



BIKE-SHARING WEBSERVICE MIT CLIENT-APPLIKATION

Tobias Hafermalz, Philipp Thöricht

Dresden, 28.01.2014

Inhalt

Aufgabe

Vorgehen

Funktionen des Webservice

Webclient

Implementierung des OAuth2-Servers

Client-Anwendung

Fazit

Quellen

Aufgabe

- Erstellung eines REST-Webservices
 - Implementierung in PHP mit Slim
 - Authentifizierung durch OAuth2
 - Verschlüsselung durch HTTPS
- Erstellung einer Webanwendung als Client
- Szenario: Bikesharing

Vorgehen

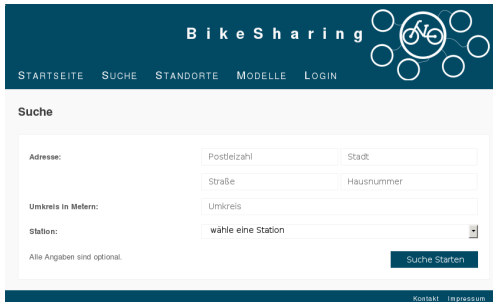
- public API implementiert
- Client (Webanwendung) implementiert
- OAuth2 Server implementiert
- protected APIs implementiert
- Client-Anwendung vervollständigt
- Bugfixes

Funktionen des Webservice

Name	Method	URL	Access
Alle verfügbare Fahrradstationen	GET	/stations	public
Spezielle Station	GET	/stations/stationID	public
Alle verfügbaren Fahrräder	GET	/bikes	public
Spezielles Fahrrad	GET	/bikes/bikesID	public
Alle Fahrradmodelle	GET	/models	public
Spezielles Fahrradmodell	GET	/models/modelID	public
Alle Buchungen	GET	/bookings	protected
Buchung erstellen	POST	/bookings	protected
Einzelne Buchung	GET	/bookings/bookingID	protected
Einzelne Buchung stornieren	DELETE	/bookings/bookingID	protected
Einzelne Buchung bearbeiten	PUT	/bookings/bookingID	protected
Accountinformationen	GET	/account	protected

Webclient

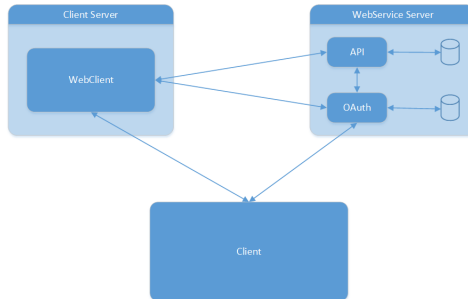
- erst funktionabel, dann schön
- Nutzung von Bibliotheken (jQuery, OpenLayers)
- Gerüst mit PHP erstellt, Inhalt mit Javascript gefüllt
- ansatzweise auch für mobile Endgeräte nutzbar



The screenshot shows a web client for a bike-sharing service. The header is dark blue with the text "B i k e S h a r i n g" in white, where the letters are spaced out. To the right of the text is a logo consisting of a bicycle icon inside a circle, surrounded by several smaller empty circles. Below the header is a navigation bar with the links: STARTSEITE, SUCHE, STANDORTE, MODELLE, and LOGIN. The main content area is titled "Suche" (Search). It contains a search form with the following fields: "Adresse:" (Address), "Postleitzahl" (Postal code), "Stadt" (City), "Straße" (Street), "Hausnummer" (House number), "Umkreis in Metern:" (Radius in meters), "Umkreis" (Radius), "Station:" (Station), and "wähle eine Station" (Select a station). Below the "Station:" field, there is a note: "Alle Angaben sind optional." (All data is optional). A blue button labeled "Suche Starten" (Start Search) is located at the bottom right of the search form. At the bottom of the page, there is a dark blue footer with the links "Kontakt" and "Impressum".

Implementierung des OAuth2-Servers

- Verwendung der „OAuth2 Server Library for PHP“



Client-Anwendung

Demo

Fazit

- Implementierung des Webclient nach Vorlage einer durchdachten API gut machbar
- Das Slim-Framework war eine gute Wahl, da die Verwendung sehr einfach und fehlerfrei verlief
 - OAuth-Middleware hat leider nicht funktioniert
- Implementierung eines OAuth-Servers ist relativ kompliziert

Quellen

- `slimframework.com`
- `jquery.com`
- `openlayers.org`
- `https://github.com/bshafter/oauth2-server-php`