# Spring boot with Hibernate

Entity Classes

```
package com.jbk.entity;

// Generated Jul 17, 2020 8:16:54 AM by Hibernate Tools 5.2.3.Final


import java.util.Date;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import static javax.persistence.GenerationType.IDENTITY;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;


/**
 * Employee generated by hbm2java
 */
@Entity
@Table(name = "employee", catalog = "springbootdb")
public class Employee implements java.io.Serializable {
    private Integer id;

    private Country country;

    private String name;

    private String phoneno;

    private String departmentit;

    private String status;

    private Date createddtm;

    private String createdby;

    private Date updateddtm;

    private String updatedby;


    public Employee() {

    }


    public Employee(Country country, Date createddtm, Date updateddtm) {

        this.country = country;

        this.createddtm = createddtm;

        this.updateddtm = updateddtm;

    }


    public Employee(Country country, String name, String phoneno, String
departmentit, String status, Date createddtm,

            String createdby, Date updateddtm, String updatedby) {

        this.country = country;

        this.name = name;

        this.phoneno = phoneno;
```

```
		this.departmentit = departmentit;
		this.status = status;
		this.createddtm = createddtm;
		this.createdby = createdby;
		this.updateddtm = updateddtm;
		this.updatedby = updatedby;
	}

	@Id
	@GeneratedValue(strategy = IDENTITY)

	@Column(name = "id", unique = true, nullable = false)
	public Integer getId() {
		return this.id;
	}

	public void setId(Integer id) {
		this.id = id;
	}

	@ManyToOne(fetch = FetchType.EAGER)
	@JoinColumn(name = "cid", nullable = false)
	public Country getCountry() {
		return this.country;
	}
```

```
public void setCountry(Country country) {
        this.country = country;
}

@Column(name = "name", length = 45)
public String getName() {
        return this.name;
}

public void setName(String name) {
        this.name = name;
}

@Column(name = "phoneno", length = 45)
public String getPhoneno() {
        return this.phoneno;
}

public void setPhoneno(String phoneno) {
        this.phoneno = phoneno;
}

@Column(name = "departmentit", length = 45)
public String getDepartmentit() {
        return this.departmentit;
}

public void setDepartmentit(String departmentit) {
        this.departmentit = departmentit;
}

@Column(name = "status", length = 45)
public String getStatus() {
        return this.status;
}

public void setStatus(String status) {
        this.status = status;
}

@Temporal(TemporalType.TIMESTAMP)
@Column(name = "createddtm", nullable = false, length = 19)
public Date getCreateddtm() {
        return this.createddtm;
}

public void setCreateddtm(Date createddtm) {
        this.createddtm = createddtm;
}
```

```
@Column(name = "createdby", length = 45)
public String getCreatedby() {
        return this.createdby;
}

public void setCreatedby(String createdby) {
        this.createdby = createdby;
}

@Temporal(TemporalType.TIMESTAMP)
@Column(name = "updateddtm", nullable = false, length = 19)
public Date getUpdateddtm() {
        return this.updateddtm;
}

public void setUpdateddtm(Date updateddtm) {
        this.updateddtm = updateddtm;
}

@Column(name = "updatedby", length = 45)
public String getUpdatedby() {
        return this.updatedby;
}
```

```
	public void setUpdatedby(String updatedby) {
		this.updatedby = updatedby;
	}

}
```

---

```
package com.jbk.entity;
// Generated Jul 17, 2020 8:16:54 AM by Hibernate Tools 5.2.3.Final

import java.util.HashSet;
import java.util.Set;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import static javax.persistence.GenerationType.IDENTITY;
import javax.persistence.Id;
import javax.persistence.OneToMany;
import javax.persistence.Table;

/**
 * Country generated by hbm2java
 */
@Entity
@Table(name = "country", catalog = "springbootdb")
```

```
public class Country implements java.io.Serializable {

        private Integer cid;
        private String cname;
        //private Set employees = new HashSet(0);

        public Country() {
        }

        public Country(String cname) {
                this.cname = cname;
        }
/*
        public Country(String cname, Set employees) {
                this.cname = cname;
                this.employees = employees;
        }
*/
        @Id
        @GeneratedValue(strategy = IDENTITY)

        @Column(name = "cid", unique = true, nullable = false)
        public Integer getCid() {
                return this.cid;
        }
```

```
	public void setCid(Integer cid) {
		this.cid = cid;
	}

	@Column(name = "cname", nullable = false, length = 45)
	public String getCname() {
		return this.cname;
	}

	public void setCname(String cname) {
		this.cname = cname;
	}

	/*
	@OneToMany(fetch = FetchType.LAZY, mappedBy = "country")
	public Set getEmployees() {
		return this.employees;
	}

	public void setEmployees(Set employees) {
		this.employees = employees;
	}*/

}
```

**Controller code:-**

```
import com.jbk.SpringBootWithHibernateDemo.service.EmployeeService;
import com.jbk.entity.Country;
import com.jbk.entity.Employee;
@RestController
public class EmployeeController {
	@Autowired
	EmployeeService service;
	//API 1-
	//url:localhost:8080/showallemployee
	// method:Get
	@RequestMapping("/showallemployee")
	public List<Employee> showallemployee()
	{
		List<Employee> listemp=service.showallemployee();
		return listemp;

	}
	//API 3
	//url:localhost:8080/showemployeebyid/{id}
	// method:Get

	@RequestMapping("/showemployeebyid/{id}")
	public List<Employee> showemployeebyid(@PathVariable int id)
```

```
    {

        List<Employee> listemp=service.showemployeebyid(id);
        return listemp;
    }
    //API 4
    //url:localhost:8080/showallemployeebyname/{name}
    // method:Get

    @RequestMapping("/showemployeebyname/{name}")
    public List<Employee> showemployeebyname(@PathVariable String name)
    {

        List<Employee> listemp=service.showemployeebyname(name);

        return listemp;
    }
    //API 11
     //url:localhost:8080/showemployeebydepartment/{department}
    // method:Get

    @RequestMapping("/showemployeebydepartment/{department}")
    public List<Employee> showallemployeebydepartment(@PathVariable
String department)
    {
```

```
                List<Employee>
listemp=service.showallemployeebydepartment(department);

                return listemp;
        }



        //API 2
        //url:localhost:8080/showallemployeebystatus/{status}
        // method:Get

        @RequestMapping("/showemployeebystatus/{status}")
        public List<Employee> showallemployeebystatus(@PathVariable String
status)
        {



                List<Employee> listemp=service.showallemployeebystatus(status);

                return listemp;
        }
        //API 6
        //url:localhost:8080/addemployee
        // method:Post

        @PostMapping("/addemployee")
```

```
public String addEmployee(@RequestBody Employee emp)
{
        String result=service.addEmployee(emp);
        return result;

}




//API 7
//url:localhost:8080/addcountry
// method:Post

@PostMapping("/addcountry")
public String addCountry(@RequestBody Country ctr)
{
        String result=service.addCountry(ctr);
        return result;
}

//API-10
//url:localhost:8080/deleteemployeebyid/{id}
// method:Delete
```

```
@DeleteMapping("/deleteemployeebyid/{id}")
public String deleteEmployeebyid(@PathVariable int id)
{
	String result=service.deleteEmployeebyid(id);
	return result;
}
//API12
//url:localhost:8080/deletecountrybyid{cid}
// method:Delete

@DeleteMapping("/deletecountrybyid/{countryid}")
public String deleteCountrybyid(@PathVariable int countryid)
{
	String result=service.deleteCountrybyid(countryid);
	return result;
}

//API-9
//url:localhost:8080/deletecountrybyname/{cname}
// method:Delete

@DeleteMapping("/deletecountrybyname/{cname}")
public String deleteCountrybyName(@PathVariable String cname)
{
        String result=service.deleteCountrybyName(cname);
        return result;
}
//API-13
//url:localhost:8080/getcountrybycid/{cid}
// method:Get

@RequestMapping("/getcountrybyid/{cid}")
public List<Country> getCountrybyId(@PathVariable int cid)
{
        List<Country> listcountry=service.getCountrybyId(cid);
        return listcountry;
}



//API-8
//url:localhost:8080/updatecountry
// method:Put

@RequestMapping(value="/updatecountry", method=RequestMethod.PUT)
public String updateCountry(@RequestBody Country country)
```

```
    {
        String result=service.updateCountry(country);
        return result;
    }
    //API 5
    //url:localhost:8080/getemployeelistbeforetoday
    // method:Get

    @RequestMapping("/getemployeebeforetoday")
    public List<Employee> getemployeeBeforeToday()
    {
        List<Employee> listemp=service.getemployeeBeforeToday();
        return listemp;
    }

    //API14
    //url:localhost:8080/updateemployee
    // method:Put
    @RequestMapping(value="/updateemployee",
method=RequestMethod.PUT)
    public String  updateEmployee(@RequestBody Employee employee)
    {
        String msg=service.updateEmployee(employee);
        return msg;
    }
    //API 15
```

```
        //url:localhost:8080/showemployeebycountryname/{cname}

        // method:Get



        @RequestMapping("/showemployeebbycounrtryname/{cname}")

        public List<Employee> showallemployeebycountryname(@PathVariable
String cname)

        {

                List<Employee>
listemp=service.showallemployeebycountryname(cname);

                return listemp;

        }



}
```

**Service class code:-**

```
package com.jbk.SpringBootWithHibernateDemo.service;

import java.util.List;


import com.jbk.entity.Country;
import com.jbk.entity.Employee;

@Service
public class EmployeeService {

        @Autowired
        EmployeeDao dao;

        public List<Employee> showallemployee() {

                List<Employee> listemp=dao.showallemployee();
                return listemp;

        }

        public List<Employee> showemployeebyid(int id) {
                List<Employee> listemp=dao.showemployeebyid(id);
```

```
		return  listemp;
	}

	public List<Employee> showemployeebyname(String name) {
		List<Employee> listemp=dao.showemployeebyname(name);
		return  listemp;
	}

	public List<Employee> showallemployeebydepartment(String department) {
		List<Employee> listemp=dao.showallemployeebydepartment(department);
		return  listemp;
	}

	public List<Employee> showallemployeebystatus(String status) {
		List<Employee> listemp=dao.showallemployeebystatus(status);
		return  listemp;
	}

	public String addEmployee(Employee emp) {
		String result=dao.addEmployee(emp);
		return result;
	}

	public String addCountry(Country ctr) {
		String result=dao.addCountry(ctr);
```

```
        return result;

}

public String deleteEmployeebyid(int id) {
        String result=dao.deleteEmployeebyid(id);
        return result;
}

public String deleteCountrybyid(int countryid) {
        String result=dao.deleteCountrybyid(countryid);
        return result;
}

public String deleteCountrybyName(String cname) {
        String result=dao.deletecountrybyName(cname);
        return result;
}

public List<Country> getCountrybyId(int cid) {
        List<Country>listcountry=dao.getCountrybyId(cid);
        return listcountry;
}

public String updateCountry(Country country) {
        String result=dao.updateCountry(country);
        return result;
    }

    public List<Employee> getemployeeBeforeToday() {
        List<Employee> listemp=dao.getemployeeBeforeToday();
        return listemp;
    }

    public String updateEmployee(Employee employee) {
        String msg=dao.updateEmployee(employee);

        return msg;
    }

    public List<Employee> showallemployeebycountrybyid(int cid) {
            List<Employee>
listemp=dao.showallemployeebycountrybyid(cid) ;
        return listemp;
    }

    public List<Employee> showallemployeebycountryname(String cname) {
        List<Employee>
listemp=dao.showallemployeebycountryname(cname) ;
        return listemp;
    }
```

```
}
```

**Dao Class code:-**

```
import java.util.Date;
import java.util.List;
import com.jbk.entity.Country;
import com.jbk.entity.Employee;

@Repository
public class EmployeeDao {

	@Autowired
	SessionFactory sessionFactory;

	public List<Employee> showallemployee() {

		Session session=sessionFactory.openSession();
		Criteria criteria=session.createCriteria(Employee.class);
		List<Employee> listemp=criteria.list();
		return listemp;
	}
	public List<Employee> showemployeebyid( int id)
	{
		Session session=sessionFactory.openSession();
		Criteria criteria=session.createCriteria(Employee.class);
		criteria.add(Restrictions.eq("id", id));
```

```
        List<Employee> listemp=criteria.list();

        return listemp;
}

public List<Employee> showemployeebyname(String name)
{
        Session session=sessionFactory.openSession();
        Criteria criteria=session.createCriteria(Employee.class);
        criteria.add(Restrictions.eq("name", name));

        List<Employee> listemp=criteria.list();

        return listemp;
}

public List<Employee> showallemployeebydepartment(String department)
{
        Session session=sessionFactory.openSession();
        Criteria criteria=session.createCriteria(Employee.class);
        criteria.add(Restrictions.eq("departmentit", department));

        List<Employee> listemp=criteria.list();
        return listemp;
}

public List<Employee> showallemployeebystatus(String status)
{
        Session session=sessionFactory.openSession();
        Criteria criteria=session.createCriteria(Employee.class);
        criteria.add(Restrictions.eq("status",status));

        List<Employee> listemp=criteria.list();

        return listemp;
}
public String addEmployee(Employee emp) {

        System.out.println(emp.getName());

        Country country=new Country();
        country.setCid(emp.getCountry().getCid());
        country.setCname(emp.getCountry().getCname());



        Employee emp1=new Employee();

        String name=emp.getName();
        emp1.setName(name);
        emp1.setPhoneno(emp.getPhoneno());
        emp1.setDepartmentit(emp.getDepartmentit());
        emp1.setStatus(emp.getStatus());
        emp1.setCreateddtm(emp.getCreateddtm());
        emp1.setCreatedby(emp.getCreatedby());
        emp1.setUpdateddtm(emp.getUpdateddtm());
        emp1.setUpdatedby(emp.getUpdatedby());
        emp1.setCountry(country);

        Session session=sessionFactory.openSession();
        Transaction transaction=session.beginTransaction();
        session.save(emp1);
        transaction.commit();

        return "Employee Inserted Successfully";
}
public String addCountry(Country ctr) {


        System.out.println(ctr.getCname());

        Country country=new Country();
        country.setCname(ctr.getCname());
        Session session=sessionFactory.openSession();
        Transaction transaction=session.beginTransaction();
        session.save(country);
        transaction.commit();
        return "Country Added Successfully";
    }
    public String deleteEmployeebyid(int id) {

        System.out.println("Id"+id);

        Session session=sessionFactory.openSession();
        Transaction transaction=session.beginTransaction();
        Query query=session.createQuery("delete from Employee where
id=:id");
        query.setParameter("id",id);

        int i=query.executeUpdate();
        transaction.commit();
        if(i>0)
        {
            return "Employee deleted Succefully";
        }
        else
        {
            return "Something Went wrong";
        }
    }
    public String deleteCountrybyid(int countryid) {


        System.out.println("Country id:- "+countryid);
        Session session=sessionFactory.openSession();
        Transaction transaction=session.beginTransaction();


    Query query=session.createQuery("delete from Country where cid=:countryid");
    query.setParameter("countryid", countryid);
    int i=query.executeUpdate();
    transaction.commit();
        if(i>0)
        {
            return "Country deleted Successfully!"+countryid+"country id";
        }
        else
        {
            return "Something went Wrong!";
        }



    }
    public String deletecountrybyName(String cname) {
        System.out.println("Country Name:-"+cname);
        Session session=sessionFactory.openSession();
        Transaction transaction=session.beginTransaction();
        Query query=session.createQuery("delete from Country where cname=:cname");
        query.setParameter("cname", cname);
        int i=query.executeUpdate();
        transaction.commit();
        if(i>0)
        {
            return " "+cname+" Country deleted successfully";
        }
        else
        {
            return "something went wrong";
        }



    }
    public List<Country> getCountrybyId(int cid) {

        Session session=sessionFactory.openSession();
        Criteria criteria=session.createCriteria(Country.class);
        criteria.add(Restrictions.eq("cid", cid));
```

```
		List<Country> listcountry=criteria.list();
		return listcountry;




	}
	public String updateCountry(Country country) {
		System.out.println(country.getCid());
		System.out.println(country.getCname());
		String countryname=country.getCname();
		Session session=sessionFactory.openSession();
		Transaction transaction=session.beginTransaction();
		Query query=session.createQuery("update Country set cname=:cname where cid=:cid");
		query.setParameter("cname",country.getCname());
		query.setParameter("cid", country.getCid());
		int result=query.executeUpdate();
		transaction.commit();
		if(result>0)
		{
			return "country "+countryname+" Updated Successfully";
		}
		else
		{
			return "something Went Wrong";
		}
	}
	public List<Employee> getemployeeBeforeToday() {
		
		Date date=new Date();
		System.out.println(date);
		
		
		Session session=sessionFactory.openSession();
		Query query=session.createQuery("from Employee where createddtm
<:date");
		query.setParameter("date", date);
		List<Employee> listemp=query.list();
		System.out.println(listemp);
		return listemp;
	}
	public String updateEmployee(Employee employee) {
		int id=employee.getId();
		System.out.println(employee.getName());
		Session session=sessionFactory.openSession();
		Transaction transaction=session.beginTransaction();
		Query query=session.createQuery("update Employee set name=:name
,phoneno=:phoneno,departmentit=:departmentit,status=:status,createddtm=:created
dtm,createdby=:createdby,updateddtm=:updateddtm,updatedby=:updatedby,countr
y=:country where id=:id  ");
		query.setParameter("name", employee.getName());
```

```
        query.setParameter("phoneno", employee.getPhoneno());
        query.setParameter("departmentit", employee.getDepartmentit());
        query.setParameter("status", employee.getStatus());
        query.setParameter("createddtm", employee.getCreateddtm());
        query.setParameter("createdby", employee.getCreatedby());
        query.setParameter("updateddtm", employee.getUpdateddtm());
        query.setParameter("updatedby", employee.getUpdatedby());
        query.setParameter("country", employee.getCountry());
        query.setParameter("id", employee.getId());
        int i=query.executeUpdate();
        transaction.commit();
        if(i>0)
        {
                return "Employee "+id+" Updated Successfully!";
        }
        else
        {
                return "Something went Wrong!";
        }

}
public List<Employee> showallemployeebycountrybyid(int cid) {
         Session session=sessionFactory.openSession();
                Criteria criteria=session.createCriteria(Employee.class);
                Criteria criteria1=session.createCriteria(Country.class);
        Country country=session.get(Country.class,cid);
        System.out.println(country.getCname());
        criteria.add(Restrictions.eq("country",country));
        List<Employee> listemp=criteria.list();

        return listemp;
}
public List<Employee> showallemployeebycountryname(String cname) {
    Session session=sessionFactory.openSession();
        Criteria criteria=session.createCriteria(Country.class);
        criteria.add(Restrictions.eq("cname",cname));
        int cid=0;
        String country_name=null;

        List<Country> listcountry=criteria.list();
        for(Country country:listcountry)
        {
            cid=country.getCid();
          System.out.println(cid);
          country_name=country.getCname();
         System.out.println(country_name);
        }
        Country ctr=new Country();
        ctr.setCid(cid);
        ctr.setCname(cname);
```

```
                Criteria criteria_emp=session.createCriteria(Employee.class);
                 criteria_emp.add(Restrictions.eq("country", ctr));
                 List<Employee>listemp=criteria_emp.list();


                return listemp;
        }



}
```