

# Spring CRM REST - Delete Customer





# Real-Time Project

HTTP Method		CRUD Action
POST	/api/customers	<u>C</u> reate a new customer
GET	/api/customers	<u>R</u> ead a list of customers
GET	/api/customers/{customerId}	<u>R</u> ead a single customer
PUT	/api/customers	<u>U</u> pdate an existing customer
DELETE	/api/customers/{customerId}	<u>D</u> delete an existing customer



# Real-Time Project

Checkpoint

HTTP Method		CRUD Action
POST	/api/customers	<u>C</u> reate a new customer
GET	/api/customers	<u>R</u> ead a list of customers
GET	/api/customers/{customerId}	<u>R</u> ead a single customer
PUT	/api/customers	<u>U</u> pdate an existing customer
DELETE	/api/customers/{customerId}	<u>D</u> eleate an existing customer



# Real-Time Project

Checkpoint

HTTP Method			CRUD Action
✓	POST	/api/customers	<u>C</u> reate a new customer
✓	GET	/api/customers	<u>R</u> ead a list of customers
✓	GET	/api/customers/{customerId}	<u>R</u> ead a single customer
✓	PUT	/api/customers	<u>U</u> pdate an existing customer
	DELETE	/api/customers/{customerId}	<u>D</u> delete an existing customer



# Real-Time Project

Checkpoint

HTTP Method			CRUD Action
✓	POST	/api/customers	<u>C</u> reate a new customer
✓	GET	/api/customers	<u>R</u> ead a list of customers
✓	GET	/api/customers/{customerId}	<u>R</u> ead a single customer
✓	PUT	/api/customers	<u>U</u> pdate an existing customer
✎	DELETE	/api/customers/{customerId}	<u>D</u> elete an existing customer



# Application Interaction

REST  
Client

Customer  
REST  
Controller



# Application Interaction





# Application Interaction





# 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

File: StudentRestController.java

Check to see if customer exists

If not, throw exception

```
@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

Check to see if customer exists

If not, throw exception

```
@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;
}
```

Happy Path