

RESTful Web Service mit Django

Web Services werden immer beliebter. Mit Hilfe von REST ([Representational State Transfer](#)) und [HTTP](#) kann man einfache und trotzdem leistungsfähige Web Services bauen.

In diesem Workshop werden wir eine bereits existierende Django-Applikation um einen REST Web Service erweitern. Dieser wird dann über ein JavaScript-Frontend angesprochen.

Vorkenntnisse

Grundkenntnisse in den folgenden Gebieten:

- Objektorientierte Programmierung
- HTTP
- Document Object Model (DOM)

Inhalt im Detail

Nach einer kurzen Einführung in die Grundarchitektur von Django und der Vorstellung der Struktur des Workshops werden wir gemeinsam mit den Workshopteilnehmern das [Django-Webframework](#) auf jedem Rechner einrichten. Danach werden wir eine von uns erstellte Django-Applikation - es handelt sich um ein einfaches CRM-System - auf den Rechnern der Teilnehmer installieren.

Nun beginnen wir mit der Erstellung des Webservice. Wir werden dazu eine weitere Django-Applikation namens [Piston](#) installieren. Diese werden wir verwenden, um eine eigene Schnittstelle für die bestehende Applikation zu programmieren. Zunächst ermöglicht uns diese Schnittstelle, über eine einfache Browseranfrage die in der Application hinterlegten Daten zunächst als XML oder JSON zu erhalten. Diese Schnittstelle werden wir danach erweitern, um auch komplexere Anfragen an den Webservice stellen zu können.

Im Folgenden werden die Workshopteilnehmer mittels einer von uns bereits erstellten Rich Internet Application (RIA) auf Basis eines JavaScript-GUI-Frameworks eine Einsatzmöglichkeit des oben gebauten Webservice ausprobieren.

Die weiteren Module des Workshops sind optional. Sie können je nach verfügbarer Zeit in beliebiger Reihenfolge erarbeitet werden:

- Erweiterung der Schnittstelle um eine POST-Anfrage zur Änderung der Daten über die JavaScript-Applikation
- Löschen von Einträgen über die JavaScript-Applikation (DELETE-Operation)
- Erstellen eines alternativen Ausgabeformats, z.B. zur Generierung von CSV-Daten (Programmieren eines eigenen Handlers)

Links

- [Django](#)
- [Piston](#)