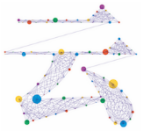

eMem

弹性云缓存系统（The cloud cache system based on redis/memcached）

2014 全国高校云计算应用创新大赛

客户端使用手册

eMem——弹性云缓存系统

上海交通大学 eMem 团队

2014-12

客户端使用手册

一、 客户端使用非常简单，只需将“客户端 JAR 包”的相应 jar 包打包给用户即可投入使用。

二、 注意在使用之前，需要通过 eMem 运维人员联系获取用户权限 token 值

三、 Jar 包的 API 文档请参考“客户端 JAR 包”目录下的“JavaDoc”目录。

用户可以使用 JAVA 语言来使用缓存服务. 需要引入下面的两个包:

http/rmi-cache-client.jar

groovy-all.x.x.x.jar (groovy 依赖包)

使用时, 首先使用 emem.cacheclient.CacheClient 的构造方法创建一个实例, 如下:

```
import emem.cacheclient.CacheClient;
HTTPCacheClient cacheClient = new HTTPCacheClient(host, port, token);
```

然后使用的是与 HTTP 协议等效的方法, 例如 set 操作:

```
cacheClient.set(key, value)
cacheClient.set(key, value, expire)
```

其他的操作类似, 就不一一列出了.

另外, 为了便利 JAVA 接口的使用, 可以直接将一个完整的对象存储进缓存, 以后可以从缓存提取这个对象, 例如:

```
Person person = new Person(name, age)
cacheClient.set(key, person)
person = (Person)cacheClient.get(key)
```

下面是客户端 JAR 包的 JAVA DOC 文档摘要:

限定符和类型	方法和说明
java.lang.String	**get**(java.lang.String key) 返回绑定到制定键的字符串类型的值

java.lang.String	**get**(java.lang.String key, int expire)
java.io.Serializable	**getObject**(java.lang.String key) 返回对象类型的数据，该对象需要实现 Serializable 接口.
java.io.Serializable	**getObject**(java.lang.String key, int expire)
java.lang.String	**hashGet**(java.lang.String key, java.lang.String index) 返回 hash 类型的数据，仅仅对其中的某个索引进行操作.
java.lang.String	**hashGet**(java.lang.String key, java.lang.String index, int expire)
java.util.Map<java.lang.String, java.lang.String>	**hashGetAll**(java.lang.String key) 返回 hash 类型的完整数据
java.util.Map<java.lang.String, java.lang.String>	**hashGetAll**(java.lang.String key, int expire)
void	**hashRemove**(java.lang.String key, java.lang.String index) 可以针对 hash 数据中的某个索引进行删除
void	**hashRemove**(java.lang.String key, java.lang.String index, int expire)
void	**hashSet**(java.lang.String key, java.lang.String index, java.lang.String value) 设置 hash 类型的数据，仅仅对其中的某个索引进行操作.
void	**hashSet**(java.lang.String key, java.lang.String index,

	java.lang.String value, int expire)
void	**hashSetAll**(java.lang.String key, java.util.Map<java.lang.String,java.lang.String> map) 设置hash类型的完整数据，其中hash值用一个字符串到字符串的映射 map 表示
void	**hashSetAll**(java.lang.String key, java.util.Map<java.lang.String,java.lang.String> map, int expire)
long	**hashSize**(java.lang.String key) 返回的是 hash 数据中索引的个数
long	**hashSize**(java.lang.String key, int expire)
void	**set**(java.lang.String key, java.lang.String value) 存储单纯的字符串类型的键值对
void	**set**(java.lang.String key, java.lang.String value, int expire)
void	**setObject**(java.lang.String key, java.io.Serializable obj) 存储对象类型的数据，该对象需要实现 Serializable 接口.
void	**setObject**(java.lang.String key, java.io.Serializable obj, int expire)