online Übungssystem

HTTP-Proxy mit Autorisierungsprüfung im Übungssystem

Autor:
Immo Schulz-Gerlach, ZMI,
FernUni-Hagen.de

Version:
0.3 – 05.09.2017

FernUniversität in Hagen

Inhaltsverzeichnis

Einleitung

Grundlegende Funktionalität

Das Online-Übungssystem bietet einen HTTP-Proxy-Service mit folgenden Eigenschaften:

- Authentifizierung von FernUni-Studenten oder -Mitarbeitern,
- Autorisierungsprüfung auf Basis einer Kursbelegung (bzw. Korrektor- oder Betreuer-Zuordnung zu einem Übungssystem-Kurs)
- Weiterleitung von HTTP(S)-Requests an Server im FernUni-Netzwerk, die selbst keine (Basic) Authentication verwenden.

D.h. es können HTTP-Requests, die eigentlich von einem Übungssystem-externen (FernUni-)Server bearbeitet werden sollen, durchs Übungssystem durchgeschleift werden, wobei das Übungssystem zunächst eine Authentifizierung des Aufrufers durchführt und den Request nur bei erfolgreicher Autorisierungsprüfung an den externen Server weiterleitet.

Haupteinsatzzweck ist also die Ausnutzung der bereits im Online-Übungssystem bestehenden Autorisierungfunktionalität für externe Dienste, die sich auf diese Weise nicht selbst um die Autorisierung ihrer Nutzer kümmern müssen. Damit externe Aufrufer diese Autorisierungsprüfung nicht einfach umgehen können, indem sie den Zielserver direkt statt übers Übungssystem ansprechen, sieht der typische Anwendungsfall vor, dass der anzusprechende Server nur innerhalb des FernUni-Netzwerkes und nicht direkt aus dem Internet erreichbar ist (z.B. durch Firewall-Regeln).

Zusätzlich übermittelt das Übungssystem noch Informationen über den authentifizierten Nutzer (Username oder Matrikelnummer) sowie den Autorisierungskontext (Übungssystem-Kurs, den der angemeldete Benutzer belegt hat bzw. betreut oder dessen Einsendearbeiten er korrigiert) an das Zielsystem, die dieses bei Bedarf berücksichtigen kann.

Einsatz-Szenario

Ein möglicher Anwendungsfall dieses Proxys lässt sich wie folgt skizzieren:

- Das Ausfüllen des Aufgabenformulars einer Aufgabe im Online-Übungssystem soll per „Ajax“ dynamisiert werden, d.h. auf Eingaben des Studenten hin soll, ohne die Eingaben bereits ans Übungssystem einzusenden, mit Änderungen am Formular reagiert werden.
    - Z.B. könnten abhängig von einer bestimmten Auswahl weitere spezifische Fragen nachgeladen werden.
    - Oder man könnte eine Art erstes Feedback zu einer Einsendung geben (ähnlich einer Vorkorrektur, aber ohne Einsendung und Speicherung der unvollständigen Eingaben).
    - Man könnte eine Art „Tipp“-Funktion realisieren, d.h. der Benutzer hat ein paar Schritte zur Lösung der Aufgabe eingegeben und kann sich Hilfe anfordern, wie er ausgehend von seinem Ansatz weiter zu verfahren hat, um der Lösung der Aufgabe näher zu kommen.
- Weiter sei angenommen, dass diese dynamischen Schritte nicht allein in JavaScript im Browser ausgeführt werden können, sondern die Mitarbeit einer serverseitigen Anwendung benötigen, die übers HTTP(S)-Protokoll (per „Ajax“, genauer: XMLHttpRequest-Object aka XHR) angesprochen werden soll.
- Diese serverseitige Software soll, z.B. aus lizenzrechtlichen Gründen, ihre Dienste nicht im gesamten Internet und ohne Authentifizierung anbieten, sondern nur für FernUni-Studenten

zugänglich sein. Genauer genügt es sogar, ihre Anwendung auf die Studenten einzuschränken, die überhaupt berechtigt sind, diese Aufgabe im Online-Übungssystem zu bearbeiten.

- Die Authentifizierung soll für die Studenten transparent erfolgen, d.h. sie sollen nicht – nachdem sie sich bereits im Übungssystem zum Aufrufen der Aufgabenseite anmelden müssen – erneut (ggf. andere) Logindaten für diesen interaktiven Hintergrundserver eingeben müssen.
- Die entsprechende Hintergrund-Software kann nicht (oder soll aus Gründen des hohen Aufwands nicht) selbst die Authentifizierung der Studenten mit ihren LDAP-Accounts und ggf. die Autorisierung über eine Kursbelegungs-Prüfung vornehmen.

In diesem Beispiel kann der Übungssystem-Autorisierungs-Proxy im „Studenten-Modus“ zum Einsatz kommen:

- Die externe Software wird ohne jegliche Authentifizierung in Betrieb genommen, steht aber nur im FernUni-LAN und nicht im Internet zur Verfügung.
- Die Ajax-Requests werden nicht direkt an diesen Anwendungsserver gesendet, sondern an den Proxy-Service des Online-Übungssystems. Dabei wird im URL des Requests Bezug auf den Kurs genommen, in dessen Kontext die Aufgabe angeboten wird (so dass jeder, der die Aufgabe bearbeiten darf, weil er den zugehörigen Kurs belegt hat, auch diese Requests stellen darf).
- Die Antwort des Servers, die zurück ans Online-Übungssystem übermittelt wurde, wird von diesem an den aufrufenden Browser weitergeleitet.
- Der Browser sendet auch im Ajax-Request dieselben Credentials ans Übungssystem wie bei „normalen“ Browser-Requests, so dass die Authentifizierung / Autorisierung für den Studenten transparent erfolgt.

Abgrenzung zu SOAP-Vorkorrekturmodulen

Mit Vorkorrekturmodulen gibt es bereits einen Mechanismus, über den externe Software an das Online-Übungssystem angebunden werden kann. Die Anwendung von Vorkorrekturmodulen unterscheidet sich jedoch erheblich von der dieses Proxys:

- Die externe Software (Vorkorrekturmodul) muss ein bestimmtes SOAP-Interface implementieren.
- Sie lässt sich nicht direkt per Ajax von dynamischem Code in Aufgabenseiten ansprechen, sondern höchstens indirekt: Indem der Browser eine Einsendung für eine Teilaufgabe vornimmt (egal ob durch normalen Formular-Submit oder Ajax-Post-Request), bei der das Vorkorrekturmodul registriert ist.
    - Auch hierbei erfolgt eine Authentifizierung (Student muss einen Account bei der FernUni besitzen) und Autorisierung (Student muss den Kurs belegt haben).
    - Die Einsendung wird dann zunächst in der Datenbank des Übungssystems gespeichert.
    - Anschließend leitet das Übungssystem die eingesendeten Daten per SOAP an das Vorkorrekturmodul weiter.
    - Die Rückgabe des Vorkorrekturmoduls wird wiederum zunächst in der Datenbank gespeichert.
    - Dann wird eine Antwort an den Browser gesendet, die durch Ausfüllen einer Korrekturseitenvorlage (aka „Korrekturschablone“) – typischerweise, aber nicht notwendig, eine HTML-Datei – ermittelt wird. Dabei kann insbesondere die Vorkorrektur in diese Seite eingebunden werden (oder auch allein den gesamten Inhalt der Response darstellen).

Vorkorrekturmodule sind nicht mit Blick auf interaktive Aufgabenbearbeitung in mehreren Schritten

ausgelegt worden, sondern ihr primäres Einsatzgebiet ist das Feedback nach einer in gewissem Sinne „fertigen" Einsendung. Durch Speicherung der Vorkorrekturen sind die Ergebnisse auch nicht „flüchtig", sondern können auch später im Rahmen einer Korrektur immer wieder eingesehen werden (vom Studenten oder auch von einem menschlichen Korrektor, der eine endgültige Korrektur und Bewertung vornehmen soll).

Man kann zwar Vorkorrekturmodule auch für Ajax-Requests wie oben skizziert „zweckentfremden", das birgt aber gegenüber der Proxy-Lösung einige potentielle Nachteile. Insbesondere ist deutlich höherer Zeitbedarf für die Requestbearbeitung einzukalkulieren, bedingt durch Datenbankzugriffe, Bildung und Ausführung eines SOAP-Requests, Ausfüllen einer Korrektuschablone etc. Die Speicherung der Daten kann andererseits in gewissen Anwendungsfällen vielleicht auch ein Vorteil sein, weil so bei Unterbrechen einer interaktiven Aufgabenbearbeitung und später erneutem Aufruf ggf. ein Wiedereinstieg beim letzten Bearbeitungsschritt realisierbar ist.

Ist eine Speicherung der Teileinsendungen dagegen nicht nötig, ist die Verwendung des Proxys einfacher und schneller. Weiterhin erschließt der Proxy zusätzliche Einsatzbereiche, indem sein Autorisierungsmechanismus nicht auf Studenten/Kursbeleger eingeschränkt ist, sondern alternativ auch für Korrektoren oder Betreuer eines Übungssystem-Kurses eingesetzt werden kann. So wäre z.B. die Nutzung bestimmter Serveranwendungen für Korrektoren zu deren Unterstützung bei der Aufgabenkorrektur denkbar.

Einschränkungen

- Dieser Service ist zunächst absichtlich darauf eingeschränkt, lediglich Server im FernUni-Netz (`132.176.*` | `*.fernuni-hagen.de` | `*.feu.de`) anzusprechen. Er kann nicht als Proxy zum Zugriff auf externe Server genutzt werden.
    - Unter Umständen ist ein Zugriff des Übungssystem-Servers auf Ihren Zielserver nicht ohne explizite Firewall-Regel möglich. Wenn der Zugang also nicht auf Anhieb funktioniert, kontaktieren Sie diesbezüglich den Helpdesk.
- Weiterhin wird derzeit vorausgesetzt, dass die gesamte Authentifizierung und Autorisierung über diesen Proxy erledigt wird. Der Ziel-Server darf nicht seinerseits Basic Authentication verwenden[1], denn ihm werden niemals Authorization-Header übermittelt.
- Dieser Service ist primär auf einfachen Datenaustausch in einem einzigen Request ausgelegt (z.B. Aufrufe von SOAP- oder REST-WebServices). Für den Abruf von HTML-Webseiten per Browser ist er – anders als ein „echter" HTTP-Proxy-Server) nur eingeschränkt geeignet: Falls über den Proxy HTML-Seiten abgerufen werden, die ihrerseits das Nachladen von Ressourcen mit mit „/" beginnenden URLs (z.B. `/images/test.png` oder `/styles/main.css`) anfordern, so würde der Browser daraus Requests wie `https://online-uebungssystem.fernuni-hagen.de/images/test.png` generieren, die natürlich nicht über den Proxy abgewickelt würden, sondern in aller Regel schlicht fehlschlügen (404: Not found) oder falsche Daten lieferten.

Verwendung

Im Unterschied zu „gewöhnlichen“ HTTP-Proxy-Servern ist hier kein Request mit Original-URL lediglich an eine IP eines Proxy-Servers zu senden, sondern zunächst ein regulärer Requests ans Online-Übungssystem zu stellen, mit einem übungssystemspezifischen URL, aus dem die für die Autorisierungprüfung benötigten Informationen entnommen werden. Der Ziel-URL ist dem Übungssystem zusätzlich zu übermitteln. Der Request wird dann (leicht modifiziert) an den Zielserver weitergeleitet, sofern die Autorisierungsprüfung zuvor erfolgreich war.

URL

Der URL hat zunächst den Standard-Aufbau von URLs im Kurskontext mit einem speziellen Servicenamen und dem Schlüssel (Veranstaltername, Kursnummer, Versionsnummer/Semester) des Kurses, für den die Autorisierungsprüfung vorgenommen werden soll. Aus dem Servicenamen geht weiterhin die Art der Autorisierungsprüfung hervor (d.h. ob der Aufrufer ein Beleger, Korrektor oder Betreuer des Kurses sein muss, um den Dienst aufrufen zu dürfen). Der Ziel-URL, an den der Request weiterzuleiten ist, wird als Suffix an den URL angehängt.

Der URL hat also folgenden Aufbau (in RegEx-Syntax[2]):

```
https://online-uebungssystem.fernuni-hagen.de↵
/<veranstaltername>/(Student|Betreuer|Korrektor)?
AuthProxy/<kursnr>/<versionsnr>↵
/https?://<Ziel-Domain>:(<Port>)?(/<Path>)?(\?<QueryString>)?
```

Die Abschnitte in spitzen Klammern sind Platzhalter, die mit konkreten Werten zu füllen sind:

- *Kursschlüssel*: `veranstaltername`, `kursnr` und `versionsnr` identifizieren den Kurs, für den die Autorisierung verlangt wird.
- Das *Präfix des Servicenamens* (`Student`, `Betreuer` oder `Korrektor`) legt fest, was für ein Login-Typ erwartet wird.
  - Die Angabe `Student` schließt dabei auch Teststudenten und Mentoren mit ein.
  - Das Präfix ist optional, der Default ist `Student`. D.h. an Stelle von `StudentAuthProxy` kann auch kurz der Servicename `AuthProxy` verwendet werden.
- Nach der Versionsnr (z.B. `WS14`) folgt, durch Slash getrennt, der vom Proxy anzusprechende *Target-URL*.
  - Dieser muss mit `http://` oder `https://` beginnen,
  - gefolgt von einem Domain-Namen mit entweder der `fernuni-hagen.de`- oder `feu.de`-Domain oder alternativ eine IP-Adresse aus dem FernUni-Netz.
  - Wie bei allen URLs kann darauf optional eine Portnummer, ein Path und/oder ein Query-String (eingeleitet von einem Fragezeichen) folgen.

Beispiele:

- `https://online-uebungssystem.fernuni-hagen.de/six/AuthProxy/01613/WS10/http://abc.fernuni-hagen.de/?q=test`
- `https://online-uebungssystem.fernuni-hagen.de/six/BetreuerAuthProxy/01613/WS10/https://xyz.feu.de:50101/some/path`

HTTP-Methods

Unterstützt werden derzeit lediglich `GET`, `POST` und `PUT`.[3]

HTTP-Header

HTTP-Auth

Die Anfrage muss – wie alle Anfragen ans Online-Übungssystem – einen Authorization-Header für Basic-Authentication (aka HTTP-Auth) enthalten, andernfalls lehnt das Übungssystem die Anfrage mit Response Code 401 (Unauthorized) ab.
(Aufbau des Headers: `Authorization: Basic <base64(username:password)>`)

Nur wenn die so übergebenen Credentials einen Studenten, Korrektor bzw. Betreuer authentifizieren und dieser zum Zugriff auf den Kurs autorisiert ist, wird die Anfrage an die Remote-URL weitergeleitet, andernfalls antwortet der Übungssystem-Server entweder mit 401 oder 403 (abhängig davon, ob eine neue Passworteingabe angefordert werden soll oder ob die Credentials zwar einen User authentifizieren, dieser aber keine Autorisierung hat).

Der `Authorization`-Header wird nicht ans Zielsystem weitergeleitet, sondern vom Übungssystem aus dem Request entfernt.

X-Header

Statt des Authorization-Headers werden neue, nicht im HTTP-Standard definierte Header in den Request ans Zielsystem eingebunden:

- `X-Username`: Loginname des authentifizierten Benutzers. Bei Betreuern und Korrektoren ist dies immer der vollständige Loginname. Bei Studenten wird in der Regel ein Name der Art `q`+Matrikelnummer ausgegeben, wenn es sich um einen normalen Studenten handelt. Bei Mentoren oder Teststudenten wird direkt der lokale Loginname (z.B. `7777777` für den typischen Teststudenten mit Sonderrechten) ausgegeben.
- `X-Matrikelnr`: Dieser Header wird nur für Studenten-Accounts ausgegeben und enthält die (rein numerische) Matrikelnummer des Studenten. (Es wird nicht sichergestellt, dass es sich um einen echten FernUni-Studenten handelt, auch für Teststudenten kann dieser Header ausgegeben werden, wenn diese einen nur aus Ziffern bestehenden Loginnamen verwenden, nicht dagegen z.B. für Mentoren mit nicht-numerischem Benutzernamen.)
- `X-Veranstaltername`, `X-Kursnr` und `X-Versionsnr` enthalten den Kursschlüssel aus dem URL des Original-Requests ans Online-Übungssystem, s.o.

Sonstige Header

Die meisten Header werden ansonsten 1:1 ans Zielsystem weitergereicht, nur „Hop-by-Hop“-Header[4] werden ausgefiltert. Außerdem wird natürlich der Host-Header ausgetauscht gegen den Domainnamen (bzw. die IP) aus dem Target-URL.

HTTP-Body

Der Message Body sowohl von Requests (POST oder PUT) als auch von Responses (alle Methods) wird unverändert übertragen, also insbesondere vom Übungssystem weder interpretiert noch manipuliert.

Fußnoten

1. Bei Bedarf ließe sich ggf. eine Erweiterung implementieren, nach welcher bei Vorliegen von gesonderten Proxy-Authorization-Headern zusätzlich zu „normalen“ Authorization-Headern im Request das Übungssystem nur die Proxy-Authorization auswertet und Authorization-Header dann ans Zielsystem weiterleitet.
2. Syntax regulärer Ausdrücke. Insbesondere steht `\?` für ein Fragezeichen im URL, während `?` anzeigt, dass das vorhergehende einzelne Zeichen bzw. der Inhalt der vorhergehenden Klammern optional ist. Ein Pipe (`|`) ist ein Oder-Operator, Trennzeichen für eine Aufzählung von Alternativen (i.d.R. ebenfalls in runden Klammern).
3. Bei Bedarf lässt sich ggf. der Support auch für andere Methoden wie insb. `DELETE` nachrüsten.
4. Header, die laut HTTP/1.1-Standard nur an den nächsten, direkt angesprochenen Host übermittelt und insb. von Proxies nicht weitergeleitet werden. Siehe RFC2616 Sec. 13.5.1 <http://www.w3.org/Protocols/rfc2616/rfc2616-sec13.html#sec13.5.1>