# Übung 5 – REST
## Methoden des Software Engineering

Christian Kroiß

"RESTful applications maximize the use of the pre-existing, well-defined interface and other built-in capabilities provided by the chosen network protocol, and minimize the addition of new application-specific features on top of it." (Wikipedia)

- A RESTful web service (also called a RESTful [web API](#)) is a simple web service implemented using HTTP and the principles of REST. It is a collection of resources, with three defined aspects:
  - the base URI for the web service, such as http://example.com/resources/
  - the [MIME type](#) of the data supported by the web service. This is often [JSON](#), [XML](#) or [YAML](#) but can be any other valid MIME type.
  - the set of operations supported by the web service using [HTTP methods](#) (e.g., POST, GET, PUT or DELETE).

| Resource | GET | PUT | POST | DELETE |
|---|---|---|---|---|
| **Collection URI, such as** `http://example.com /resources/` | **List** the URIs and perhaps other details of the collection's members. | **Replace** the entire collection with another collection. | **Create** a new entry in the collection. The new entry's URL is assigned automatically and is usually returned by the operation. | **Delete** the entire collection. |
| **Element URI, such as** `http://example.com /resources/142` | **Retrieve** a representation of the addressed member of the collection, expressed in an appropriate Internet media type. | **Update** the addressed member of the collection, or if it doesn't exist, **create** it. | Treat the addressed member as a collection in its own right and **create** a new entry in it. | **Delete** the addressed member of the collection. |

"The JAX-RS API uses Java programming language annotations to simplify the development of RESTful web services."

```
@Path("/helloworld")
public class HelloWorldResource {

    @GET
    @Produces("text/plain")
    public String getClichedMessage() {
        return "Hello World";
    }
}
```