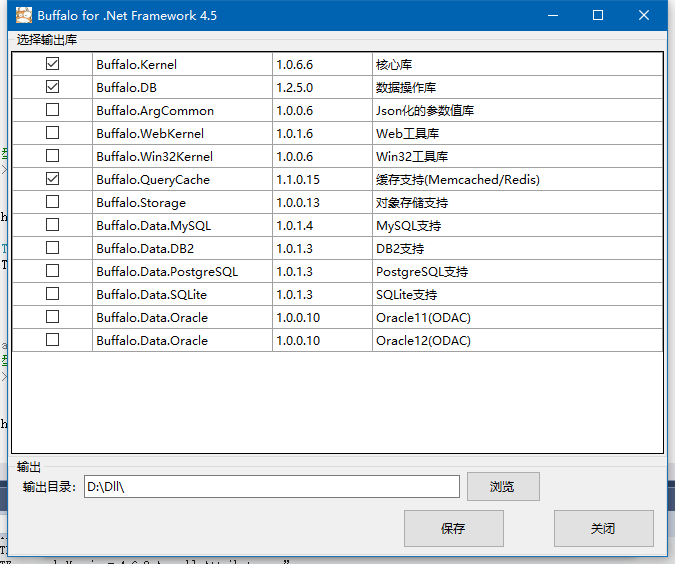
# Buffalo缓存/分布式缓存类使用

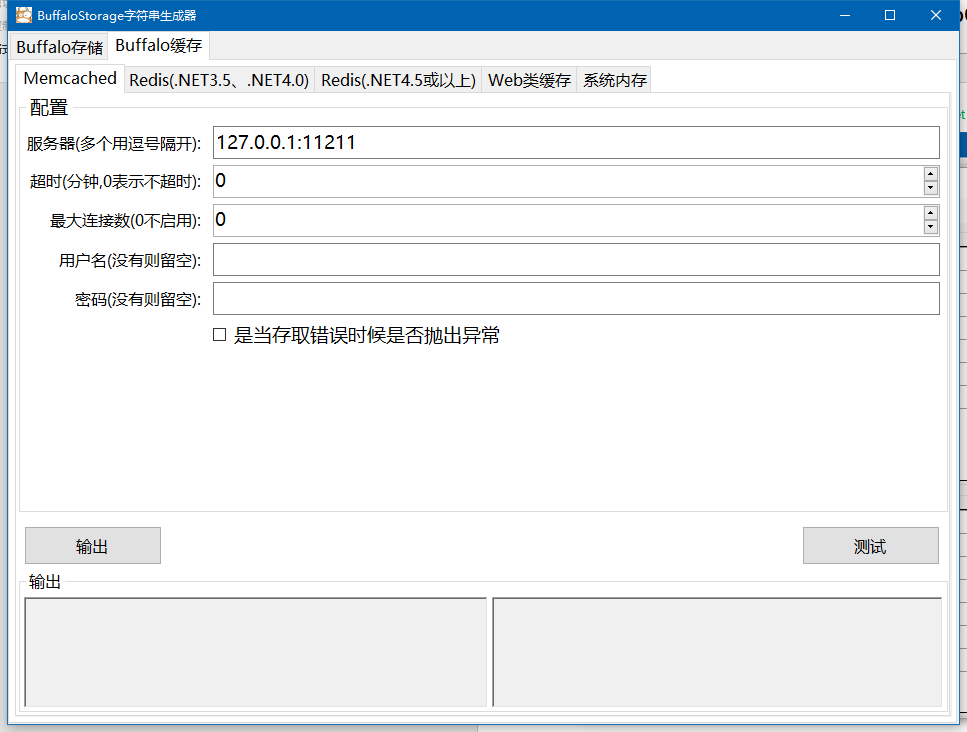
## 介绍

Buffalo.QueryCache是一个支持多种缓存调用的缓存封装包，通过此包可以方便使用和切换多种缓存

项目依赖文件输出可以在 Buffalo安装工具🡪选择对应的.net版本🡪输出 获取



连接字符串生成器可以在Buffalo安装工具🡪帮助🡪连接字符串🡪选择Buffalo缓存选项卡 然后选择对应的缓存选项卡，填入相关信息就可以输出对应的连接字符串



## 缓存类型支持

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 参数 | 连接字符串 | 说明 |
| **Memcached** | memcached | server=127.0.0.1:11211;expir=30;poolsize=30;throw=0;uid=abc;pwd=123456 | *expir=30(可选) 是表示30分钟过期，没这个项则表示不过期(可选)*  *poolsize是最大连接数，(可选)*  *throw=0 是当存取错误时候是否抛出异常(可选)*  *uid=abc 登录账户(可选)*  *pwd=123456 登录密码(可选)*  *--分布式缓存* |
| **Redis** | redis | server=127.0.0.1:6379;readserver=127.0.0.1:6381;expir=30;poolsize=30;throw=0;uid=abc;pwd=123456 | *server 是连接服务器*  *readserver 是只读服务器*  *expir=30(可选) 是表示30分钟过期，没这个项则表示不过期(可选)*  *poolsize是最大连接数(可选)*  *throw=0 是当存取错误时候是否抛出异常(可选)*  *uid=abc 登录账户(可选)*  *pwd=123456 登录密码(可选)*  *--分布式缓存* |
| **ASP.Net Cache** | web | expir=30 | expir=30 是表示30分钟过期，没这个项则默认过期方式*(可选)*  使用HttpContext.Current.Cache实现，Winform可用  *--单机缓存* |
| **系统内存** | system |  | 使用静态变量实现  *--单机缓存* |

## 使用

示例

QueryCache cache = CacheUnit.CreateCache(“memcached”, "server=127.0.0.1:11211,127.0.0.1:11212;expir=30;poolsize=30;throw=0");

实例化缓存后可以使用其功能

QueryCache方法

|  |  |  |
| --- | --- | --- |
| 方法 | 返回类型 | 说明 |
| SetValue(string key, int expirSeconds, object value) | void | 写入值  string key //键  int expirSeconds //过期时间(可选)  object value //值 |
| SetValue<E>(string key,  int expirSeconds,E value) | void | 写入值  E //类型  string key //键  int expirSeconds //过期时间(可选)  E value //值 |
| GetValue(string key) | object | 获取值  string key //键 |
| GetValue<E>(string key,E defaultValue) | E  如果不存在此键则返回defaultValue | 获取值  string key //键  E defaultValue //默认值 |
| GetValues(string[] keys) | IDictionary<string, object>  返回键值对应列表 | 批量获取值  String[] keys //要获取的键 |
| DoIncrement(string key,uint inc) | long  返回自增后的值 | 自增  string key //键  uint inc //值(可选，不填则自增1) |
| DoDecrement(string key, uint dec) | long  返回自减后的值 | 自减  string key //键  uint dec //值(可选，不填则自减1) |
| DeleteValue(string key) | void | 删除键值  string key //键 |
| Clear() | void | 清除所有数据 |