

Eine erste REST Modellierung - Definition der Verben

Definition der Verben - Ziele

- Eine erste Definition der Semantik der REST Verben für ein eigenes Problemszenario durchführen können.
- Die Eigenschaften von HTTP Verben erklären können.

Erstellung eines neuen „order“

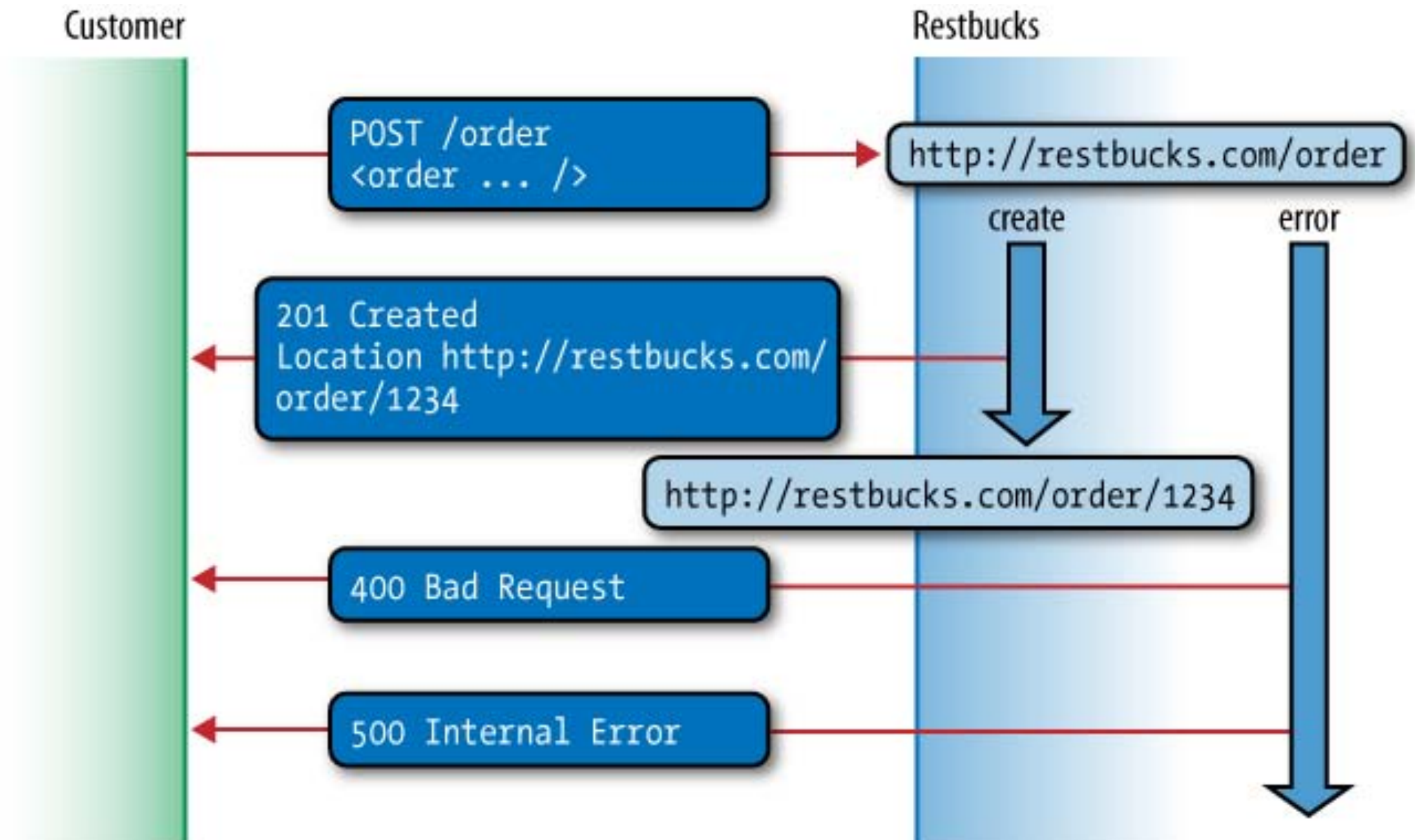


Figure 4-4. *Creating an order via POST*

Quelle: Webber et al: REST in Practice, Abruf 3.4.2017)

Der zugehörige HTTP POST Request

Example 4-1. *Creating a coffee order via POST*

POST /order HTTP/1.1

Host: restbucks.com

Content-Type: application/xml

Content-Length: 239

```
<order xmlns="http://schemas.restbucks.com/order">
  <location>takeAway</location>
  <items>
    <item>
      <name>latte</name>
      <quantity>1</quantity>
      <milk>whole</milk>
      <size>small</size>
    </item>
  </items>
</order>
```

Quelle: Webber et al: REST in Practice, Abruf
3.4.2017)

.. und die Response des Servers

HTTP/1.1 201 Created

Content-Length: 267

Content-Type: application/xml

Date: Wed, 19 Nov 2008 21:45:03 GMT

Location: <http://restbucks.com/order/1234>

```
<order xmlns="http://schemas.restbucks.com/order">
  <location>takeAway</location>
  <items>
    <item>
      <name>latte</name>
      <quantity>1</quantity>
      <milk>whole</milk>
      <size>small</size>
    </item>
  </items>
  <status>pending</status>
</order>
```

Quelle: Webber et al: REST in Practice, Abruf
3.4.2017)

... und eine Java Implementierung als Servlet

Example 4-5. *A Java servlet implementation for creating a coffee order*

```
protected void doPost(HttpServletRequest request, HttpServletResponse response) {
    try {
        Order order = extractOrderFromRequest(request);
        if(order == null) {
            response.setStatus(HttpServletResponse.SC_BAD_REQUEST);
        } else {
            String internalOrderId = saveOrder(order);
            response.setHeader("Location", computeLocationHeader(request,
                                                                    internalOrderId));
            response.setStatus(HttpServletResponse.SC_CREATED);
        } catch(Exception ex) {
            response.setStatus(HttpServletResponse.SC_INTERNAL_SERVER_ERROR);
        }
    }
}
```

Quelle: Webber et al: REST in Practice, Abruf 3.4.2017)

Anfordern einer Repräsentation einer Ressource

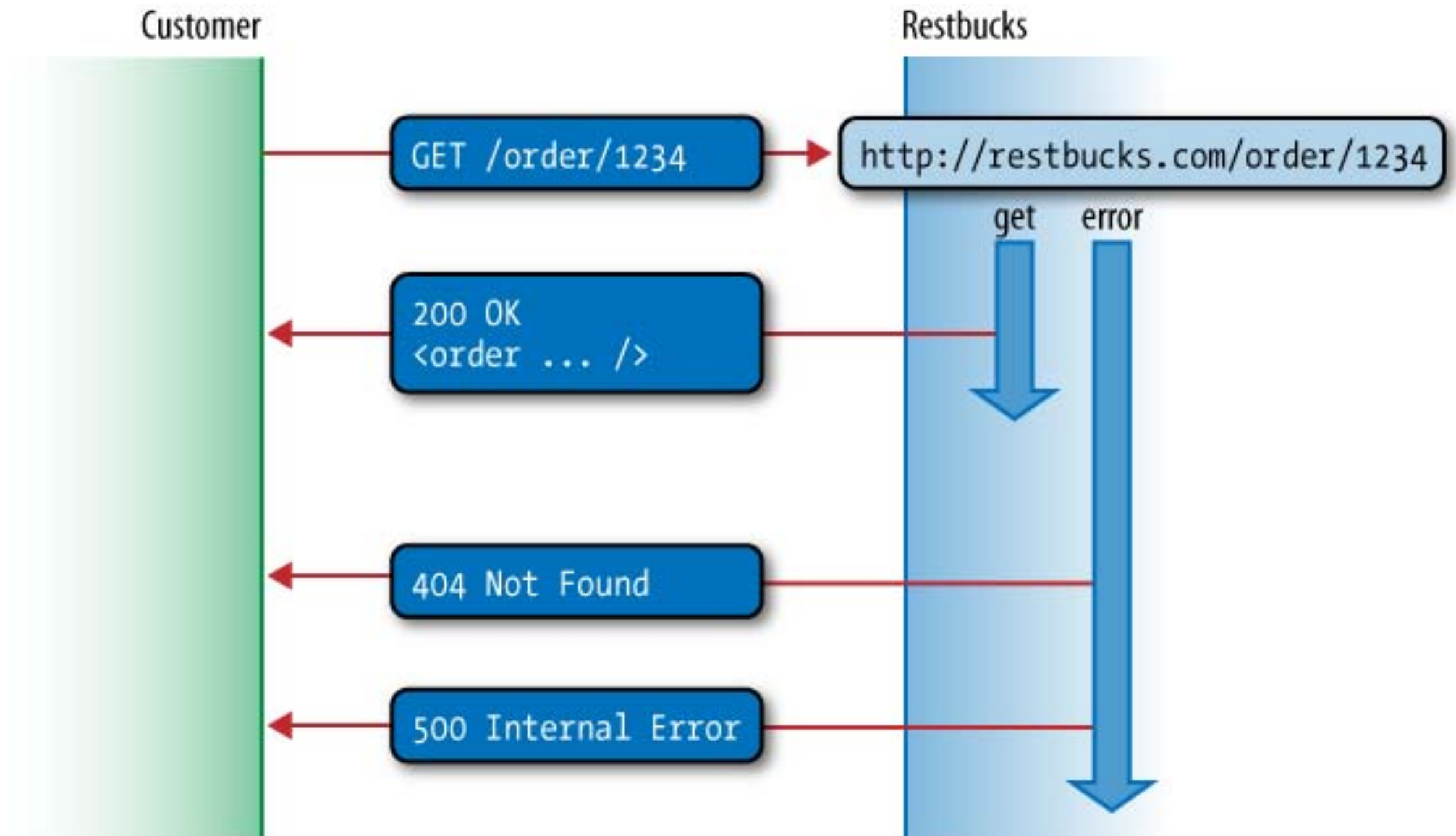


Figure 4-5. *Reading a coffee order with GET*

Quelle: Webber et al: REST in Practice, Abruf 3.4.2017)

Aktualisieren einer Ressource

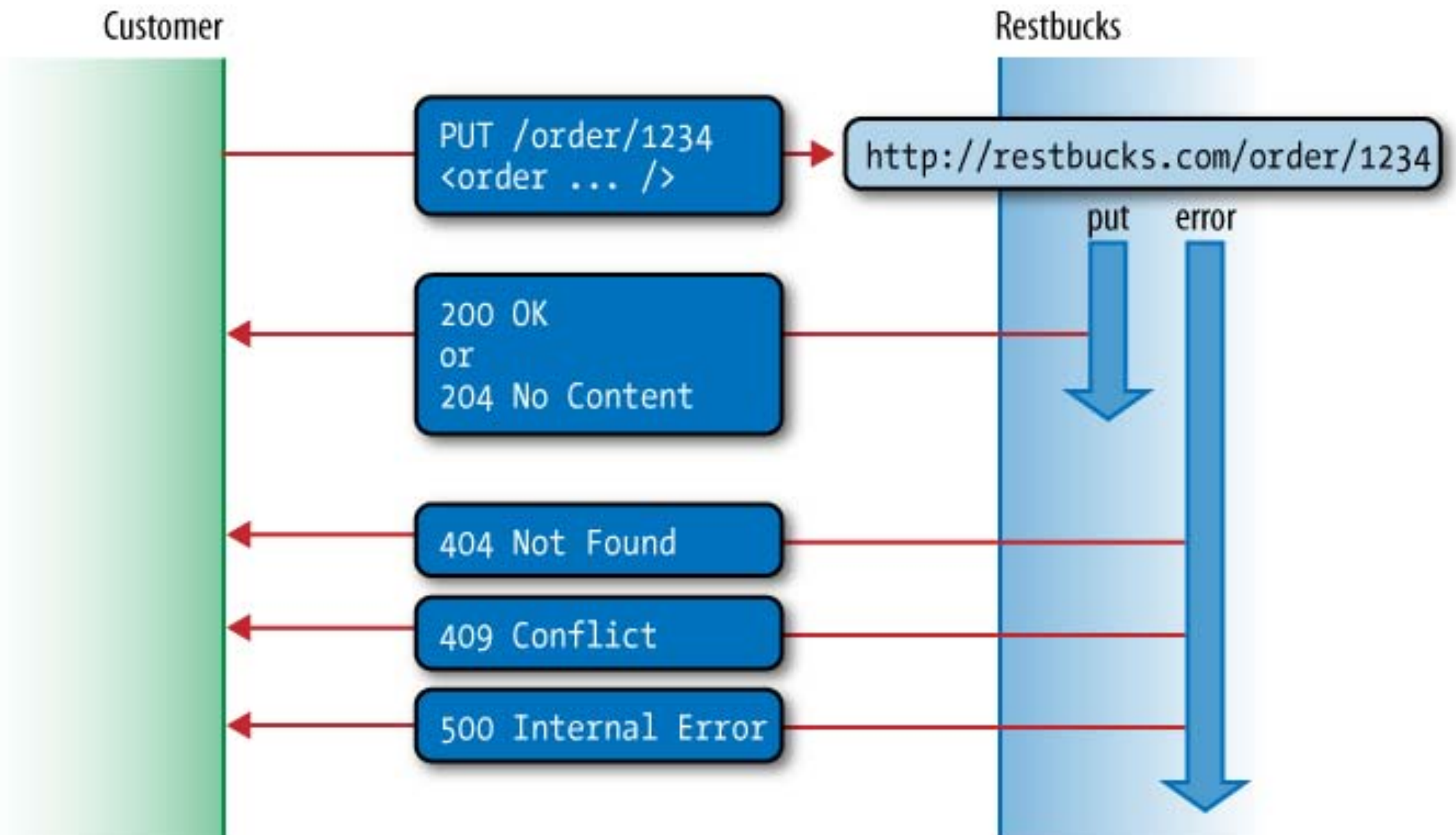


Figure 4-6. *PUT request and responses*

Quelle: Webber et al: REST in Practice, Abruf 3.4.2017)

Löschen einer Ressource

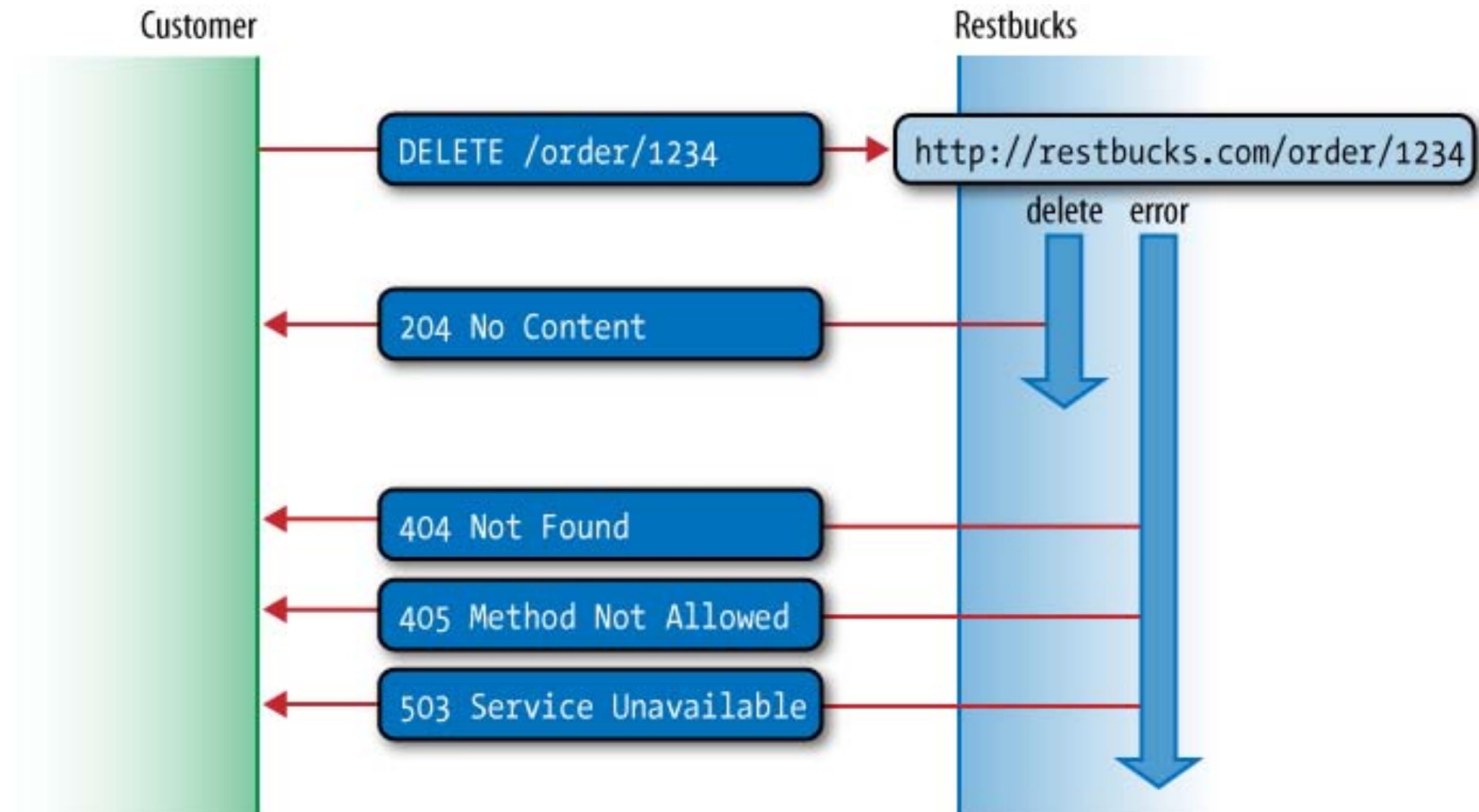


Figure 4-7. *DELETE request and responses*

Quelle: Webber et al: REST in Practice, Abruf 3.4.2017)

Übersicht über HTTP Verben

	sicher	idem- potent	Cache fähig	sichtbare Semantik
GET	x	x	x	x
PUT		x		x
POST				
DELETE		x		x

Definition der Verben- Zusammenfassung

- Eine erste Definition der Semantik der REST Verben für die Ressource „order“ in RESTbucks
- Eigenschaften von HTTP Verben: *idempotent* und *safe*