

Implement paging and sorting for the Customer entity as follows:

Routes:

- <http://localhost:8080/customers?pageSize=5>
- <http://localhost:8080/customers?pageSize=5&pageNo=1>
- <http://localhost:8080/customers?pageSize=5&pageNo=2>
- <http://localhost:8080/customers?pageSize=5&pageNo=1&sortBy=email>
- <http://localhost:8080/customers?pageSize=5&pageNo=1&sortBy=firstName>

Customer Entity :-

```
package com.PortalAssimentrestapi.model;

import java.time.Instant;

import org.springframework.data.annotation.CreatedDate;
import org.springframework.data.annotation.LastModifiedDate;

import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;

@Entity
public class Customer {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @Column
    private String firstName;

    @Column
    private String lastName;

    @Column
    private String email;

    @Column
    private double salary;

    @Column
    private String address;

    public String getAddress() {
        return address;
    }
}
```

```
        public void setAddress(String address) {
            this.address = address;
        }

        public Long getId() {
            return id;
        }

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

        public String getFirstName() {
            return firstName;
        }

        public void setFirstName(String firstName) {
            this.firstName = firstName;
        }

        public String getLastName() {
            return lastName;
        }

        public void setLastName(String lastName) {
            this.lastName = lastName;
        }

        public String getEmail() {
            return email;
        }

        public void setEmail(String email) {
            this.email = email;
        }

        public double getSalary() {
            return salary;
        }

        public void setSalary(double salary) {
            this.salary = salary;
        }
    }
}
```

CustomerController:-

```
package com.PortalAssimentrestapi.Controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.data.domain.Page;
import org.springframework.web.bind.annotation.*;

import com.PortalAssimentrestapi.Service.CustomerService;
import com.PortalAssimentrestapi.model.Customer;

import java.util.List;

@RestController
@RequestMapping("/customers")
public class CustomerController {

    @Autowired
    private CustomerService customerService;

    @PostMapping
    public Customer createCustomer(@RequestBody Customer customer) {
        return customerService.createCustomer(customer);
    }

    @GetMapping("/{id}")
    public Customer getCustomerById(@PathVariable Long id) {
        return customerService.getCustomerById(id);
    }

    @GetMapping
    public List<Customer> getAllCustomers() {
        return customerService.getAllCustomers();
    }

    @PutMapping("/{id}")
    public Customer updateCustomer(@PathVariable Long id, @RequestBody Customer
customer) {
        return customerService.updateCustomer(id, customer);
    }

    @DeleteMapping("/{id}")
    public void deleteCustomer(@PathVariable Long id) {
        customerService.deleteCustomer(id);
    }

    @GetMapping("/search")
    public List<Customer> searchCustomers(
```

```

        @RequestParam(required = false) String firstName,
        @RequestParam(required = false) String lastName,
        @RequestParam(required = false) String email,
        @RequestParam(required = false) String address
    ) {
        if (firstName != null) {
            return customerService.getAllCustomers().stream()
                .filter(c -> c.getFirstName().equalsIgnoreCase(firstName))
                .toList();
        }

        return List.of();
    }

    @GetMapping("/customers")
    public Page<Customer> getCustomers(
        @RequestParam(defaultValue = "0") int pageNo,
        @RequestParam(defaultValue = "1") int pageSize,
        @RequestParam(defaultValue = "id") String sortBy) {
        return customerService.getCustomers(pageNo, pageSize, sortBy);
    }
}

```

CustomerService :-

```

package com.PortalAssimentrestapi.Service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.data.domain.Page;
import org.springframework.data.domain.PageRequest;
import org.springframework.data.domain.Pageable;
import org.springframework.data.domain.Sort;
import org.springframework.stereotype.Service;

import com.PortalAssimentrestapi.Repository.CustomerRepository;
import com.PortalAssimentrestapi.model.Customer;

@Service
public class CustomerService {

    @Autowired
    private CustomerRepository customerRepository;

    public Customer createCustomer(Customer customer) {
        return customerRepository.save(customer);
    }
}

```

```

public Customer getCustomerById(Long id) {
    return customerRepository.findById(id).orElse(null);
}

public List<Customer> getAllCustomers() {
    return customerRepository.findAll();
}

public Customer updateCustomer(Long id, Customer customerDetails) {
    Customer customer = getCustomerById(id);
    if (customer != null) {
        customer.setFirstName(customerDetails.getFirstName());
        customer.setLastName(customerDetails.getLastName());
        customer.setEmail(customerDetails.getEmail());
        customer.setAddress(customerDetails.getAddress());
        return customerRepository.save(customer);
    }
    return null;
}

public void deleteCustomer(Long id) {
    customerRepository.deleteById(id);
}

public Page<Customer> getCustomers(int pageNo, int pageSize, String sortBy) {
    Pageable pageable = PageRequest.of(pageNo, pageSize, Sort.by(sortBy).ascending());
    return customerRepository.findAll(pageable);
}
}

```

CustomerRepository :-

```

package com.PortalAssimentrestapi.Repository;

import java.util.List;

import org.springframework.data.jpa.repository.JpaRepository;

import com.PortalAssimentrestapi.model.Customer;

public interface CustomerRepository extends JpaRepository<Customer, Long> {

    List<Customer> findByFirstName(String firstName);
    List<Customer> findByLastName(String lastName);
}

```

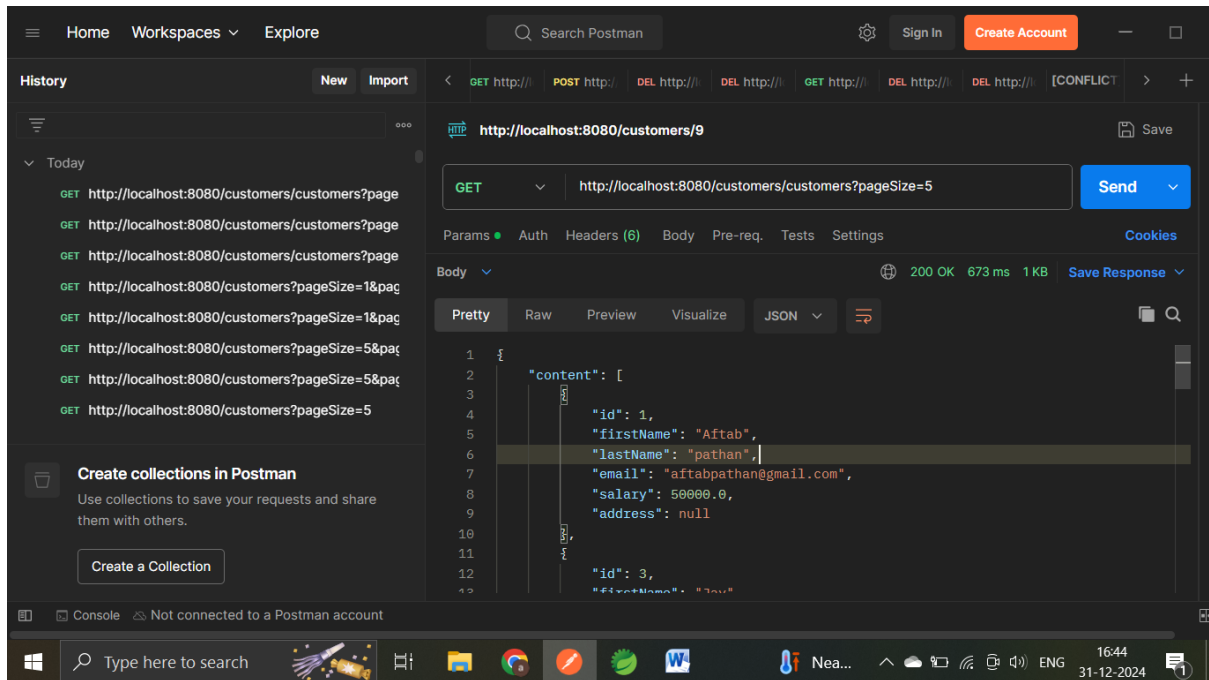
```

List<Customer> findByEmail(String email);
List<Customer> findByAddress(String address);
}

```

OUTPUT:-

1-



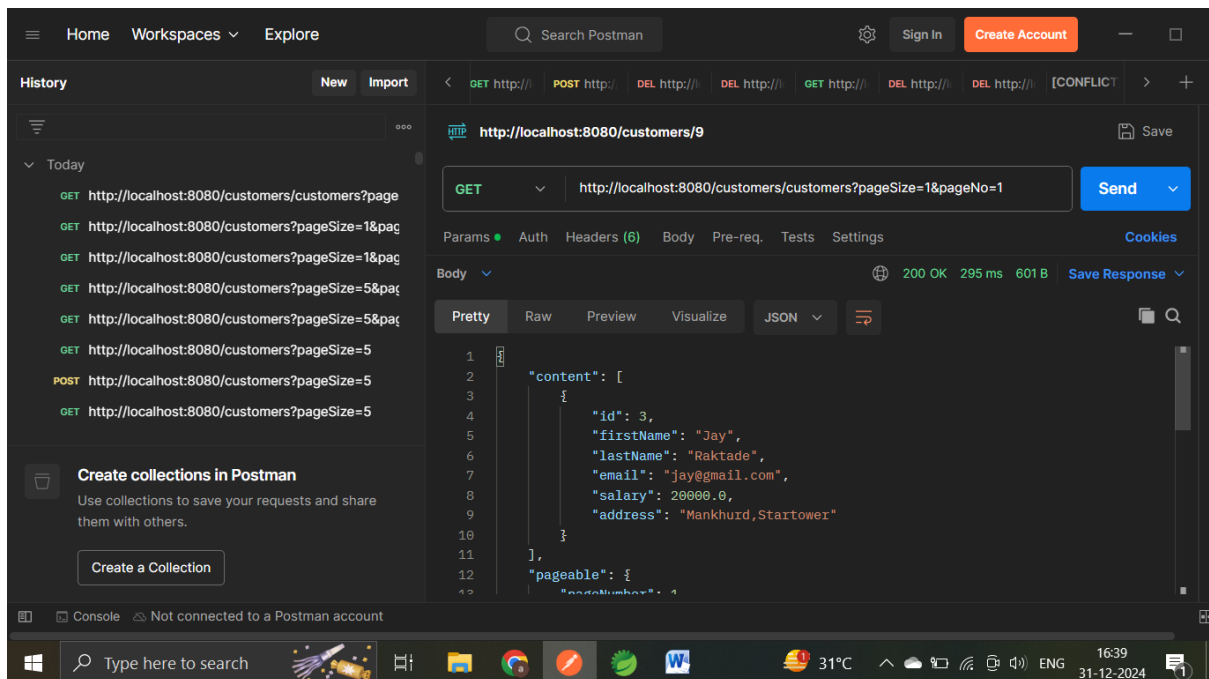
```

{
  "content": [
    {
      "id": 1,
      "firstName": "Aftab",
      "lastName": "pathan",
      "email": "aftabpathan@gmail.com",
      "salary": 50000.0,
      "address": null
    },
    {
      "id": 3,
      "firstName": "Jay",
      "lastName": "Raktade",
      "email": "jay@gmail.com",
      "salary": 20000.0,
      "address": "Mankhurd, Startower"
    },
    {
      "id": 4,
      "firstName": "Haris",

```

```
        "lastName": "Ansari",
        "email": "Hars@gmail.com",
        "salary": 40000.0,
        "address": null
    },
    {
        "id": 5,
        "firstName": "ram",
        "lastName": "khade",
        "email": "ram@gmail.com",
        "salary": 25000.0,
        "address": null
    },
    {
        "id": 6,
        "firstName": "Raj",
        "lastName": "bhor",
        "email": "RajBhor@gmail.com",
        "salary": 70000.0,
        "address": null
    }
],
"pageable": {
    "pageNumber": 0,
    "pageSize": 5,
    "sort": {
        "empty": false,
        "sorted": true,
        "unsorted": false
    },
    "offset": 0,
    "paged": true,
    "unpaged": false
},
"last": false,
"totalElements": 8,
"totalPages": 2,
"size": 5,
"number": 0,
"sort": {
    "empty": false,
    "sorted": true,
    "unsorted": false
},
"first": true,
"numberOfElements": 5,
"empty": false
}
```

2-



```
{
  "content": [
    {
      "id": 3,
      "firstName": "Jay",
      "lastName": "Raktade",
      "email": "jay@gmail.com",
      "salary": 20000.0,
      "address": "Mankhurd,Startower"
    }
  ],
  "pageable": {
    "pageNumber": 1,
    "pageSize": 1,
    "sort": {
      "empty": false,
      "sorted": true,
      "unsorted": false
    },
    "offset": 1,
    "paged": true,
    "unpaged": false
  },
  "last": false,
  "totalElements": 8,
  "totalPages": 8,
  "number": 1,
}
```

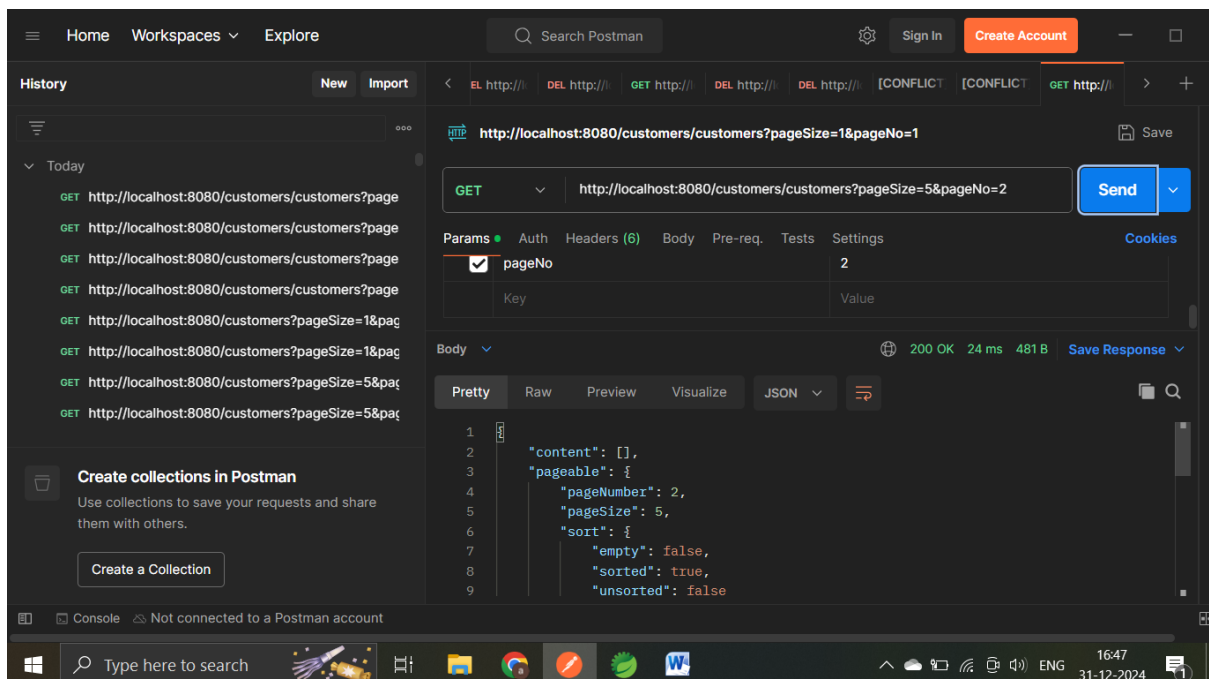


```

    "sort": {
      "empty": false,
      "sorted": true,
      "unsorted": false
    },
    "first": false,
    "numberOfElements": 1,
    "size": 1,
    "empty": false
  }
}

```

3-



```

{
  "content": [],
  "pageable": {
    "pageNumber": 2,
    "pageSize": 5,
    "sort": {
      "empty": false,
      "sorted": true,
      "unsorted": false
    }
  },
  "offset": 10,
  "paged": true,
  "unpaged": false
},
"last": true,
"totalElements": 8,
"totalPages": 2,

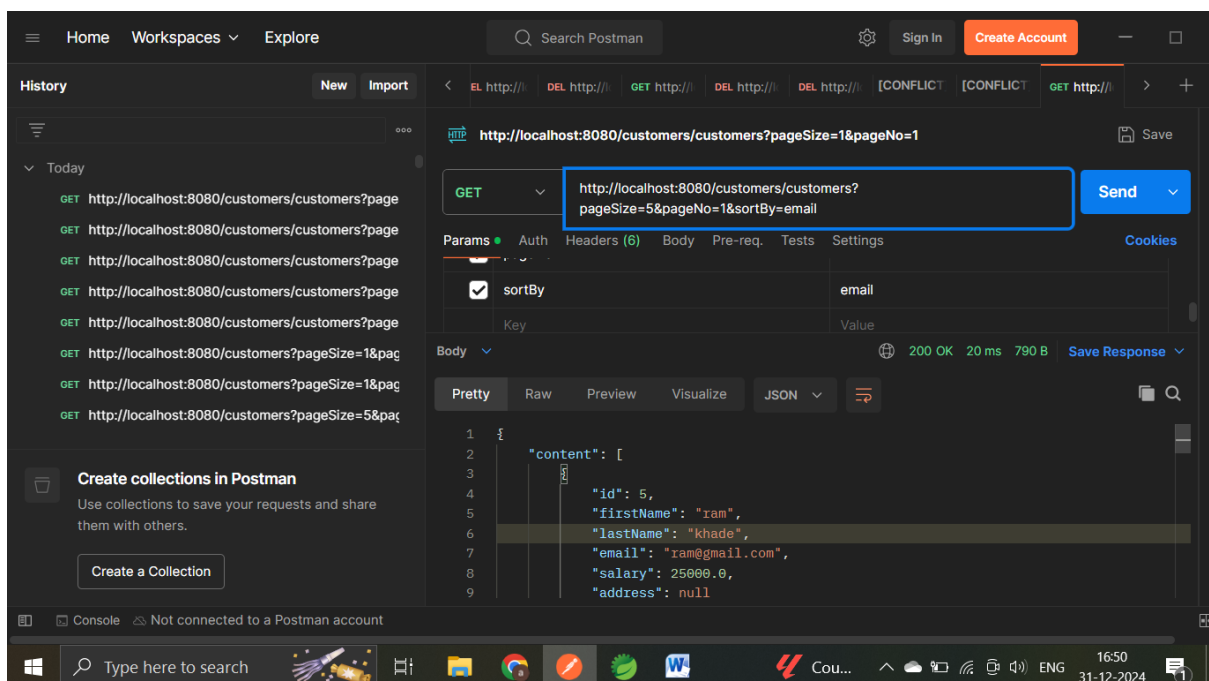
```

```

    "size": 5,
    "number": 2,
    "sort": {
      "empty": false,
      "sorted": true,
      "unsorted": false
    },
    "first": false,
    "numberOfElements": 0,
    "empty": true
  }
}

```

4-

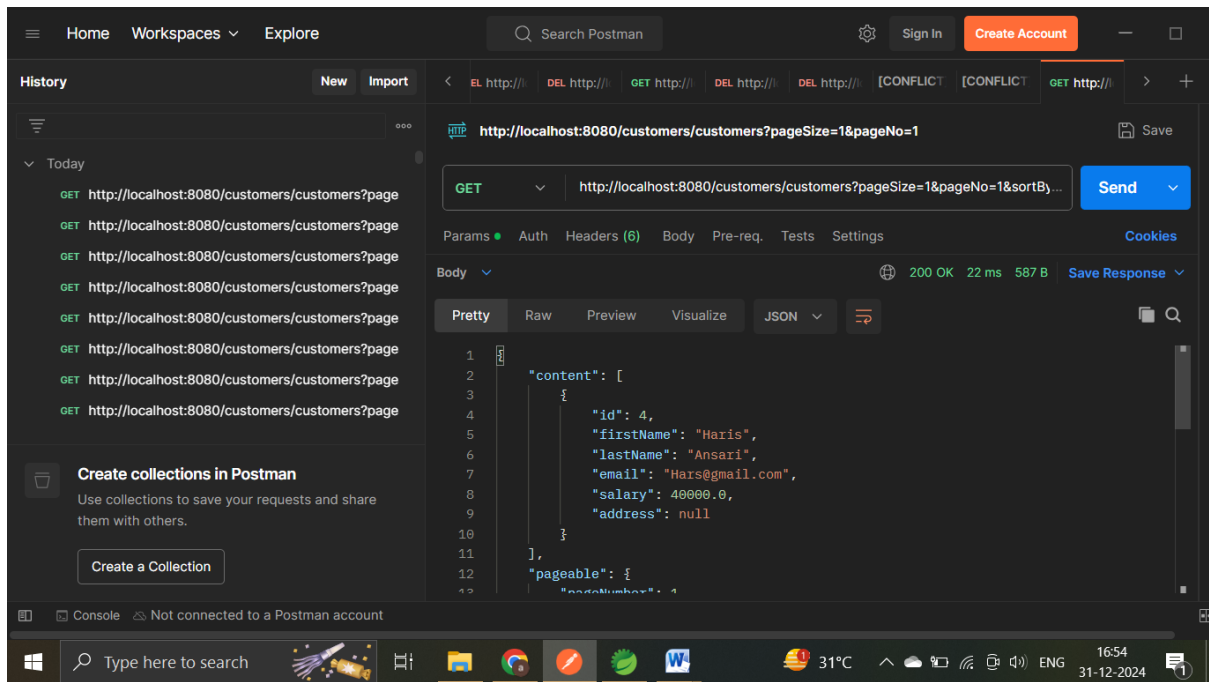


```

{
  "content": [
    {
      "id": 5,
      "firstName": "ram",
      "lastName": "khade",
      "email": "ram@gmail.com",
      "salary": 25000.0,
      "address": null
    },
    {
      "id": 8,
      "firstName": "ram",
      "lastName": "khade",
      "email": "ram@gmail.com",

```

```
        "salary": 25000.0,  
        "address": null  
    },  
    {  
        "id": 12,  
        "firstName": "Zaid",  
        "lastName": "Ansari",  
        "email": "Zaid@gmail.com",  
        "salary": 15000.0,  
        "address": null  
    }  
],  
"pageable": {  
    "pageNumber": 1,  
    "pageSize": 5,  
    "sort": {  
        "empty": false,  
        "sorted": true,  
        "unsorted": false  
    },  
    "offset": 5,  
    "paged": true,  
    "unpaged": false  
},  
"last": true,  
"totalElements": 8,  
"totalPages": 2,  
"size": 5,  
"number": 1,  
"sort": {  
    "empty": false,  
    "sorted": true,  
    "unsorted": false  
},  
"first": false,  
"numberOfElements": 3,  
"empty": false  
}
```



```
{
  "content": [
    {
      "id": 4,
      "firstName": "Haris",
      "lastName": "Ansari",
      "email": "Hars@gmail.com",
      "salary": 40000.0,
      "address": null
    }
  ],
  "pageable": {
    "pageNumber": 1,
    "pageSize": 1,
    "sort": {
      "empty": false,
      "sorted": true,
      "unsorted": false
    },
    "offset": 1,
    "paged": true,
    "unpaged": false
  },
  "last": false,
  "totalElements": 8,
  "totalPages": 8,
  "size": 1,
  "number": 1,
  "sort": {
    "empty": false,
```

```
    "sorted": true,  
    "unsorted": false  
  },  
  "first": false,  
  "numberOfElements": 1,  
  "empty": false  
}
```