0-

1.

```
class Config extends Configuration{
    private static Config config = null;
    private Conf<String> b = Holders.simple("b");
    private ObjectColConf<Collection<Test>> tests = Holders.objCollection(new ArrayList<Test>(), Test.class);
    private ObjectMapConf<Map<String, Test>> expMap = Holders.objMap(new HashMap<String, Test>(), String.class,
Test.class);
        private ColConf<Collection<String>> strings = Holders.collectionr(new ArrayList<String>(),String.class);

        public static Config getInstance() {
            if (config == null) {
                config = ConfigContext.getConfigInstance(Config.class);
            }
            return config;
        }
    public String getNameSpace() {
        return "config";
    }

        public void setStrings(List strings){
        this.strings.set(strings);
        }

        public List<String> getStrings(){
        return (List<String>)this.strings.get();
        }
    public void setB(String b) {
        this.b.set(b);
    }

    public String getB() {
        return this.b.get();
    }

    public void setTests(List<Test> tests) {
        this.tests.set(tests);
    }

    public List<Test> getTests() {
        return (List<Test>) this.tests.get();
    }

    public void removeTest(Test test) {
        this.tests.remove(test);
    }

    public void setExpMap(Map map) {
        this.expMap.set(map);
    }

    public Map getExpMap() {
        return this.expMap.get();
    }

    public void removeMapValue(String key) {
        this.expMap.remove(key);
    }
}

class Test extends UniqueKey {
    private Conf<Integer> a = Holders.simple(0);

    public Test() {
    }

    public Test(int a) {
        this.setA(a);
```

```
    }

    public void setA(int a) {
        this.a.set(a);
    }

    public int getA() {
        return this.a.get();
    }
}
```

```
class StoreDemo{
    public static void main(String[] args){
        Config config = Config.getInstance();
        config.setB("123");
        Map<String,Test> testMap = new HashMap<String, Test>();
        testMap.put("11",new Test(11));
        testMap.put("22",new Test(22));
        config.setExpMap(testMap);
        List<Test> tests = new ArrayList<Test>();
        tests.add(new Test(33));
        tests.add(new Test(33));
        config.setTests(tests);
                List<String> strings = new ArrayList<String>();
                strings.add("a");
                strings.add("b");
                config.setStrings(strings);
    }
}
```

Id	value
config.b	123
config.expMap.11.a	11
config.expMap.22.a	22
config.tests.uBD1difs.a	33
config.tests.1y3t77tf.a	33
config.strings	a,,b,,

MapCollectionConfig

```
 public void removeMapValue(String key) {
        this.expMap.remove(key);
    }
```

Holder config.getExpMap.remove(key),MapHolder config.getExpMap().get(key)

```
public static Config getInstance() {
 if (config == null) {
 config = ConfigContext.getConfigInstance(Config.class);
 }
 return config;
}
```

setMap removeholderholderdao

ConfConf

RPC