

# Spring CRM REST - Update 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> 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> 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

REST  
Client

PUT

/api/customers

```
{  
  "id" : 1,  
  "firstName" : "Daniel",  
  "lastName" : "Vega",  
  "email" : "daniel.vega@luv2code.com"  
}
```

Customer  
REST  
Controller



# Application Interaction

ID of customer to update  
With updated info

PUT

/api/customers

```
{  
  "id" : 1,  
  "firstName" : "Daniel",  
  "lastName" : "Vega",  
  "email" : "daniel.vega@luv2code.com"  
}
```

REST  
Client

Customer  
REST  
Controller



# Application Interaction

ID of customer to update  
With updated info

REST  
Client

PUT

/api/customers

```
{  
  "id" : 1,  
  "firstName" : "Daniel",  
  "lastName" : "Vega",  
  "email" : "daniel.vega@luv2code.com"  
}
```

Customer  
REST  
Controller

```
{  
  "id" : 1,  
  "firstName" : "Daniel",  
  "lastName" : "Vega",  
  "email" : "daniel.vega@luv2code.com"  
}
```



# Application Interaction

ID of customer to update  
With updated info

PUT

/api/customers

```
{  
  "id" : 1,  
  "firstName" : "Daniel",  
  "lastName" : "Vega",  
  "email" : "daniel.vega@luv2code.com"  
}
```

REST  
Client

Customer  
REST  
Controller

```
{  
  "id" : 1,  
  "firstName" : "Daniel",  
  "lastName" : "Vega",  
  "email" : "daniel.vega@luv2code.com"  
}
```

Response contains  
updated info (echoed)



# Update Customer

File: CustomerRestController.java

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

    // add mapping for PUT /customers - update existing customer

    @PutMapping("/customers")
    public Customer updateCustomer(@RequestBody Customer theCustomer) {

    }
}
```



# Update Customer

File: CustomerRestController.java

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

    // add mapping for PUT /customers - update existing customer

    @PutMapping("/customers")
    public Customer updateCustomer(@RequestBody Customer theCustomer) {

        customerService.saveCustomer(theCustomer);

        return theCustomer;
    }
}
```