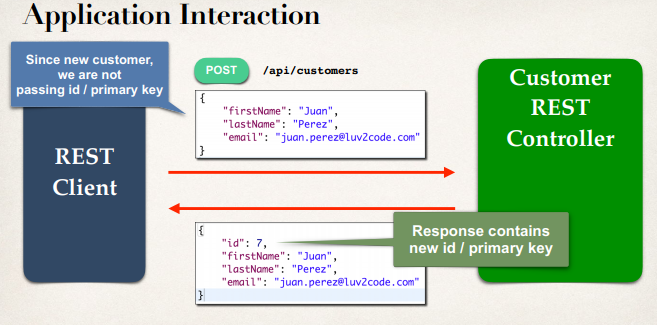
**66. Spring REST - CRUD Database Project - Add Customer**

**Add Customer**:

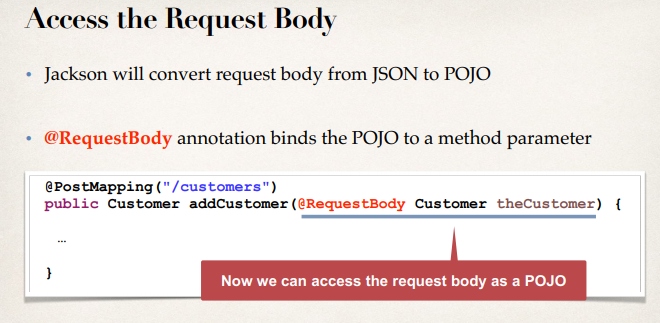
Here we will add a customer in our database. Since we add new customer, we don’t need to add the id. In the background our system will generate this automatically. After adding data successfully in database, the system will send back the data to the customer.



**Access the Request Body**:

* Jackson will convert request body from JSON to POJO
* **@RequestBody** annotation binds the POJO to a method parameter

@PostMapping("/customers")public Customer addCustomer(@RequestBody Customer theCustomer) { …}



**Add Customer**:

**public** **class** CustomerRestController {

...

// add mapping for POST /customers - add new customers

@PostMapping("/customers")

**public** Customer addCustomer(@RequestBody Customer theCustomer) {

// also just in case the pass an id in JSON ... set id to 0

// this is force a save of new item ... instead of update

theCustomer.setId(0);

customerService.saveCustomer(theCustomer);

**return** theCustomer;

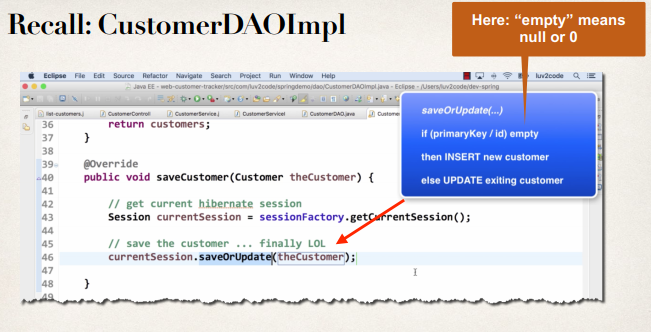
}

}

**What we set the Customer id zero**:

* In the REST controller, we explicitly set the customer id to 0
* Because our backend DAO code uses Hibernate method
  + **session.saveOrUpdate(….)**

**Recall: CustomerDAOImpl**:



**Adding customer with HTTP POST**:

• If REST client is sending a request to “add”, using HTTP POST

• Then we ignore any id sent in the request

• We overwrite the id with 0, to effectively set it to null/empty

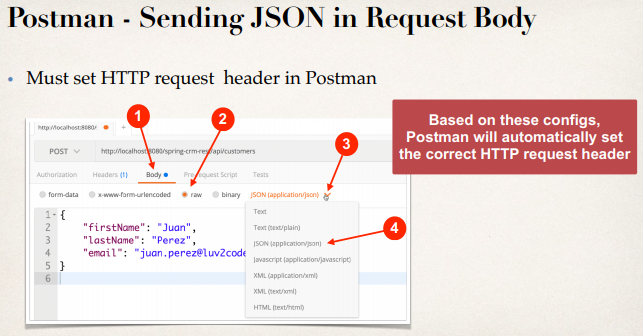
• Then our backend DAO code will “INSERT” new customer

**Sending JSON to Spring REST Controllers**:

* When sending JSON data to Spring REST Controllers
* For controller to process JSON data, need to set a HTTP request header
  + Content-type: application/JSON
* Need to configure REST client to send the correct HTTP request header

**Postman - Sending JSON in Request Body**:

Must set HTTP request handler in Postman



66. Spring REST - CRUD Database Project - Add Customer