

Making API Call

```
public int getNews() {
    @Value( "${api.url}" )
    private String uri;
    @Value( "${api.key}" )
    private String key;
    ObjectMapper mapper = new ObjectMapper();
    RestTemplate restTemplate = new
    RestTemplate();
    List<Article> result = restTemplate
    .getForObject(uri + key,
        InboundNewsResponse.class).getArticles();
    return 1
}
```

Rest Template

```
@Bean
public int getRest(RestTemplateBuilder
    restTemplateBuilder) {
    String baseUrl = <URL>
    RestTemplate restTemplate =
    restTemplateBuilder
        .setConnectTimeout(
            (int) TimeUnit.SECONDS.toMillis(10))
        .setReadTimeout(
            (int) TimeUnit.SECONDS.toMillis(10))
        .build();
    return new NCOAClient(baseUrl,
    restTemplate);
}
```

Misc.

Random Number

```
public static double getRandomNumber() {
    double x = Math.random();
    return x;
}
```

Receiving API Call

Basic Get

```
@GetMapping(path = "/enp")
protected boolean get() {
    return true;
}
```

Info From Request Body

```
(@RequestBody String blah)
```

Info From Request Header

```
(@RequestHeader("dp-token") String dpToken)
```

Receive File

```
@RequestMapping(value = "/upload", method =
    RequestMethod.POST,
    consumes = MediaType.MULTIPART_FORM_DATA_VALUE)
```

```
public String fileUpload(@RequestParam("file")
    MultipartFile file) throws IOException {
    File convertedFile = new File("/tmp/" +
        file.getOriginalFilename());
    convertedFile.createNewFile();
    FileOutputStream fout = new FileOutputStream(
        convertedFile);
    fout.write(file.getBytes());
    fout.close();
    return "File is uploaded successfully";
}
```

General

toString

```
@Override
public String toString() {
    return "MailFwdrQueue{" +
        "cfpsid='" + cfpsId +
        "', btid='" + btId +
        "'}";
}
```



Read file

```
import java.io.*;

public class Readin gFr omFile
{
    public static void main(S tring[] args) throws
Exception
    {
        // pass the path to the file as a parameter
        Fil eReader fr =
            new FileRe ade r("C :\ \Use rs \ \pa -
nka j\ \Des kto p\ \tes t.t xt");

        int i;
        while ((i=fr.re ad()) != -1)
            Sys tem.ou t.p rin t(( char) i);
    }
}
```

Run Method on Start

```
@Po stC ons truct
private void init() {
    log.in fo( " Ini tDe moA ppl -
ication initia liz ation logic ...");
    // ...
}
```

Lists, Maps, Trees

Trees

Trees Article

Maps

```
HashMa p<S tring, Intege r> map = new HashMa p<>
();
map.pu t("v ish al", 10);
map.ge t("v ish al");
```

Lists

```
List a = new ArrayL ist();
List b = new Linked List();
List c = new Vector();
```

Lists, Maps, Trees (cont)

```
> List d = new Stack();
List<Integer> l1 = new ArrayList<Integer>();
l1.add(0, 1); // adds 1 at 0 index
l1.add(1, 2); // adds 2 at 1 index
```

Jooq

```
private final DSLContext jooq;

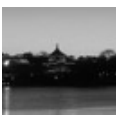
@Trans act ional
public boolean select Exi sts (Long <var> ) {
    return jooq.f etc hEx ists(
        sel ect()
            .fr om( COA _IN FOR MED _DE -
LIVERY)
            .wh ere (<t abl e>.< co lum n>.e -
q( <va r>))
        );
}

@Trans act ion al( rea dOnly = true)
public List<M ail Fwd Ext ens ion> select (String
cfpsId) {
    Lis t<M ail Fwd Ext ens ion sRe cor d>
records = jooq
        .se lec tFr om( MAI L_ F WD_ EXT -
ENS IONS)
        .wh ere (MA IL_ FWD _EX TEN SIO -
NS.C FP SID.eq (cf psId))
        .fe tch Int o(M ail Fwd Ext ens -
ion sRe cor d.c lass);
}
```

OAuth / Auth

h2

```
spring.datasource.url=jdbc:h2:mem:testdb
spring.da tas our ce.d ri ver Cla ssN ame =or g.h -
2.D river
spring.da tas our ce.u se rna me=sa
spring.da tas our ce.p as sword=
spring.h2.co nso le.e na ble d=false
spring.h2.co nso le.e na ble d=true
http:/ /lo cal hos t:8 080 /h2 -co nsole
```



h2 (cont)

```
> @Entity
public class Person {
    @Id
    @GeneratedValue
    private int id;
    private String name;
    private int age;
    private String emailId;
    //getters and setters
}

public interface PersonRepository extends CrudRepository<Person,
Integer> {}

@Service
public class PersonService {
    @Autowired
    PersonRepository personRepository;
    public List<Person> getAllPersons() {
        List<Person> persons = new ArrayList<Person>();
        personRepository.findAll().forEach(person -> persons.add(per-
son));
        return persons;
    }
    public Person getPersonById(int id) {
        return personRepository.findById(id).get();
    }
    public void saveOrUpdate(Person person) {
        personRepository.save(person);
    }
    public void delete(int id) {
        personRepository.deleteById(id);
    }
}
```



By **bigocb**

cheatography.com/bigocb/

Not published yet.

Last updated 25th October, 2019.

Page 3 of 3.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>