

Realm Java 闪电介绍

Chen Mulong

Developer From Realm

email: mc@realm.io

github: [@beeender](#)

Realm是什么

- 开源的数据库框架
- 励志于替代SQLite和Core Data
- 自主实现的数据存储引擎
- 快速，简单，线程安全
- 跨平台（Android、iOS、Mac.....）
- 完全免费

谁在使用Realm?

amazon

hipmunk

Pinterest

ebay

3M

GROUPON

Budweiser

AVIS

SAP

HYATT

BBC

BANDAI

intel

SIEMENS

RITE AID

intuit

L'ORÉAL

HEARST

ROBEAT

HEARST



豆瓣阅读
文艺青年爱读书。



聚美优品-极速免税店
中国最大跨境美妆电商，
限时特卖商城。



BearyChat
是时候换一种全新的工作
方式了。



阿里卖家
众多外贸企业首选且正在
使用的手机工作平台。

如何使用？

定义模型

```
public class Dog extends RealmObject {  
    @Required // name 不可为空  
    private String name;  
    @Index // 为 age 建立索引  
    private long age;  
    private Date birthday;  
  
    public String getName() { return name; }  
    public void setName(String name) { this.name = name; }  
    public long getAge() { return age; }  
    public void setAge(long age) { this.age = age; }  
    public Date getBirthday() { return birthday; }  
    public void setBirthday(Date birthday) { this.birthday = birthday; }  
}
```

Realm支持的数据类型

- byte, short, int, long, Byte, Short, Integer, Long
- float, double, Float, Double
- boolean, Boolean
- String
- Date
- RealmObject, RealmList<? extends RealmObject>

定义关系

```
public class Owner extends RealmObject {  
    private String name;  
    private RealmList<Dog> dogs; // 一对多  
    private Cat cat; // 一对一  
  
    public Cat getCat() { return cat; }  
    public void setCat(Cat cat) { this.cat = cat; }  
    public RealmList<Dog> getDogs() { return dogs; }  
    public void setDogs(RealmList<Dog> dogs) { this.dogs =  
dogs; }  
    public String getName() { return name; }  
    public void setName(String name) { this.name = name; }  
}
```

写入

```
Realm realm = Realm.getInstance(context);
```

```
realm.beginTransaction();
```

```
Dog dog = realm.createObject(Dog.class);  
dog.setName("旺财"); dog.setAge(73);
```

```
realm.commitTransaction();
```


查询

```
Dog dog = realm.where(Dog.class)
    .equalTo("name", "旺财")
    .findFirst();
```

```
RealmResults<Dog> results =
realm.where(Dog.class)
    .greaterThan("age", 10)
    .findAll();
```

关联查询

```
Owner owner = realm.where(Owner.class)
    .equalTo("cat.name", "咪咪")
    .findFirst();
```

```
RealmResults<Owner> results =
    realm.where(Owner.class)
        .greaterThan("dogs.age", 10)
        .findAll();
```

通知

```
private Dog dog; // 需要从Realm取出进行初始化
```

```
...
```

```
private RealmChangeListener listener = new  
RealmChangeListener() {  
    @Override  
    public void onChange() {  
        textView.setText(dog.getName());  
    }  
};
```

```
...
```

```
realm.addChangeListener(listener);
```

Realm与线程

- Realm、RealmObject和RealmResults实例都不可跨线程使用
- 请在每个线程获取单独的Realm实例来进行操作
- 某一线程写入操作的结果可以自动反馈到其他线程的Realm中
- 支持异步查询与异步写入

更多

- 异步查询与异步写入
- 加密
- RealmBaseAdapter
- 从JSON导入
- 数据迁移 Migration
-

<http://realm.io>



@realm



/realm



#realm



@RealmChina

Thanks for watching!



mc@realm.io

Questions?