

Aufgabe 2A: RESTful Webservices

Erstellung eines RESTful-Webservice mit PHP

Die Webanwendung *PhotonPainter* arbeitet serviceorientiert und greift daher serverseitig über eine REST-Webservice-Schnittstelle auf die zugrundeliegenden Daten zu und generiert daraus ein XHTML-Frontend. Der Dienst *PhotonEmitter* soll die Aufgabe der Datenbereitstellung zuverlässig leisten.

Ihre Aufgabe ist es, einen RESTful-Webservice zur Verwaltung des Bilddatenstamms zu entwickeln. Neben XML soll JSON als Format für den Datenaustausch dienen, wobei die jeweiligen Schemas fest vorgegeben sind. Bilder werden direkt im Binärformat übertragen. Die eigentlichen Daten sollen in einer internen Datenbank gespeichert werden, die Bilder in einem dafür vorgesehenen Ordner im Dateisystem. Zusätzlich soll der Webservice Vorschauansichten für die hochgeladenen Bilder erzeugen.

Das Ergebnis muss folgende Anforderungen erfüllen:

- Der RESTful-Webservice muss alle durch die WADL-Datei spezifizierten Operationen mittels der darin definierten URLs abdecken.
- Fehler wie etwa ungültige IDs, fehlende Eingabedaten, etc. müssen erkannt und mit einer Fehlermeldung beantwortet werden. Hierfür sind die HTTP-Fehlercodes zu nutzen (siehe WADL). Die zusätzliche Übertragung von menschenlesbaren Fehlernachrichten ist optional.
- *PhotonEmitter* muss sich für die Datenübertragung an das zugrundeliegende Schema `photonpainter.xsd` halten. Zu unterstützende Bildformate sind JPEG und PNG, Bilder bis 10 MB Dateigröße und beliebiger Abmessungen sind zu unterstützen.
- Von hochgeladenen Bildern müssen mehrere Vorschauversionen erzeugt werden – eine Miniatur (`tinyimg`, max. 50x50 Pixel), eine für die Listenansicht (`thumbnail`, max. 120x120 Pixel) und eine für eine größere Detailansicht, bevor das Bild in voller Auflösung angezeigt wird (`smallimg`, max. 400x400 Pixel).
- Nutzen Sie eine interne, Dateisystem-basierte Datenbank wie SQLite3.
- *Alle verwendeten Bestandteile müssen entweder Public Domain sein oder die erforderlichen Nutzungsrechte müssen schriftlich vorliegen.*

Abzugeben sind:

- Gut dokumentierte Quellen (z.B. als gepacktes Verzeichnis)

Tipp:

- Nutzen Sie die Möglichkeiten von existierenden PHP-Frameworks wie CakePHP. Diese bieten viele Vorteile, wie etwa ORM-Mapper für einen einfachen Zugriff auf die SQL-Daten, Möglichkeiten zum Routing von URLs, Rechteverwaltung oder auch die Verwendung von Templates für unterschiedliche Ausgabeformate.
- Erstellen Sie zunächst Stubs für die benötigten URLs. Widmen Sie sich dann nacheinander der Implementierung der einzelnen Operationen: Beginnen Sie mit der Ausgabe der Bild-Datensätze, setzen Sie dann das Einfügen von Daten und zuletzt das Hochladen von Bildern um.