**Assignment on Micro services and cloud**

1. Design a Micro service for ProductCatalog

```
@GetMapping
	public List<Product> listAll() {
		return prodService.listAll();
	}

	@PostMapping
	public void saveProduct(@RequestBody Product product) {
		prodService.saveProduct(product);
	}

	@GetMapping(value = "/{id}")
	public Product findById(@PathVariable("id") String id) {
		return prodService.findById(id);
	}

	@DeleteMapping
	public void deleteProduct(@RequestBody Product product) {
		prodService.deleteProduct(product);
	}

	@DeleteMapping("/{id}")
	public void deleteProductById(@PathVariable("id") String id) {
		prodService.deleteById(id);
	}

	@GetMapping(value = "/byName/{name}")
	public List<Product> findByName(@PathVariable("name") String name) {
		return prodService.findByName("%name + "%");
	}

	@GetMapping("/byNameSort/{name}/conf/{sortType}")
	public List<Product> findByAndSort(@PathVariable("name")String
name,@PathVariable("sortType")String sortType ,Sort sort) {
		return prodService.findByAndSort("%"+name+"%", new
Sort(Sort.Direction.ASC,sortType));
	}


	@GetMapping("/byDate/{fromDate}/conf/{toDate}")
	public List<Product> findByDate(@PathVariable("fromDate") String
fromDate, @PathVariable("toDate") String toDate) throws ParseException {

		return prodService.findByDate(new SimpleDateFormat("yyyy-mm-
dd").parse(fromDate), new SimpleDateFormat("yyyy-mm-dd").parse(toDate));
```

```
	}

	@GetMapping(value="/byPrice/{min}/conf/{max}")
	public List<Product> findByPrice(@PathVariable("min") double min,
@PathVariable("max") double max) {
		return prodService.findByPrice(min, max);
	}
```

2. Make sure you are creating Service and Dao classes as per the method signature given from above RestController class.
3. Your application properties file should have following data base information

```
spring.jpa.hibernate.ddl-auto=update
spring.datasource.url=jdbc:mysql://localhost:3306/test
spring.datasource.username=root
spring.datasource.password=root1234
spring.jpa.show-sql=true

server.port=8081
```

   Note: You can configure any database server which is of your choice
4. Create a config server and add above properties file into **product-catalog.properties** At https://github.com/ctsjava/globalconfig web site (or you may use your won)

5. Now write a Config Client program to read all your properties file from the config server and access the end point urls from the above-mentioned controller class.