



Bike-Sharing Webservice mit Client-Applikation

BIKE-SHARING WEBSERVICE MIT CLIENT-APPLIKATION

Tobias, Hafermalz, Philipp Thöricht

CONTENTS

1	Einleitung und Aufgabe	2
2	Funktionen Webservice	3
3	Webclient	4
4	Zusatzaufgabe WS-Security	6
4.1	OAuth2	6
4.2	Implementierung	6
5	Fazit	8
6	Quellen	9

1 **EINLEITUNG UND AUFGABE**

Implementierung eines Webservice in PHP mit dem Slim Framework.


Zusatzaufgabe WS-Security gelöst durch Authentifizierung mittels OAuth2 und Verschlüsselung mittels HTTPS.

2 FUNKTIONEN 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

3 WEBCLIENT

B i k e S h a r i n g



[STARTSEITE](#) [SUCHE](#) [STANDORTE](#) [MODELLE](#) [LOGIN](#)

Suche

Adresse:

Postleitzahl

Stadt

Straße

Hausnummer

Umkreis in Metern:

Umkreis

Station:

wähle eine Station

Alle Angaben sind optional.

Suche Starten

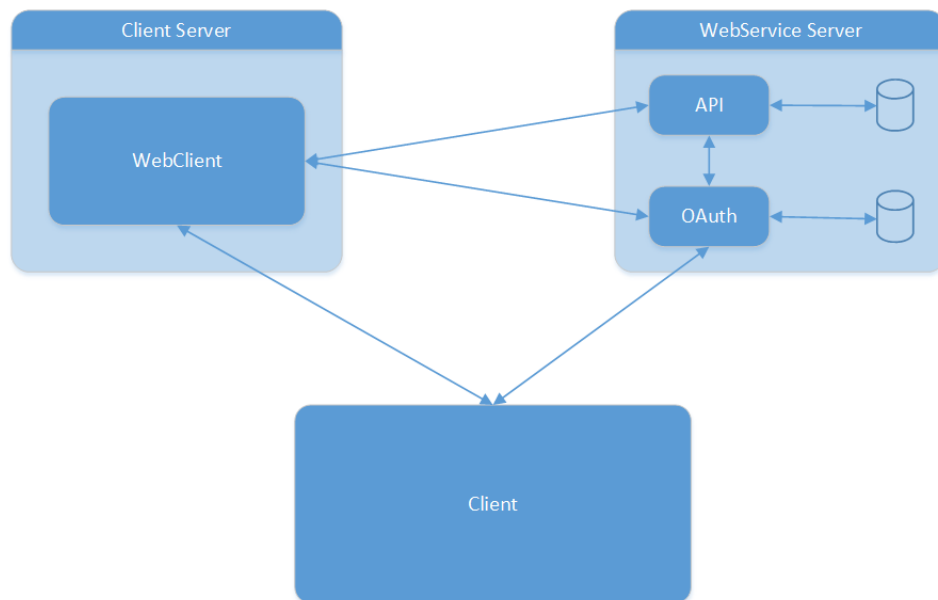
[Kontakt](#) [Impressum](#)

4 ZUSATZAUFGABE WS-SECURITY

4.1 OAUTH2

Das Architekturkonzept von OAuth2 wird durch folgendes Schema verdeutlicht.

4.2 IMPLEMENTIERUNG



5 FAZIT

Die Implementierung des Webclient ist nach Vorlage einer durchdachten API gut machbar.

Das Slim-Framework war eine gute Wahl, da die Verwendung sehr einfach und fehlerfrei verlief.
Die OAuth-Middleware hat leider nicht funktioniert.

Implementierung eines OAuth-Servers ist relativ kompliziert.

6 QUELLEN