
B. Strobel, K. Krammer, L. Hnatek, S. Fischer
Webengineering - Blatt 02
WiSe 2013/14, Inst. f. Medieninformatik, Universität Ulm

02. Aufgabe: HTTP

1. Eine idempotente HTTP-Methode kann mehrere male aufgerufen werden, ohne dass sich das Ergebnis ändert.

Mit POST wird etwas erstellt. Durch die POST-Methode wird gefordert, dass der Server die mitgeschickte Repräsentation akzeptiert, welche durch die Zielressource verarbeitet werden soll.

Mit PUT wird etwas erstellt oder geupdated. Durch die PUT-Methode wird gefordert, dass der Zustand der Zielressource erzeugt oder ersetzt wird durch den definierten Zustand der Repräsentation, welche im Request mitgeschickt wird.

Beispiele:

“POST/buch” wird ein neues Buch kreieren und der Server antwortet mit einer neue URL, die das Buch identifiziert. Also beispielsweise “/buch/2”.

“PUT/buecher/2” wird entweder das neue Buch mit der ID 2 erstellen, oder das bereits existierende Buch mit der ID 2 ersetzen.

<http://tools.ietf.org/html/draft-ietf-httpbis-p2-semantics-21#section-5.3.4>

2. Methoden die “safe” sind, werden nie die Repräsentation einer Ressource manipulieren (Z.B. HEAD, OPTIONS und GET).

<http://restcookbook.com/HTTP%20Methods/idempotency/>

3. Der Zweck einer persistenten HTTP-Verbindung ist, die gleiche TCP-Verbindung zu nutzen um mehrere Requests/Responses zu senden/erhalten, statt immer wieder die Verbindug zum Server neu aufzubauen. Der Client kann mehrere Requests senden, ohne zuerst auf jeden Response warten zu müssen (effizienter in kürzerer Zeit).

Eine persistente Verbindung muss nicht aufgebaut werden, wenn sowieso nur ein Request/Response erfolgen soll.

<http://tools.ietf.org/html/draft-ietf-httpbis-p2-semantics-21#section-5.3.4>

4. Die angegebene Seite

`http://mobile3.informatik.uni-ulm.de/we01/cat.jpg`

ist eigentlich eine HTML Datei. Der Browser erkennt das anhand der übertragenen MIME Types und zeigt die Datei korrekt an.

Da über CSS das Hintergrundbild für den Body angegeben wurde, kann auch dieses angezeigt werden. Unabhängig davon, dass der Dateiname auf .html endet, wird auch hier über den MIME Type die Datei erkannt. D.h. der Response des Servers enthält die Information:

Content-Type: image/jpeg

Der Browser integriert dieses File dann korrekt in den DOM-Baum und rendert das Bild mit dem für den Typen definierten Renderer, weshalb zuletzt das Katzenbild erscheint.

`http://www.w3schools.com/cssref/pr_background-image.asp`