*

Nella sezione 1.2 è stato detto che l'API *XMLHttpRequest* impone alcune limitazioni sulla manipolazione di alcuni headers. La documentazione [10] indica questi headers come *Forbidden header name*.

"A forbidden header name is the name of any HTTP header that cannot be modified programmatically; specifically, an HTTP request header name [...]. Modifying such headers is forbidden because the user agent retains full control over them."

Di seguito si riportano tali headers:

- Accept-Charset;
- Accept-Encoding;
- Access-Control-Request-Headers;
- Access-Control-Request-Method;
- Connection;
- Content-Length;

- Cookie;
- Cookie2;
- Date;
- DNT;
- Expect;
- Feature-Policy;
- Host;
- Keep-Alive;
- Origin;
- Proxy-;
- Sec-;
- Referer;
- TE;
- Trailer;
- Transfer-Encoding;
- Upgrade;
- Via.

1.3 Strumenti comuni alle due versioni

- Wireshark è lo strumento utilizzato per l'analisi del traffico HTTP nell'interfaccia di loopback;

- Apache HTTP Server è un software che realizza le funzioni di trasporto delle informazioni, di *internetwork* e di collegamento, ed ha il vantaggio di offrire funzioni di controllo per la sicurezza come quelle effettuate da un proxy [11];
- PHP linguaggio di scripting interpretato utilizzato per sviluppare applicazioni web lato server, ma può essere usato anche per scrivere script a riga di comando o applicazioni stand-alone con interfaccia grafica [12];
- Relational database management system: MySQL;
- Structured Query Language: linguaggio per operazioni *CRUD* nel database.

2 Configurazione server

Apache HTTP Server si configura tramite semplici files di testo; il file principale è *httpd.conf*. Le configurazioni si spezzano in files più piccoli e semplici da gestire che vengono inclusi nel file di configurazione principale per mezzo di direttive di inclusione.

Il server viene configurato inserendo le direttive di configurazione in questi files di configurazione. Una direttiva è una parola chiave seguita da una o più argomenti che ne impostano il valore.

Le Figure 1 e 2 mostrano esempi di configurazione.

```
RewriteEngine on
RewriteCond %{REQUEST_FILENAME} !-d
RewriteCond %{REQUEST_FILENAME} !-f
RewriteRule . routes.php [L]
```

Figure 1: Direttive di un file *.htaccess* per configurare *routes.php* come pagina index

```
<VirtualHost *:80>
    DocumentRoot "C:/xampp/htdocs/RESTful_services/public"
    ServerName bookstore
</VirtualHost>

<VirtualHost *:80>
    DocumentRoot "C:/xampp/htdocs/e-CommerceDApp"
    ServerName e-CommerceDApp
</VirtualHost>

<VirtualHost *:80>
    DocumentRoot "C:/xampp/htdocs/paginationTable/public"
    ServerName table
</VirtualHost>

<VirtualHost *:80>
    DocumentRoot "C:/xampp/htdocs"
    ServerName localhost
</VirtualHost>

<VirtualHost *:80>
    DocumentRoot "C:/xampp/htdocs/Telematica/progettoTelematica_test"
    ServerName telematica
</VirtualHost>

<VirtualHost *:80>
    DocumentRoot "C:/xampp/htdocs/paginationTable"
    ServerName table1
</VirtualHost>
```

Figure 2: Direttive del file *httpd-vhosts.conf* per configurare il server in virtual-host

```

# localhost name resolution is handled within DNS itself.
#       127.0.0.1       localhost
#       ::1             localhost

127.0.0.1       bookstore
127.0.0.1       e-CommerceDApp
127.0.0.1       table
127.0.0.1       telematica
127.0.0.1       table1

```

Figure 3: Configurazione del file *hosts* interrogato prima del sistema DNS

3 Interazione Client-Server

Il paradigma di interazione utilizzato è quello Client-Server asincrono.

AJAX è una tecnica che permette di sviluppare velocemente pagine Web dinamiche.

Permette inoltre di aggiornare porzioni di pagine web senza dover ricaricare l'intera pagina in maniera asincrona. Il browser non rimane bloccato in attesa di ricevere la risposta dal server, ma continua a gestire altri eventi che si scatenano all'interno della pagina, o gestisce le risposte di altre richieste *AJAX* fatte al server.

La Figura 4 mostra il paradigma di interazione appena descritto per via grafica.

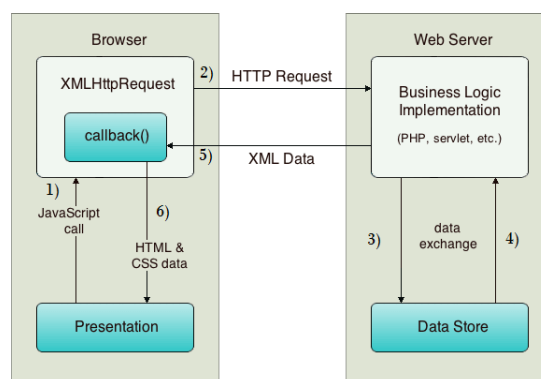


Figure 4: Paradigma di interazione asincrono

4 Risultati delle simulazioni

Il progetto realizzato è stato testato solo in parte. Una funzionalità immediatamente testabile – trattata durante il corso di Telematica – riguarda l’organizzazione dei web server in virtual-host. Infatti, è stato possibile osservare che modificando il contenuto dell’header host, vengono fornite risposte differenti.

Nel caso degli header di tipo *Accept* invece, non è stato possibile effettuare un test, in quanto la gestione della *Content-Negotiation* avviene lato server e non lato client. Occorrerebbe dunque programmare il server in maniera tale che possa gestire richieste che contengono tali headers¹.

Ulteriori test, caratterizzati da esiti positivi, sono stati condotti sul body dei messaggi HTTP.

¹Durante la ricerca di documentazione utile al progetto, sono emerse informazioni circa la gestione della content-negotiation. Quello riportato è un link ad un esempio di programmazione (in JAVA) di server in grado di gestire gli headers di tipo *Accept*. <https://www.ibm.com/docs/en/was-nd/9.0.5?topic=applications-implementing-content-negotiation-based-http-headers>

Bibliography

- [1] <https://openjdk.java.net/groups/net/httpclient/intro.html>.
- [2] Xmlhttprequest - web apis: Mdn. <https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest?retiredLocale=it>.
- [3] Introducing json. <https://www.json.org/json-en.html>.
- [4] Xml. <https://it.wikipedia.org/wiki/XML>, May 2021.
- [5] Html. <https://it.wikipedia.org/wiki/HTML>, Jun 2021.
- [6] Css. <https://it.wikipedia.org/wiki/CSS>, May 2021.
- [7] Bootstrap (framework). [https://it.wikipedia.org/wiki/Bootstrap_\(framework\)](https://it.wikipedia.org/wiki/Bootstrap_(framework)), Mar 2021.
- [8] Javascript. <https://it.wikipedia.org/wiki/JavaScript>, Jul 2021.
- [9] JQuery. <https://it.wikipedia.org/wiki/JQuery>, Mar 2021.
- [10] Forbidden header name - mdn web docs glossary: Definitions of web-related terms: Mdn. https://developer.mozilla.org/en-US/docs/Glossary/Forbidden_header_name.
- [11] Apache http server. https://it.wikipedia.org/wiki/Apache_HTTP_Server, Apr 2021.
- [12] Php. <https://it.wikipedia.org/wiki/PHP>, Jul 2021.