Loading & Storing Data

Grant Little

www.grantlittle.me

Overview

Read Through/Write Through Caching

- CacheWriters
- CacheLoaders

Expiry Policies

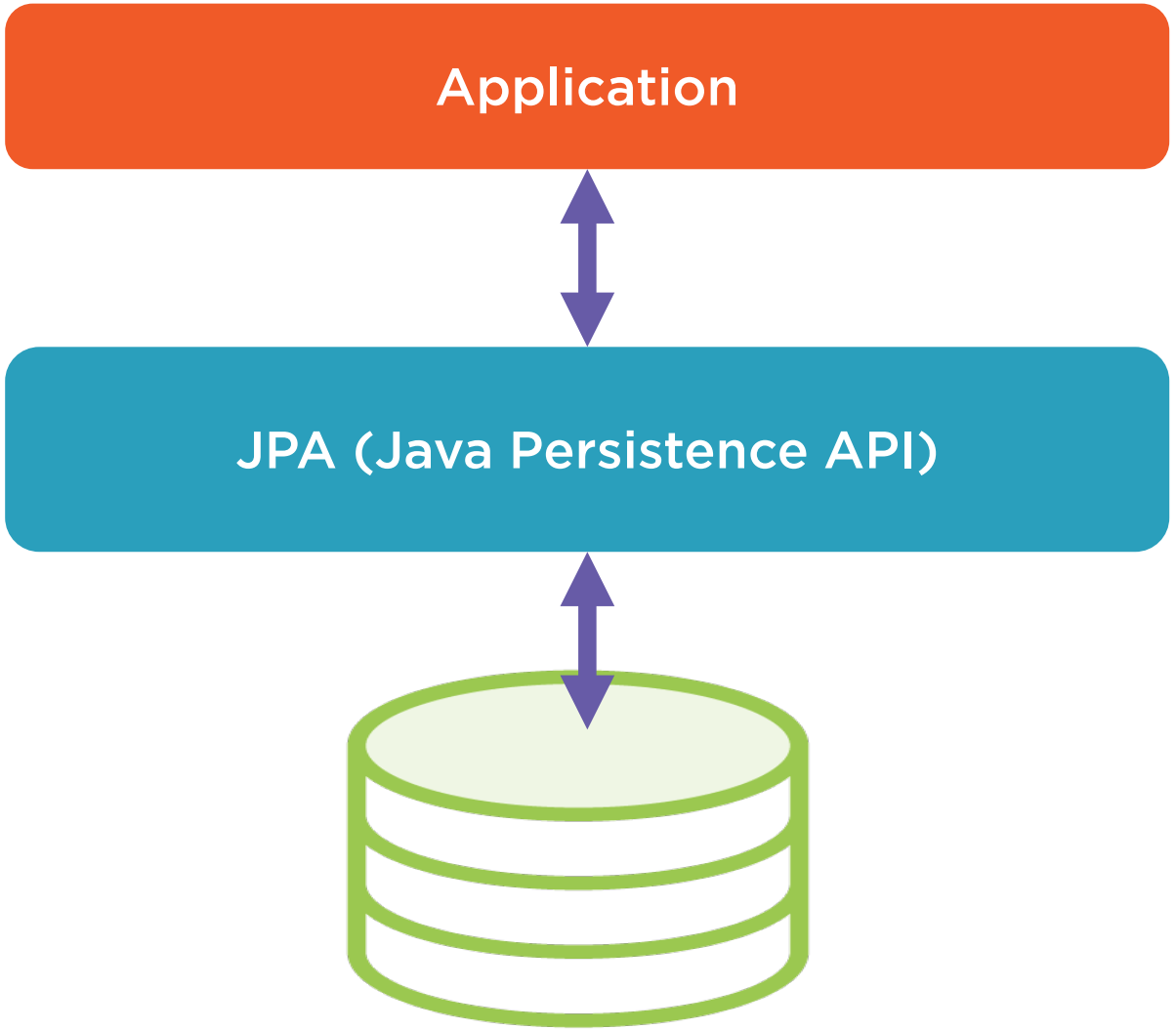
Using an External Data Store

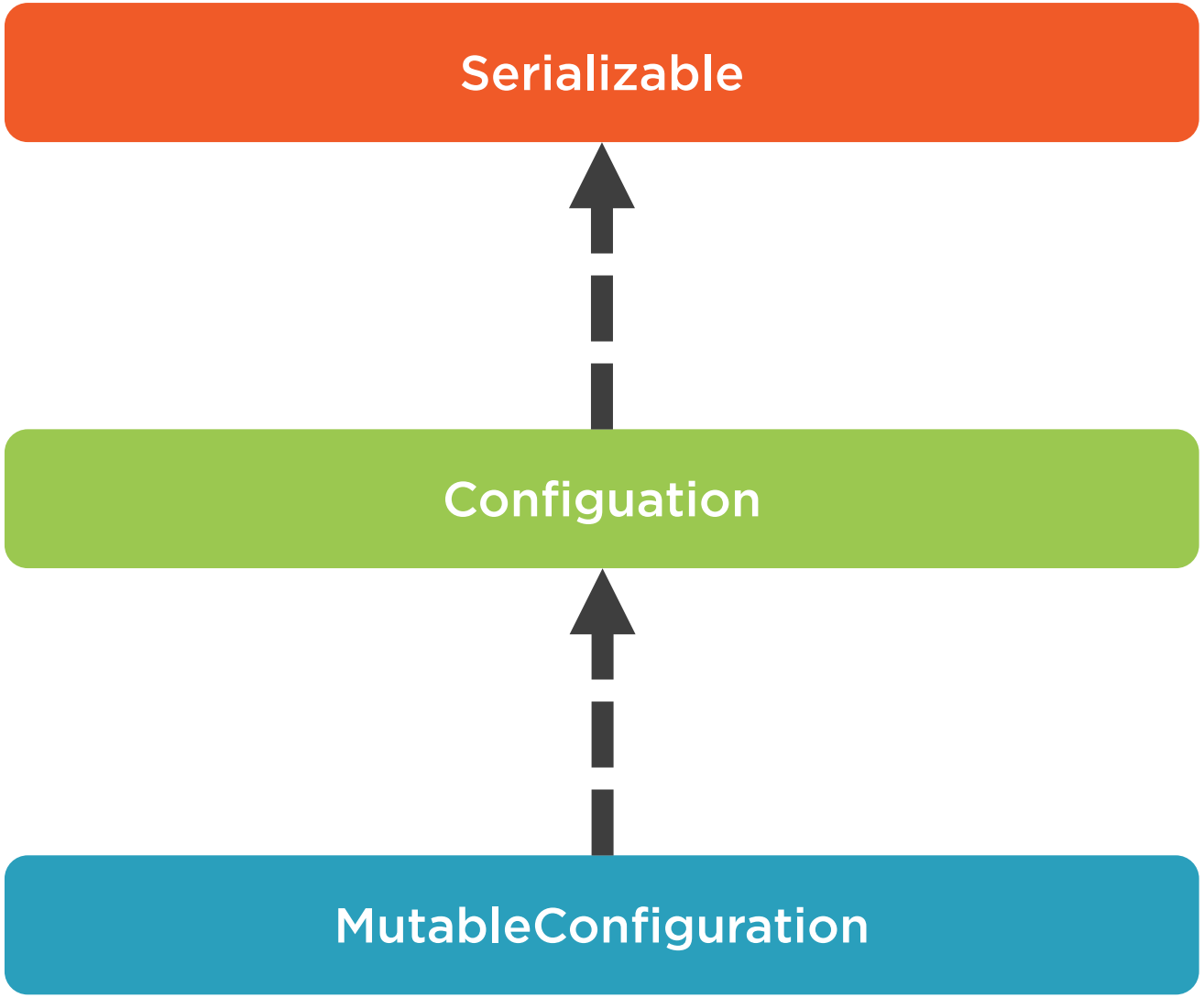
Data Resilience

Distributed Data

Permanent Persistence

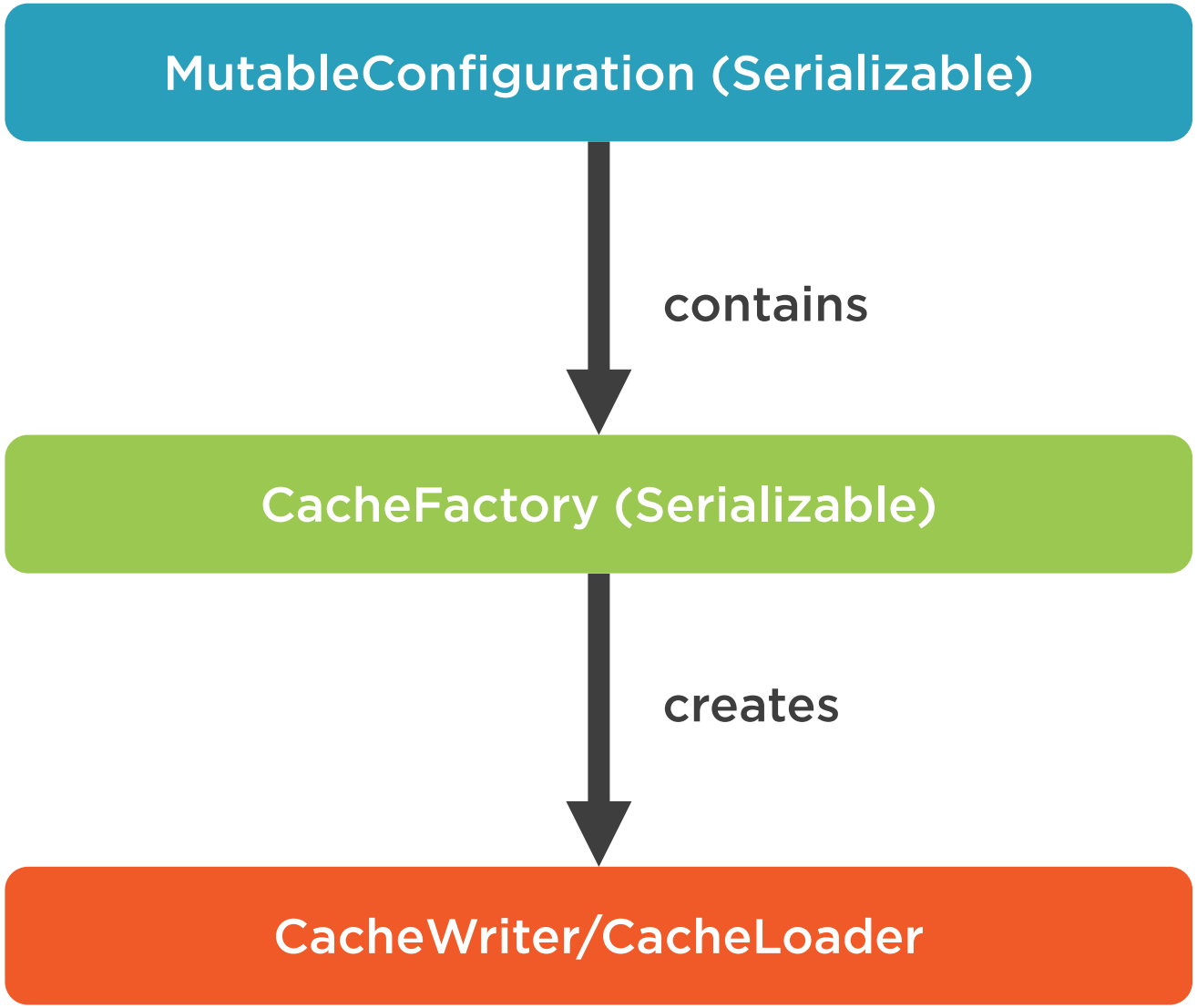
Data Resilience




Serializable
Configuation
MutableConfiguration

CacheWriter/Loader difficult to Serialize

- JDBC connections
- Connection pools
- Other non non serializable references


MutableConfiguration (Serializable)
contains
CacheFactory (Serializable)
creates
CacheWriter/CacheLoader

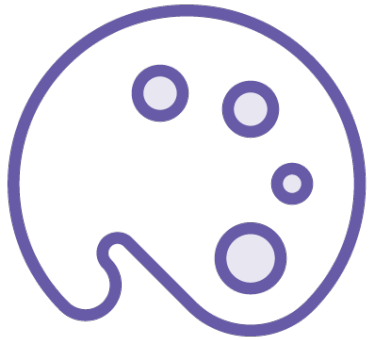
Expiry Policies

Defines when an entry may be removed from the cache based on some conditions

CacheWriters are not invoked when entries is expired

Performing a get request on the cache will cause the CacheLoader to retrieve the data

Available Expiry Policies

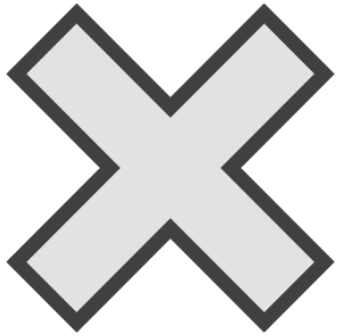


Created

Modified

Accessed

Touched



Eternal

```
mutableConfiguration.setExpiryPolicyFactory( . . . );
```

MutableConfiguration

```
public class MyFactory implements Factory<Object> {
    public Object create() {
        return new Object();
    }
}
```

Factories

Custom

```
FactoryBuilder.factoryOf(MyObject.class);

FactoryBuilder.factoryOf(“com.pluralsight.MyObject”);

FactoryBuilder.factoryOf(new MyObject());
```

FactoryBuilder

```
TouchedExpiryPolicy.factoryOf(
        new Duration(TimeUnit.SECONDS, 20)
)
```

Object Specific Factories

Expiry Policies

Eviction

The process of removing entries from a Cache when the Cache has exceeded a resource limit

Review

Write Through Caching

- Cache Writers

Read Through Caching

- Cache Loaders

Factories

Expiry Policies