

## Ehcache 设值、取值、删除

```
package com.haohua.util;

import java.io.Serializable;

import net.sf.ehcache.Cache;
import net.sf.ehcache.CacheManager;
import net.sf.ehcache.Element;

public class CacheUtil {
    private static CacheManager cacheManager = new CacheManager();

    /**
     * 根据名称获取对应的cache对象
     * @param name
     * @return
     */
    public static Cache getCache(String name) {
        return cacheManager.getCache(name);
    }

    /**
     * 存值
     * @param cache 缓存对象
     * @param key 键值
     * @param value 值
     */
    public static void set(Cache cache, Object key, Object value) {
        Element element = new Element(key, value);
        cache.put(element);
    }

    public static void set(Cache cache, Serializable key, Serializable value) {
        Element element = new Element(key, value);
        cache.put(element);
    }

    public static void set(String name, Object key, Object value) {
        Cache cache = getCache(name);
        Element element = new Element(key, value);
        cache.put(element);
    }

    public static void set(String name, Serializable key, Serializable value) {
        Cache cache = getCache(name);
        Element element = new Element(key, value);
        cache.put(element);
    }

    /**
     * 取值
     */
}
```

```

    * @param cache
    * @param key
    * @return
    */
    public static Object get(Cache cache, Object key) {
        Element element = cache.get(key);
        return element == null ? null : element.getObjectValue();
    }

    public static Object get(Cache cache, Serializable key) {
        Element element = cache.get(key);
        return element == null ? null : element.getObjectValue();
    }

    public static Object get(String name, Serializable key) {
        Cache cache = getCache(name);
        Element element = cache.get(key);
        return element == null ? null : element.getObjectValue();
    }

    public static Object get(String name, Object key) {
        Cache cache = getCache(name);
        Element element = cache.get(key);
        return element == null ? null : element.getObjectValue();
    }

    /**
     * 清除缓存中的所有数据
     * @param cache
     */
    public static void removeAll(Cache cache) {
        cache.removeAll();
    }

    public static void removeAll(String name) {
        Cache cache = getCache(name);
        cache.removeAll();
    }

    /**
     * 根据key删除数据
     * @param cache
     * @param key
     */
    public static void remove(Cache cache, Object key) {
        cache.remove(key);
    }

    public static void remove(String name, Serializable key) {
        Cache cache = getCache(name);
        cache.remove(key);
    }
}

```

