Dennis Blöte

Single Sign-On mit OpenID

Ein Leitfaden zur OpenID-Integration in bestehende Websites

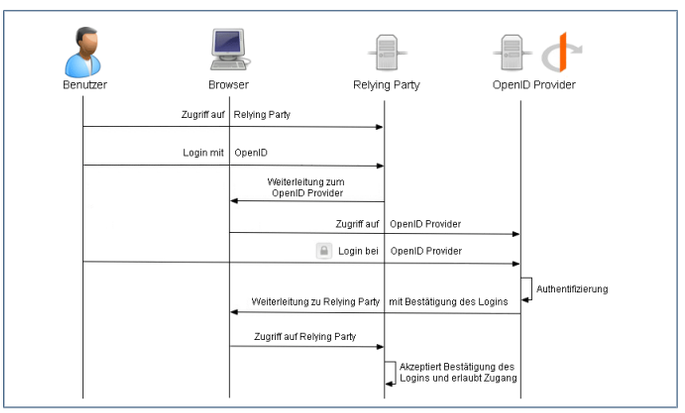
OpenID ist ein offener Standard, der es Nutzern ermöglicht, sich mit einer einzigen digitalen Identität auf verschiedensten Websites einzuloggen. Die persönliche OpenID dient dabei als Ersatz für die herkömmliche Anmeldung mittels Benutzername und Passwort und ermöglicht so einen komfortablen Login ohne vorherige Registrierung. Dieser Artikel zeigt die Integration von OpenID in bestehende Websites und beschreibt, was Entwickler dabei beachten sollten.

Immer mehr Webanwendungen fordern von ihren Besuchern eine Registrierung, damit sie das Angebot voll nutzen können. Anwender sehen sich daher gezwungen, eine kontinuierlich steigende Zahl von Benutzerkonten und Passwörtern zu verwalten. Mit OpenID ist mittlerweile jedoch eine interessante Alternative entstanden, die es dem Anwender ermöglicht, sich mit einer digitalen Identität auf verschiedensten Websites einzuloggen.

Die Akzeptanz von OpenID steigt zunehmend, da das Konzept sowohl für Nutzer als auch für Website-Betreiber gleichermaßen interessant ist: Der Anwender verwaltet seine Daten bei einem selbst gewählten Identitätsanbieter seines Vertrauens und übermittelt seine Informationen von dort aus den Websites, die er nutzen möchte. Auf diesen Websites ist kein Registrierungsvorgang mehr erforderlich, weil die Benutzerdaten vom jeweiligen Identitätsanbieter bezogen werden. In einer OpenID-basierten Identitätsverwaltung gibt es zwei Parteien: zum einen den Identity-Provider, bei dem die Daten des Nutzers liegen, zum anderen die Relying-Party, die auf die Daten zugreifen möchte.

Ein Identity-Provider dient der Bereitstellung und Verwaltung digitaler Identitäten. Bei so einem Identitätsanbieter handelt es sich um einen Dienst, der Benutzer authentifizieren kann und ihnen die Möglichkeit gibt, ihre Identitätsinformationen (Attribute) mit anderen Diensten auszutauschen.

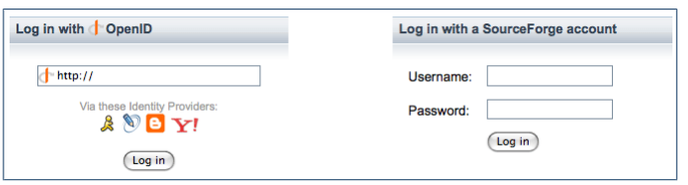
Relying-Parties (vertrauende Parteien) sind Dienste, die den Informationen eines Identity-Providers vertrauen und darauf basierend Zugriff auf die Anwendung erlauben. Dieser Artikel zeigt die Integration von OpenID in eine Relying-Party-Anwendung und soll als Hilfestellung für Website-Betreiber dienen, die ihren Nutzern den Login per OpenID ermöglichen wollen.



Der Ablauf des Authentifizierungsvorgangs mit OpenID.

Ablauf einer OpenID-Authentifizierung

Ausgangspunkt einer OpenID-Authentifizierung ist der Login des Benutzers auf der Website der Relying-Party. Im Gegensatz zur herkömmlichen Authentifizierung mittels Benutzername und Passwort befindet sich auf OpenID-unterstützenden Websites ein Eingabefeld für die OpenID-URL des Benutzers. Dieses Feld ist üblicherweise durch das OpenID-Symbol gekennzeichnet, um die Erkennung zu erleichtern und eine konsistente Benutzerführung zu gewährleisten.



Beispiel einer Loginseite, die sowohl die Anmeldung mit OpenID als auch mit Benutzername und Passwort ermöglicht.

Der Benutzer gibt seine OpenID-URL ein, daraufhin leitet ihn die Relying-Party zur eigentlichen Authentifizierung an den zuständigen Identity-Provider weiter. Nachdem der Benutzer sich bei seinem Identity-Provider authentifiziert hat, kann dieser die Anfrage der Relying-Party beantworten. Das Sequenzdiagramm verdeutlicht diesen Ablauf.

Austausch der Benutzerdaten

Zusätzlich zur Authentifizierungsbestätigung können auch Benutzerdaten ausgetauscht werden, sofern der Identity-Provider die Erweiterungen „Simple Registration“ oder „Attribute Exchange“ unterstützt. Der genaue Ablauf des Vorgangs ist von der Implementierung auf Seiten des Identity-Providers abhängig, läuft allerdings üblicherweise nach folgendem Muster ab: Der Benutzer wird nach dem Login nicht direkt an die Relying-Party weitergeleitet, sondern bekommt eine Seite präsentiert, auf der er die von der Relying-Party angeforderten Attribute sieht und entscheiden kann, welche er freigeben möchte. Im Anschluss daran wird der Benutzer wie beim normalen Authentifizierungsvorgang an die Relying-Party weitergeleitet. Dabei enthält die Antwort des Identity-Providers neben der Authentifizierungsbestätigung auch die vom Benutzer freigegebenen Informationen.

Simple Registration

Die Protokollerweiterung „Simple Registration“ (SReg) wurde entwickelt, um Profildaten des Benutzers zu übertragen und dadurch den Registrierungsvorgang bei einer Relying-Party zu überspringen. Bei der Authentifizierungsanfrage an den Identity-Provider kann die Relying-Party zusätzlich folgende acht Attribute anfragen: Benutzername, E-Mail-Adresse, Name, Geburtsdatum, Geschlecht, Postleitzahl, Land, Zeitzone und Sprache. Dabei wird außerdem festgelegt, welche der Attribute für eine Registrierung des Benutzers erforderlich und welche optional sind.

Der Nachteil von SReg ist, dass es sich bei den acht Attributen um eine fixe Menge handelt, die nicht erweitert werden kann. Somit können beispielsweise nicht Wohnort, Telefonnummer oder URL der Website des Benutzers erfragt werden. SReg eignet sich

daher nur für den Austausch rudimentärer Benutzerdaten. Vorteil hingegen ist, dass SReg von vielen Identity-Providern unterstützt wird, da es schon sehr früh Bestandteil der OpenID-Entwicklung war. Mittlerweile ist „Simple Registration" jedoch theoretisch veraltet, da es mit „Attribute Exchange" eine viel flexiblere Lösung zum Austausch von Attributen gibt.

Attribute Exchange

Attribute Exchange (AX) definiert im Gegensatz zu SReg kein fixes Set von Attributen, sondern stellt einen Namensraum dar, in dem eigene Attribute definiert werden können. Ein Attribut besteht dabei aus den vier Komponenten „Type Identifier" (URL als Definition), „Title" (Bezeichner für den Endnutzer), „Count" (Anzahl der gewünschten Werte) und „Value" (eigentlicher Wert). Das Attribut E-Mail-Adresse eines Benutzers kann sich beispielsweise folgendermaßen zusammensetzen:

- Type Identifier: http://axschema.org/contact/email
- Title: E-Mail
- Count: 1
- Value: meine@emailadresse.de

Voraussetzung für den Attributaustausch ist, dass Identity-Provider und Relying-Party das gleiche Set an Attributen unterstützen. Ausschlaggebend dabei ist der Type Identifier: Die URL dient als Bezeichner und legt fest, um was für ein Attribut es sich genau handelt. Auf AXSchema.org findet sich eine Liste mit bislang definierten Type Identifier, unter denen sich auch solche für die in Simple Registration definierten Attribute befinden. Da „Attribute Exchange" erst im Dezember 2007 spezifiziert wurde, unterstützen es bislang leider nur wenige Identity-Provider.

Integration von OpenID

Es wird vorausgesetzt, dass die Website schon eine Benutzerverwaltung besitzt und den Nutzern zusätzlich die Option zum Login per OpenID geboten werden soll. Der Großteil dieser Funktionalität kann mittels OpenID-Bibliotheken [1], die bereits für eine Vielzahl von Programmiersprachen vorhanden sind, abgedeckt werden. Diese Bibliotheken implementieren die Kommunikation zwischen Relying-Party und Identity-Provider, sodass man bei der Integration lediglich festlegen muss, welche Daten angefragt und wie sie verarbeitet werden sollen.

Bevor wir mit der Integration beginnen können, muss das Datenschema der Anwendung erweitert werden: Es wird eine zusätzliche Datenbanktabelle benötigt, in der die Zuordnung von Benutzer zu OpenID festgehalten wird. Es empfiehlt sich, dass jeder Benutzer mehrere OpenIDs angeben kann, damit man sich auch nach einem Wechsel des Identity-Providers mit einer neuen OpenID einloggen kann. Die Tabelle speichert die ID des Benutzers und die dazugehörige OpenID:

MySQL

```
CREATE TABLE openids (
  user_id int(11) NOT NULL,
  openid_url varchar(255) NOT NULL,
  KEY index_user_id (user_id),
  UNIQUE KEY unique_openid_url (openid_url)
);
```

Listing 1

Neben dem normalen Login mit Benutzername und Passwort wird ein weiteres Formular für die OpenID des Benutzers benötigt. Das OpenID-Textfeld wird üblicherweise mit dem OpenID-Logo gekennzeichnet und mit der HTML-ID „openid_identifier" versehen, damit es gegebenenfalls von Plugins erkannt werden kann.

CSS

```
#openid_identifier {
  background: #FFFFFF url('/images/openid-icon-small.gif') no-repeat 0 50%;
  padding-left: 20px;
}
```

Listing 2

Das Loginformular zielt auf das Skript, das die Authentifizierungsanfrage einleitet. In der Beispielimplementierung ist dies die start-Action im OpenID-Controller.

Beim Speichern der OpenID-URL sollten Entwickler unbedingt darauf achten, die vom Identity-Provider zurückgelieferte OpenID in normalisierter Form abzuspeichern. Durch die Normalisierung werden die URLs so umgewandelt, dass sie immer einem bestimmten Schema entsprechen, dies erleichtert das Abgleichen beziehungsweise Auffinden von OpenIDs bei einem späteren Login. Beispielsweise wird die URL in Kleinschreibung umgewandelt und etwaige Slashes werden hinzugefügt oder entfernt [2]. Die meisten Bibliotheken verfügen über eine Normalisierungsfunktion beziehungsweise liefern die OpenID-URL bereits normalisiert zurück.

Auch bereits existierende Benutzerkonten lassen sich nachträglich mit einer OpenID verknüpfen. Dazu muss der Benutzer auf seiner Profilseite die Möglichkeit bekommen, eine OpenID anzugeben: Fügt er eine OpenID hinzu, so muss diese mit einer Anfrage an den Identity-Provider verifiziert werden. Dieser Vorgang lässt sich wie die Authentifizierung durchführen, wobei zu beachten ist, dass die vom Benutzer angegebene OpenID erst nach der Bestätigung durch den Identity-Provider gespeichert werden darf. Ebenso muss der Benutzer die Möglichkeit bekommen, die mit seinem Benutzerkonto verknüpften OpenIDs zu löschen.

Beispielimplementierung

Auf der Heft-CD befinden sich zwei Beispielimplementierungen mit Ruby on Rails. Das erste Beispiel zeigt die Implementierung einer Authentifizierung über OpenID, bei der gleichzeitig auch Attribute per SReg und AX angefragt werden. Im zweiten Beispiel wird gezeigt, wie Anwender nachträglich ihr bestehendes Benutzerkonto mit einer OpenID verknüpfen können.

Fazit

Durch die Integration von OpenID können Website-Betreiber ihren Nutzern einen einfachen Einstieg ohne Registrierungsprozess bieten. Da die Authentifizierung zum jeweiligen Identity-Provider des Benutzers ausgelagert wird, sind Relying-Party-Anwendungen jedoch darauf angewiesen, dass die Identity-Provider der Benutzer geeignete Sicherheitsanforderungen erfüllen.

Links und Literatur

Softlink 2182

[1] OpenID-Bibliotheken: http://wiki.openid.net/Libraries
[2] OpenID Normalization: http://openid.net/specs/openid-authentication-2_0.html#normalization_example

DER AUTOR

Dennis Blöte ist Software-Entwickler aus Bremen. Im Rahmen seiner Bachelor-Thesis hat er für die Universität Bremen eine Single-Sign-On-Infrastruktur mit OpenID entwickelt und die Identity-Provider-Komponente als Open-Source-Software veröffentlicht. In seinem Blog (http://dennisbloete.de) schreibt er u.a. über Webtechnologien wie OpenID und Ruby on Rails.