|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| REST Befehl | REST Methode in Java | Beschreibung | Daten von Server | Anfrage Client |
| GET | @RequestMapping(method = RequestMethod.***GET***)  **public** ResponseEntity<List<Buch>> **getListJson**() | Durch die Annotation RequestMapping wird mit Spring aus dieser Methode eine GET REST-Methode. Diese Methode sendet eine Liste von Objekten vom Typ Buch im JSON Format. | {  "id" : 1,  "titel" : "Android-Apps entwickeln",  "autor" : "Uwe Post",  "verlag" : "Rheinwerk Computing",  "isbn" : "ISBN 978-3-8362-385-3",  "auflage" : 5,  "erscheinungsjahr" : 2015,  "kategorie" : "Fachbuch IT",  "links" : [ {  "rel" : "self",  "href" : "http://buch/1"  } ]  } | http://localhost:8080/ui/buch/bearbeiten |
| GET | @RequestMapping(value = "/{id}", method = RequestMethod.***GET***)  **public** ResponseEntity<Buch> **getBuch**(@PathVariable("id") Integer id) | Durch die Annotation RequestMapping wird mit Spring aus dieser Methode eine GET REST Methode, die erst angesprochen wird, wenn die Ziel-URL eine ID erhält. Mit dieser ID kann die Methode ein einzelnes Buch aus der Datenbank suchen. Dieses wird im JSON Format an den Client gesendet. | {  "id" : 1,  "titel" : "Android-Apps entwickeln",  "autor" : "Uwe Post",  "verlag" : "Rheinwerk Computing",  "isbn" : "ISBN 978-3-8362-385-3",  "auflage" : 5,  "erscheinungsjahr" : 2015,  "kategorie" : "Fachbuch IT",  "links" : [ {  "rel" : "self",  "href" : "http://buch/1"  } ]  } | http://localhost:8080/ui/buch/aendern/1 |
| POST | @RequestMapping(method = RequestMethod.***POST***)  **public** ResponseEntity<Integer> **speichern**(@RequestBody Buch b) | Durch die Annotation RequestMapping wird die Methode zu einer REST POST Methode. Diese Methode erhält vom Client das Objekt Buch im JSON Format. Damit dies jedoch die REST Schnittstelle erhält wird die Annotation RequestBody benötigt. Würde diese Annotation fehlen, wäre der Parameter Buch initial und somit „leer“. | {  "id" : 1,  "titel" : "Android-Apps entwickeln",  "autor" : "Uwe Post",  "verlag" : "Rheinwerk Computing",  "isbn" : "ISBN 978-3-8362-385-3",  "auflage" : 5,  "erscheinungsjahr" : 2015,  "kategorie" : "Fachbuch IT",  "links" : [ {  "rel" : "self",  "href" : "http://buch/1"  } ]  } | http://localhost:8080/ui/buch/anlegen |
| DELETE | @RequestMapping(value = "/{id}", method = RequestMethod.***DELETE***)  **public** ResponseEntity<Void> **loeschen**(@PathVariable("id") Integer id) | Durch die Annotation RequestMapping und dem Wert der Konstanten RequestMethod.DELETE wird die Methode zur DELETE REST Methode. Diese benötigt eine ID von dem Client um das Richtige Objekt aus der Datenbank zu löschen.  Die Methode gibt keine Rückgabe an den Client, da die Daten, die dem Client zur Verfügung stehen immer Synchron sind mit den Daten aus der Datenbank. |  | http://localhost/ui/buch/ bearbeiten /id |
| PATCH | @RequestMapping(value = "/{id}", method = RequestMethod.***PATCH***)  **public** ResponseEntity<Void> **bearbeiten**(@PathVariable("id") Integer id, @RequestBody Buch b) | Durch die Annotation RequestMapping und dem Wert der Konstaten RequestMethod PATCH wird die Methode zur PATCH REST Methode.  Die Methode hat die Parameter id und ein Objekt vom Typ Buch.  Die ID wird benötigt um das aktuelle Buch aus der Datenbank zu ermitteln und das Objekt vom Typ Buch das geänderte Buch zu empfangen. | {  "id" : 1,  "titel" : "Android-Apps entwickeln",  "autor" : "Uwe Post",  "verlag" : "Rheinwerk Computing",  "isbn" : "ISBN 978-3-8362-385-3",  "auflage" : 5,  "erscheinungsjahr" : 2015,  "kategorie" : "Fachbuch IT",  "links" : [ {  "rel" : "self",  "href" : "http://buch/1"  } ]  } | http://localhost:8080/ui/buch/aendern/1 |