

Richtlinien für REST-Service-Implementierungen

- **Verwendung**
 - 1) Strikte Verwendung von GET, POST, PUT, DELETE nach Rest Standard:
 - 2) Selbstsprechende "objektorientierte" Pfade, z.B:
 - 3) Dabei zusätzliche, nicht unbedingt notwendige Pfadteile soweit wie möglich vermeiden. Insbesondere am Anfang:
 - 4) Beispiel Code für einen Restservice zur Userverwaltung:

Verwendung

1) Strikte Verwendung von GET, POST, PUT, DELETE nach Rest Standard:

GET: Holen von Daten, keine Veränderung in der DB

POST: Erzeugen eines Objects in der DB

PUT: Verändern eines Objects in der DB. Unter bestimmten Bedingungen kann mit PUT auch ein Object erzeugt werden, dies ist aber im Gegensatz zu POST idempotent, ein zweites PUT erzeugt keinen neuen Datensatz.

DELETE: Löschen eines (oder mehrerer) Objects in der DB.

2) Selbstsprechende "objektorientierte" Pfade, z.B:

einen Baum erzeugen:

```
POST /tree
```

einen Baum verändern:

```
PUT /tree/{treeid}
```

einen Ast am Baum entfernen:

```
DELETE /tree/{treeid}/branch/{branchid}
```

3) Dabei zusätzliche, nicht unbedingt notwendige Pfadteile soweit wie möglich vermeiden. Insbesondere am Anfang:

Suboptimal:

```
/neudazu/botanik/heinzmueller/pflanzen/tree/
```

4) Beispiel Code für einen Restservice zur Userverwaltung:

```
@Path("/user")
public class UserService {

    @GET
    public List<UserRVO> getUserList(@PathParam("username") String username) {
        ...
    }

    @POST
    public UserRVO createUser(UserRVO userRVO) {
        ...
    }

    @GET
    @Path("/{username}")
    public UserRVO getUser(@PathParam("username") String username) {
        ...
    }

    @PUT
    @Path("/{username}")
    public UserRVO updateUser(@PathParam("username") String username, UserRVO userRVO) {
        ...
    }

    @DELETE
    @Path("/{username}")
    public UserRVO deleteUser(@PathParam("username") String username) {
        ...
    }

    @PUT
    @Path("/{username}/password")
    public Response setPassword(@PathParam("username") String username, PasswordChangeRVO passwordChange) {
        ...
    }
}
```