Generierung und Design einer Client-Bibliothek für einen RESTful Web Service am Beispiel der Spreadshirt-API

Bachelorverteidigung

Andreas Linz

HTWK - Fakultät für Informatik, Mathematik & Naturwissenschaften

21. Oktober 2013

Inhaltsverzeichnis

Aufgabe

Was?

Client-Bibliothek aus abstrakter Beschreibung eines RESTful Web Service erzeugen.

Warum?

- Vereinheitlichung bestehender Implementierungen
- Nutzung der API für externe Entwickler erleichtern (z.B. Authentifizierung kapseln)

Anforderungen

- Austauschbarkeit der Zielsprache
- einfache Bedienbarkeit der Bibliothek
- gute Lesbarkeit des erzeugten Codes
- größtmögliche Typsicherheit des erzeugten Codes
- hohe Testabdeckung
- vollständige Generierung der Methoden aus der API-Beschreibung

Spreadshirt

- führendes Unternehmen für *personalisierte Bekleidung*
- *Social-Commerce* (Interaktion mit Kunden steht im Vordergrund)
- Standorte in Europa & Nordamerika, HQ in Leipzig
- $\approx$ 450 Mitarbeiter, 50 in der IT
- $4 * 10^5$ Spreadshirt-Shops mit $33 * 10^6$ Produkten

- Online-Plattform um Kleidungsstücke, Accessoires und mehr selbst zu:
  - gestalten
  - kaufen
  - zum Verkauf anbieten (eigene Designs als Motiv oder Produkt)

Spreadshirt-API

- API erlaubt Entwicklern die Nutzung eines großen Teils der Funktionen der Online-Plattform in eigenen Applikationen
- u.a. Produkterstellung, Designupload & Warenkorbverwaltung
- Erstellen eigener Shops und kundenspezifischer Anwendungen

Abbildung 1: Beispiel für kundenspezifische Anwendung: zufallsshirt.de

RESTful Web Service

REST

- *Representational State Transfer* (Gegenständlicher Zustandstransfer)
- Softwarearchitekturstil für Webanwendungen
- Anwendungen bestehen aus *Ressourcen* mit eindeutigem Bezeichner (Abbildung 2)
- Zustand einer Ressource ist eine *Repräsentation*
- Aktionen mit einer REST-API über den Austausch von Repräsentationen

Abbildung 2: Beispiel-URI, um den Artikel 42 aus dem Warenkorb 84 anzusprechen

RESTful Web Service ist eine Webanwendung die den REST Prinzipien entspricht

Dokumentbeschreibungssprachen

WADL

maschinenlesbare Beschreibung einer HTTP-basierten (typischer *RESTful*) Webanwendung

XSD

Dokumentbeschreibungssprache zur Definition von Datentypen

Gemeinsamkeiten:

- XML-Syntax, selbst wieder gültige XML-Dokumente
- Baumgrammatiken

Codegenerierung

Codegenerator

Programm welches aus einer *höhersprachigen Spezifikation*, einer Software oder eines Teilaspektes, die *Implementierung erzeugt*

Vorteile

- Produktivitässteigerung
- hohe Konsistenz des Generats
- zentrale Stelle für Änderungen (Eingabemodell)

Generatorformen

Klassifikation nach Generierungsmenge

- teilweise
  - Inline-Code Expander
  - Mixed-Code Generator
  - Partial-Class Generator
- vollständig (Tier[1]-Generator)
- mehrfach (n-Tier Generator)

[1] Stufen

Datenmodelle

Eingabe des Generators

- Applikationsmodell:
  - WADL $\rightarrow$ REST-Modell
  - XSD $\rightarrow$ Schema-Modell
- Sprachenmodell
  - kapselt Zielsprache
  - enthält Semantik
  - ! Syntax in Ausgabemodul (LanguageVisitor, siehe Abbildung 3)

Generator für die Spreadshirt-API

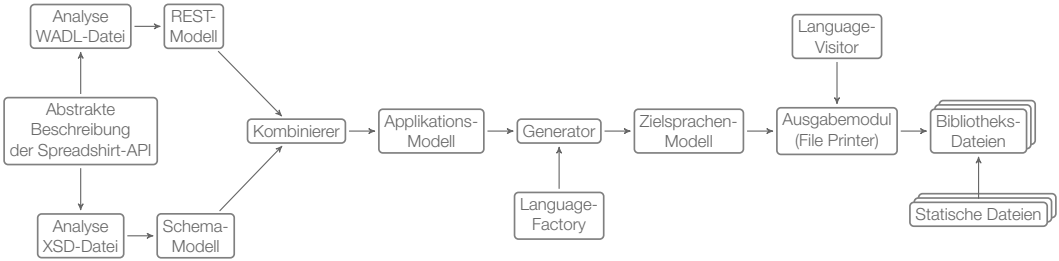

Abbildung 3: Sequenzdiagramm des Generators für die Spreadshirt-API

Datenklassen

- zielsprachenabhängige Repräsentation der Typen aus der XML-Schema Beschreibung
- Variablen bilden die Attribute und Elemente aus dem Schematyp ab
- Getter- und Setter-Methoden für alle Variablen
- transportunabhäniger Datenaustausch mit API $\rightarrow$ Methoden zur Serialisierung und Deserialisierung

```php
<?php
   require_once('Unit.php');

   class Point
   {
      private $unit; // unit
      private $y; // double
      private $x; // double

      function __construct(
            /* double */ $y,
            /* double */ $x
         )
      {
         $this->y = $y;
         $this->x = $x;
      }

      public function setUnit(
            /* unit */ $unit
         )
      {
         $this->unit = $unit;
      }
      ...
```

```
1         ...
2         public function toXML()
3         {
4            $xml =  new SimpleXMLElement(/* Point */ '<login xmlns="http://api.spreadshirt.net"/>');
5            $xml->addChild(/* string */ 'unit',/* unit */ $this->unit);
6            $xml->addChild(/* string */ 'y',/* double */ $this->y);
7            $xml->addChild(/* string */ 'x',/* double */ $this->x);
8            return $xml->asXML();
9         }
10        public static function fromXML(
11              /* SimpleXMLElement */ $xml
12           )
13        {
14           $unit = Unit::fromXML(/* SimpleXMLElement */ $xml->unit);
15           $y = $xml->y;
16           $x = $xml->x;
17           ...
18        }
19        ...
20        public function getX()
21        {
22           return $x = $this->x;
23        }
24     }
25  ?>
```

Ressourcenklassen

- zielsprachenabhängige Abbildung der Ressourcenbeschreibungen aus WADL-Datei
- Methoden der Klassen entsprechen den Methoden der abgebildeten Ressource

```php
<?php
   require_once('Static/methods.php');
   require_once('Static/apiUser.php');
   /* Create or list products for user. */
   class UsersUserIdProducts
   {
      private $baseUrl = 'http://192.168.13.10:8080/api/v1/'; // string
      ...
      /*  */
      public function POST(
            /* array */ $parameters,
            /* ApiUser */ $apiUser,
            /* ProductDTO */ $productDTO
         )
      { ... }
      ...
      function __construct(
            /* string */ $userId
         )
      {
         $this->userId = $userId;
         $this->resourceUrl = $this->baseUrl . 'users' . '/' . $userId . 'products';
      }
   }
?>
```

statische Klassen

- wurden manuell erstellt
- enthalten gemeinsam genutzten Code ohne variable Bestandteile
- Bibliothek enthält zwei dieser Klassen:
  - Kommunikation über HTTP-Methoden mit der API
  - Kapselung der Authentifizierung

Zusammenfassung

- Entwicklung der Datenmodelle
- Überführung der Beschreibung in Eingabedatenmodelle des Generators
- Erstellung und Implementierung des Sprachenmodells für PHP
- Implementierung des Generators
- Generierung der Bibliothek

Ausblick

- Parameterobjekte
- Fluent-Interface
- Java-Bibliothek (Sprachenmodell)
- Erzeugung von Dokumentation und Testdaten

Diskussion

- XSD, WADL
- RESTful Web Service
- Datenmodelle für Web Service Beschreibung und Programmiersprache
- Stufen-Generator
- PHP Client-Bibliothek