Spring CRM REST - Delete Customer

Real-Time Project

HTTP Method		CRUD Action
POST	/api/customers	Create a new customer
GET	/api/customers	Read a list of customers
GET	/api/customers/{customerId}	Read a single customer
PUT	/api/customers	Update an existing customer
DELETE	/api/customers/{customerId}	Delete an existing customer

Real-Time Project

Checkpoint

HTTP Method		CRUD Action
POST	/api/customers	Create a new customer
GET	/api/customers	Read a list of customers
GET	/api/customers/{customerId}	Read a single customer
PUT	/api/customers	Update an existing customer
DELETE	/api/customers/{customerId}	Delete an existing customer

Real-Time Project

Checkpoint

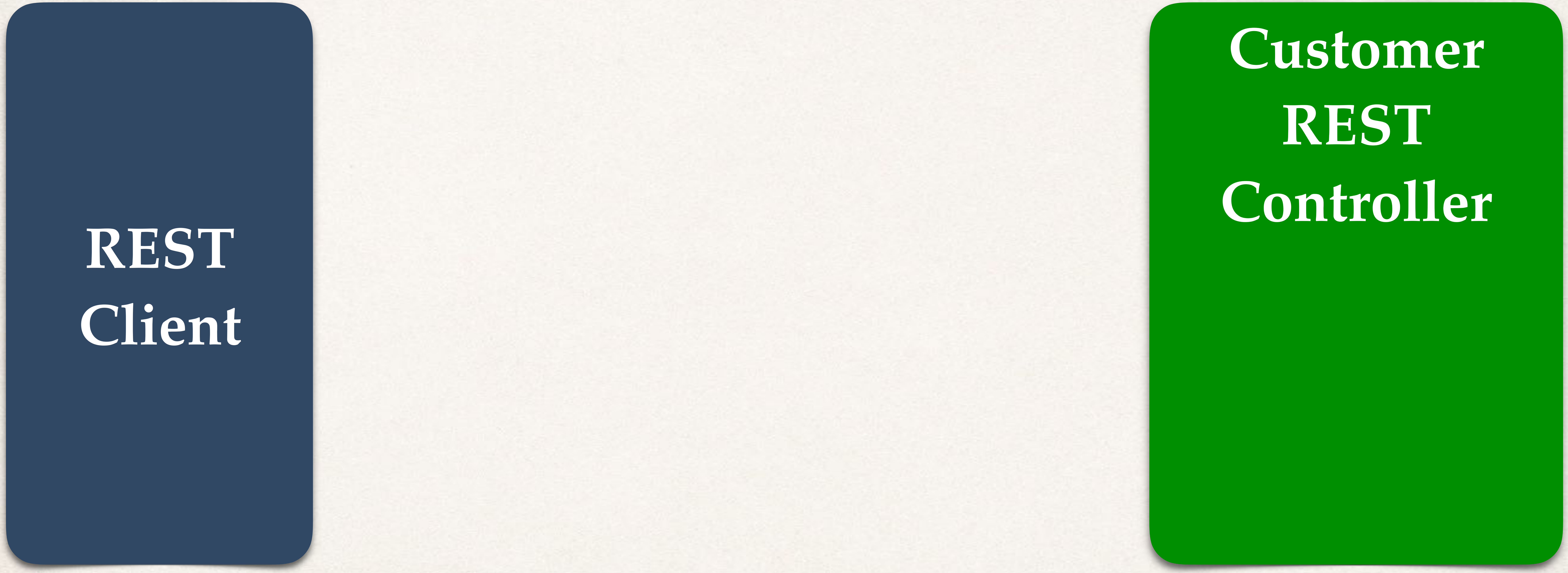
HTTP Method		CRUD Action
✓ POST	/api/customers	Create a new customer
✓ GET	/api/customers	Read a list of customers
✓ GET	/api/customers/{customerId}	Read a single customer
✓ PUT	/api/customers	Update an existing customer
DELETE	/api/customers/{customerId}	Delete an existing customer

Real-Time Project

Checkpoint

HTTP Method		CRUD Action
✓ POST	/api/customers	Create a new customer
✓ GET	/api/customers	Read a list of customers
✓ GET	/api/customers/{customerId}	Read a single customer
✓ PUT	/api/customers	Update an existing customer
✎ DELETE	/api/customers/{customerId}	Delete an existing customer

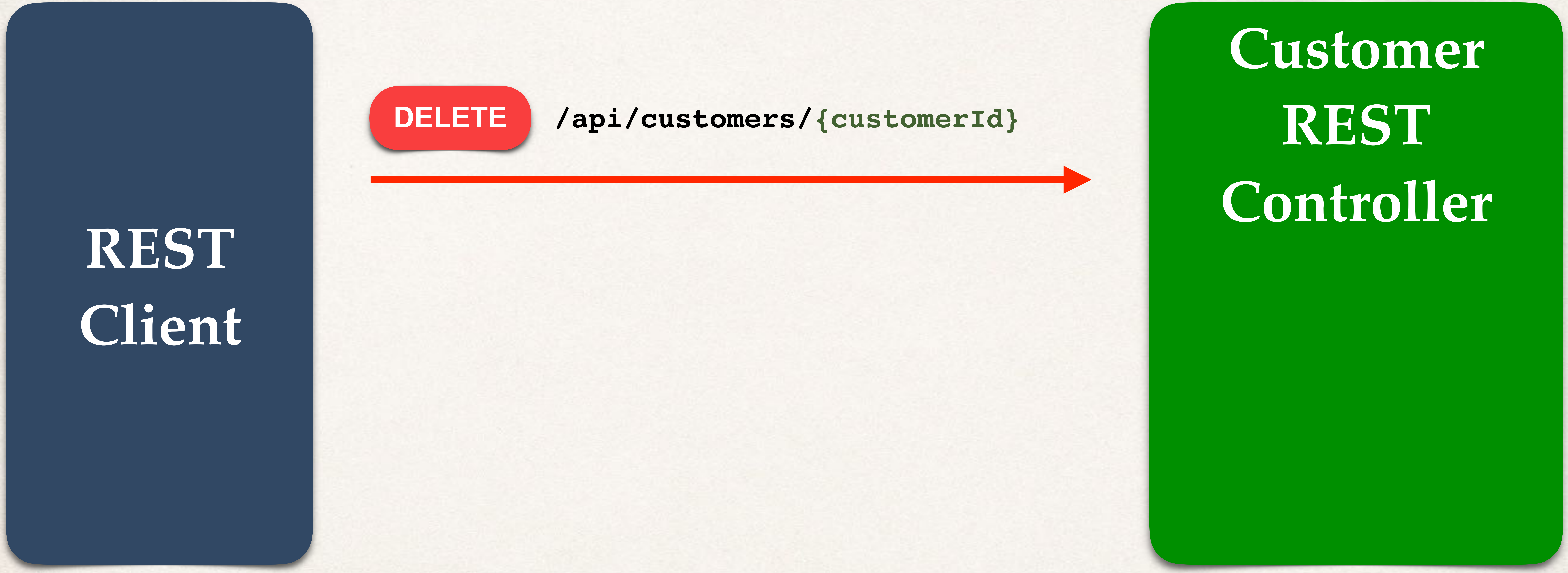
Application Interaction



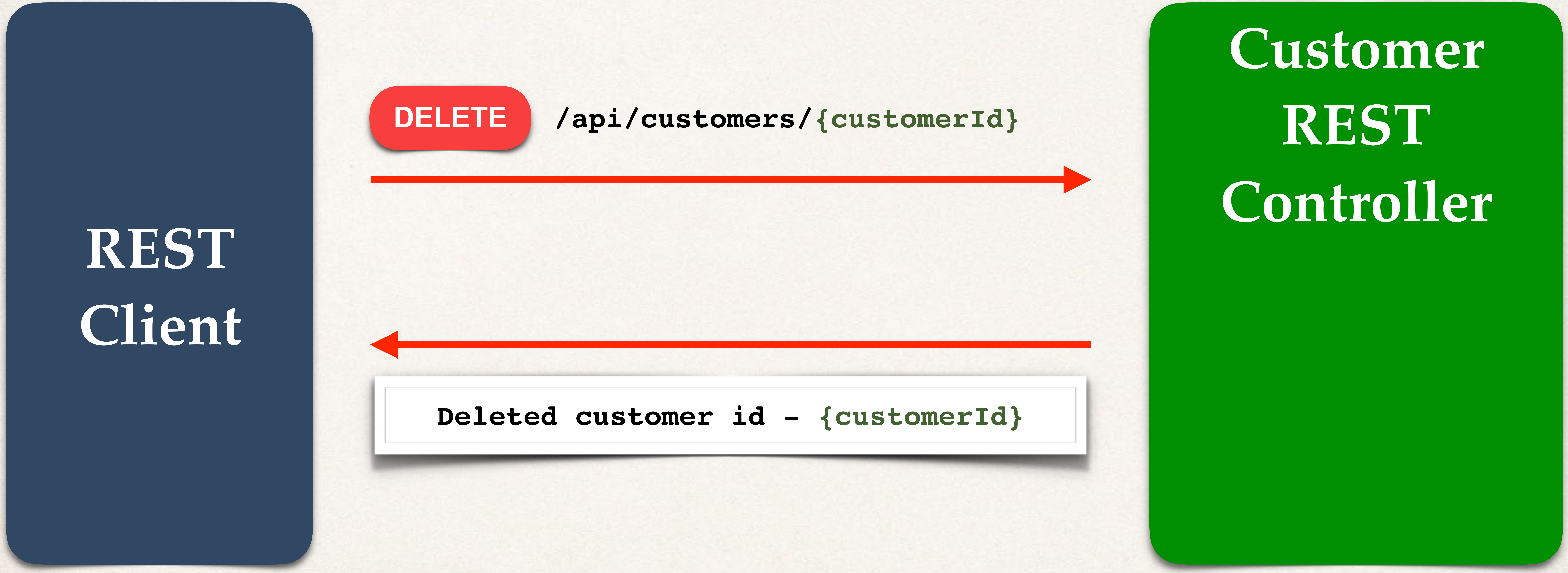
REST Client

Customer REST Controller

Application Interaction



Application Interaction



REST Client

DELETE `/api/customers/{customerId}`

`Deleted customer id - {customerId}`

Customer REST Controller

Delete Customer

File: StudentRestController.java

```
@RestController
@RequestMapping("/api")
public class CustomerRestController {
  …

  // add mapping for DELETE /customers  - delete customer

  @DeleteMapping("/customers/{customerId}")
  public String deleteCustomer(@PathVariable int customerId) {



  }

}
```

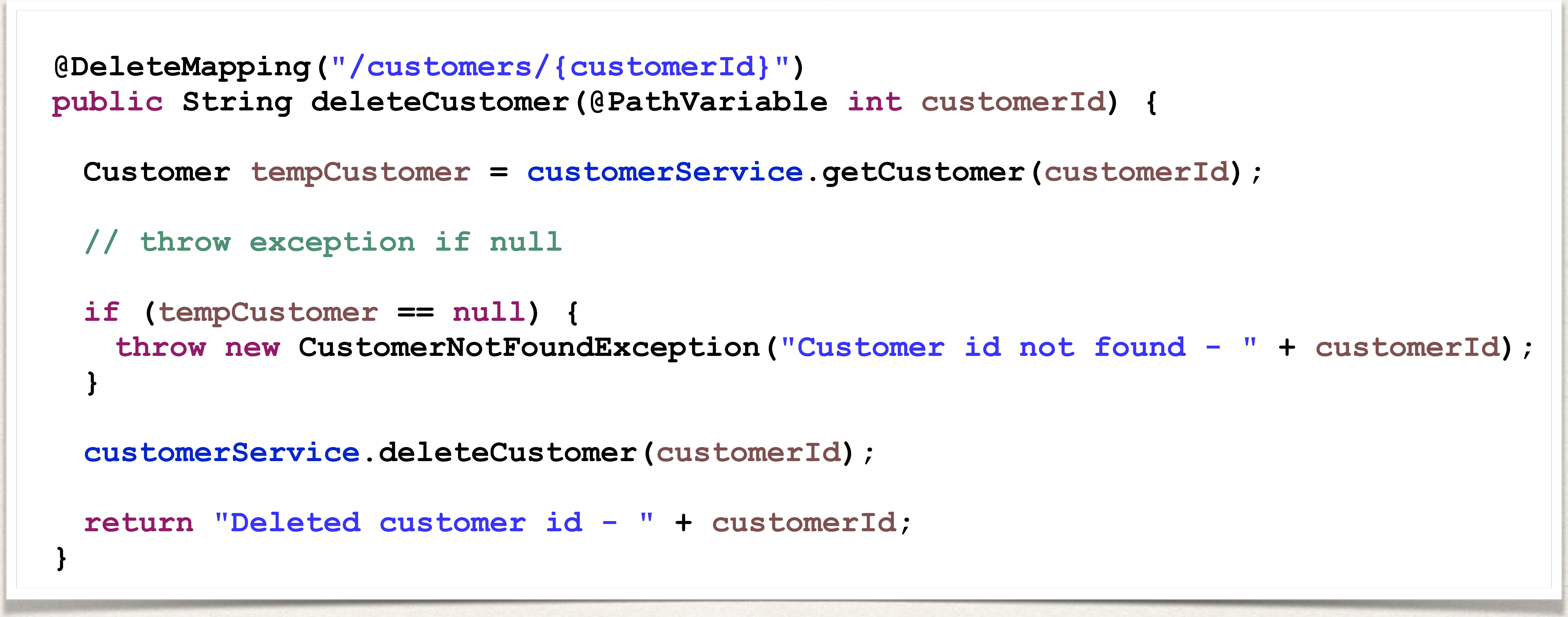
Delete Customer

File: StudentRestController.java

```
@RestController
@RequestMapping("/api")
public class CustomerRestController {
  …

  // add mapping for DELETE /customers  - delete customer

  @DeleteMapping("/customers/{customerId}")
  public String deleteCustomer(@PathVariable int customerId) {

    customerService.deleteCustomer(customerId);

    return "Deleted customer id - " + customerId;
  }

}
```

Delete Customer - Refactor

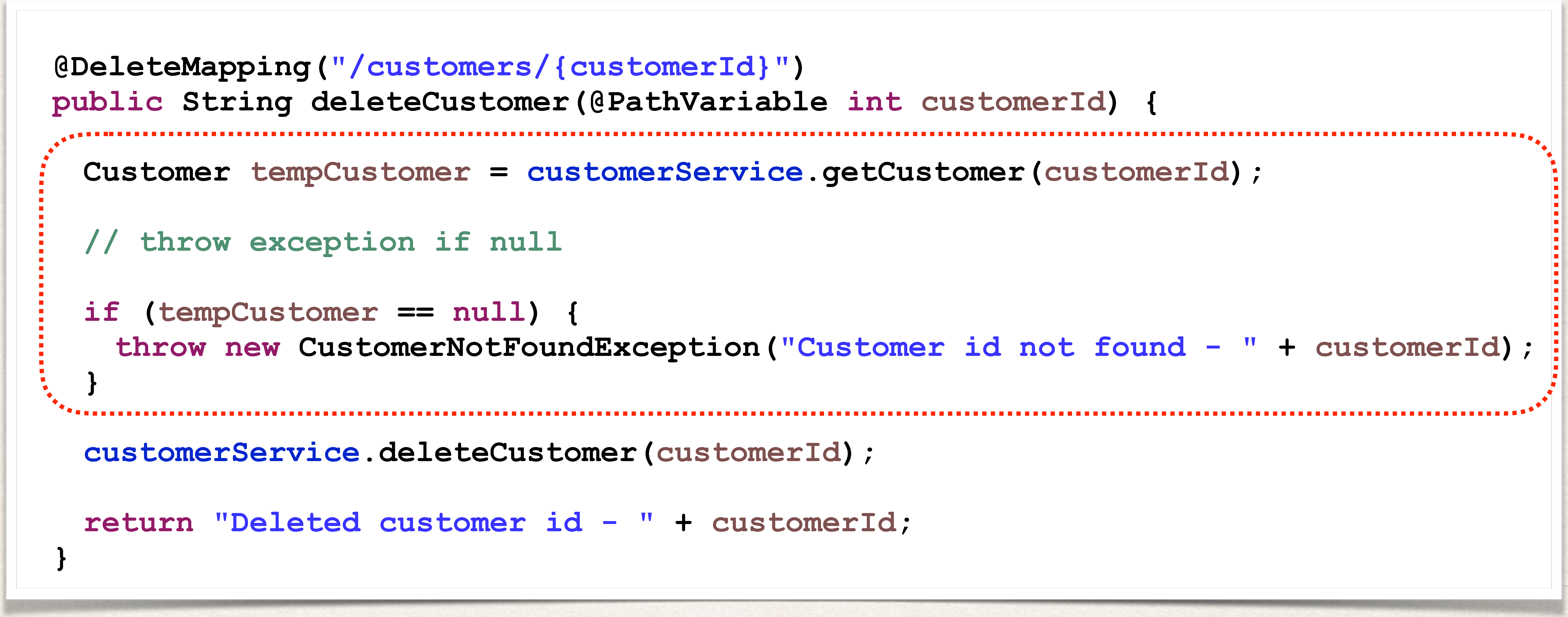
File: StudentRestController.java



```
@DeleteMapping("/customers/{customerId}")
public String deleteCustomer(@PathVariable int customerId) {

  Customer tempCustomer = customerService.getCustomer(customerId);

  // throw exception if null

  if (tempCustomer == null) {
    throw new CustomerNotFoundException("Customer id not found - " + customerId);
  }

  customerService.deleteCustomer(customerId);

  return "Deleted customer id - " + customerId;
}
```

Delete Customer - Refactor

File: StudentRestController.java



```
@DeleteMapping("/customers/{customerId}")
public String deleteCustomer(@PathVariable int customerId) {

  Customer tempCustomer = customerService.getCustomer(customerId);

  // throw exception if null

  if (tempCustomer == null) {
    throw new CustomerNotFoundException("Customer id not found - " + customerId);
  }

  customerService.deleteCustomer(customerId);

  return "Deleted customer id - " + customerId;
}
```

Delete Customer - Refactor

Check to see if customer exists

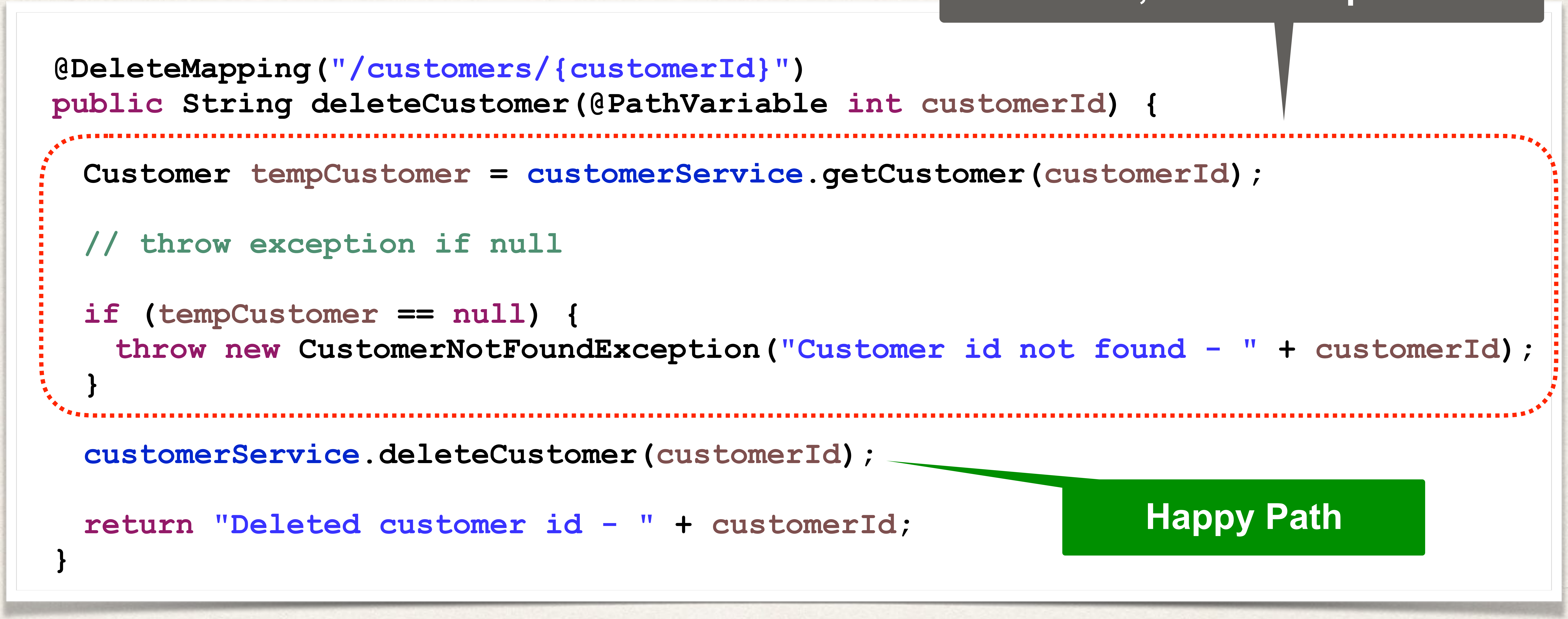
If not, throw exception

File: StudentRestController.java

```
@DeleteMapping("/customers/{customerId}")
public String deleteCustomer(@PathVariable int customerId) {

  Customer tempCustomer = customerService.getCustomer(customerId);

  // throw exception if null

  if (tempCustomer == null) {
    throw new CustomerNotFoundException("Customer id not found - " + customerId);
  }

  customerService.deleteCustomer(customerId);

  return "Deleted customer id - " + customerId;
}
```

Delete Customer - Refactor

File: StudentRestController.java



```
@DeleteMapping("/customers/{customerId}")
public String deleteCustomer(@PathVariable int customerId) {

  Customer tempCustomer = customerService.getCustomer(customerId);

  // throw exception if null

  if (tempCustomer == null) {
    throw new CustomerNotFoundException("Customer id not found - " + customerId);
  }

  customerService.deleteCustomer(customerId);

  return "Deleted customer id - " + customerId;
}
```

Check to see if customer exists

If not, throw exception

Happy Path