Redis-Based Session Store System (Using Redis CLI)

1. Introduction

Implementing a login session store system using Redis CLI. Redis is an in-memory key-value store widely used for session management due to its speed and support for key expiration (TTL).

2. Session Key Design

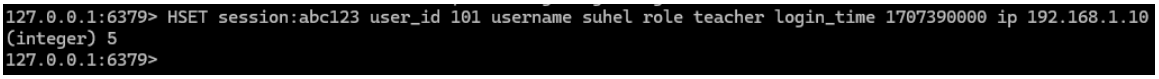
Each session is stored using the following key pattern:
session:<session_id>

Session data includes:
- user_id
- username
- role
- login_time
- IP address

Redis HASH is used to store session attributes.

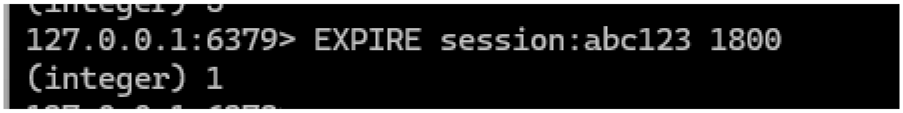
3. Creating a Login Session

Step 1: Define Session ID
// session:abc123 (Usually generated using hashing)
Step 2: Store Session Data
HSET session:abc123 user_id 101 username suhel role student login_time 1707390000 ip 192.168.1.10



```
127.0.0.1:6379> HSET session:abc123 user_id 101 username suhel role teacher login_time 1707390000 ip 192.168.1.10
(integer) 5
127.0.0.1:6379>
```

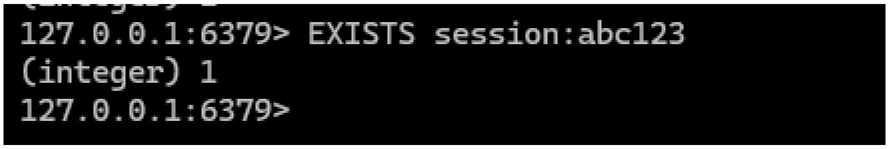
Step 3: Set Session Expiration (30 minutes)
EXPIRE session:abc123 1800



```
127.0.0.1:6379> EXPIRE session:abc123 1800
(integer) 1
```

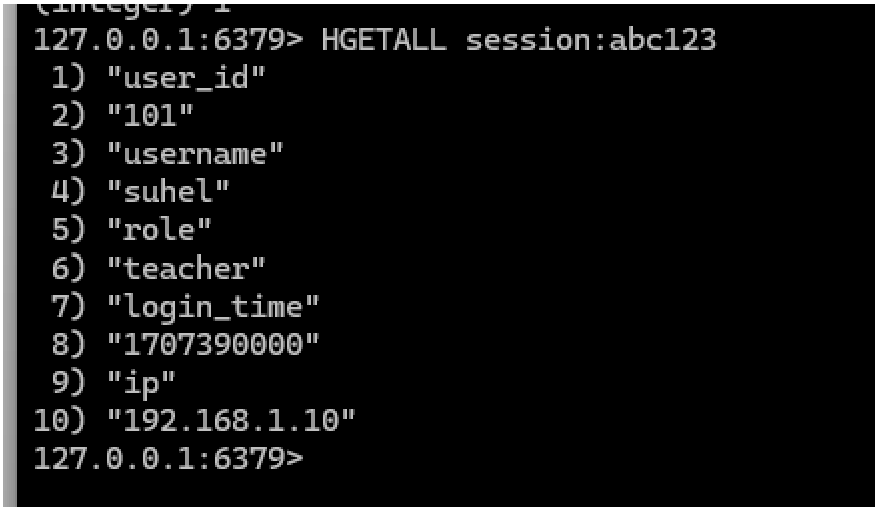
4. Session Validation

Check if session exists:
EXISTS session:abc123



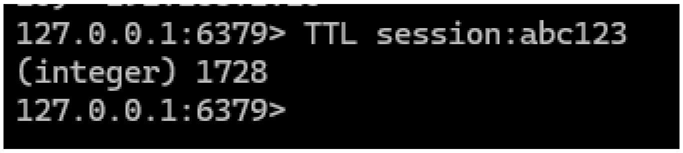
```
127.0.0.1:6379> EXISTS session:abc123
(integer) 1
127.0.0.1:6379>
```

Fetch session data:
HGETALL session:abc123



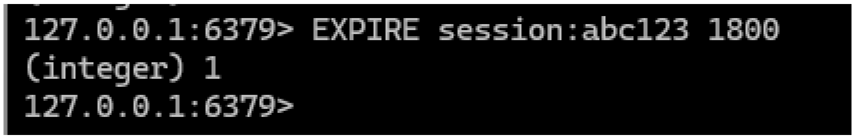
```
127.0.0.1:6379> HGETALL session:abc123
 1) "user_id"
 2) "101"
 3) "username"
 4) "suhel"
 5) "role"
 6) "teacher"
 7) "login_time"
 8) "1707390000"
 9) "ip"
10) "192.168.1.10"
127.0.0.1:6379>
```

Check remaining TTL:
TTL session:abc123



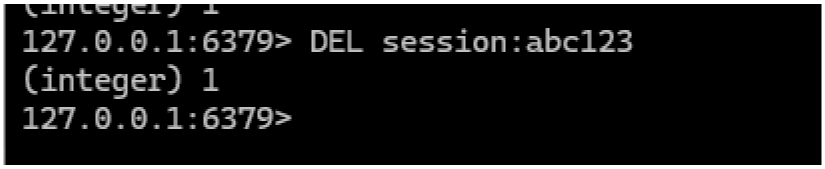
```
127.0.0.1:6379> TTL session:abc123
(integer) 1728
127.0.0.1:6379>
```

5. Sliding Session (Extend on Activity)
EXPIRE session:abc123 1800



```
127.0.0.1:6379> EXPIRE session:abc123 1800
(integer) 1
127.0.0.1:6379>
```

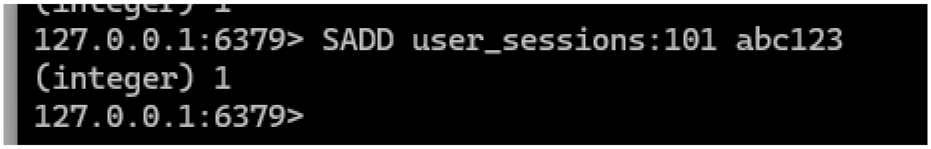
6. Logout (Session Destruction)
DEL session:abc123



```
127.0.0.1:6379> DEL session:abc123
(integer) 1
127.0.0.1:6379>
```

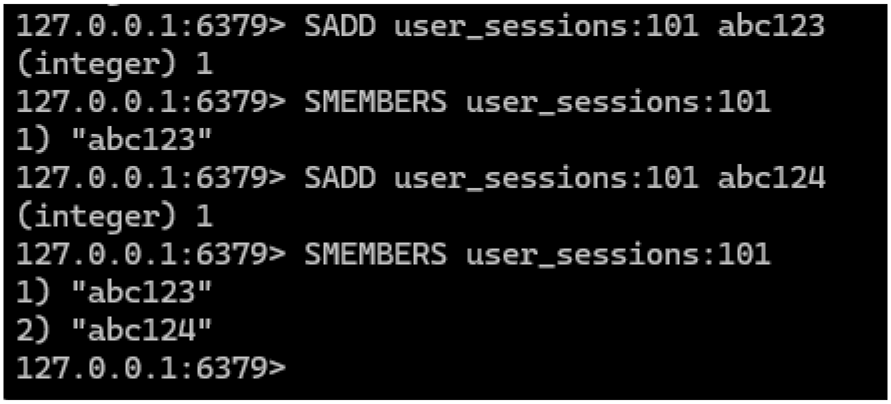
7. Multiple Sessions per User (Optional)

Add session to user session list:
SADD user_sessions:101 abc123



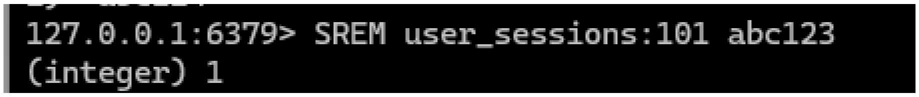
```
127.0.0.1:6379> SADD user_sessions:101 abc123
(integer) 1
127.0.0.1:6379>
```

View all sessions:
SMEMBERS user_sessions:101



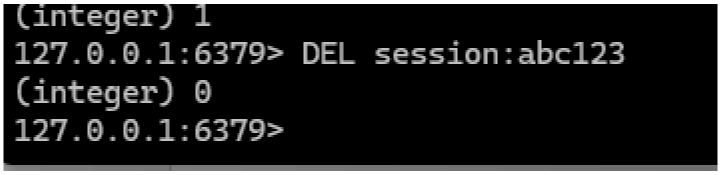
```
127.0.0.1:6379> SADD user_sessions:101 abc123
(integer) 1
127.0.0.1:6379> SMEMBERS user_sessions:101
1) "abc123"
127.0.0.1:6379> SADD user_sessions:101 abc124
(integer) 1
127.0.0.1:6379> SMEMBERS user_sessions:101
1) "abc123"
2) "abc124"
127.0.0.1:6379>
```

Remove session:
SREM user_sessions:101 abc123



```
127.0.0.1:6379> SREM user_sessions:101 abc123
(integer) 1
```

DEL session:abc123



```
(integer) 1
127.0.0.1:6379> DEL session:abc123
(integer) 0
127.0.0.1:6379>
```

8. Note: Automatic Cleanup
Redis automatically removes expired session keys without manual intervention.

Advantages of Redis for Session Storage
- Fast in-memory access
- Built-in expiration
- Atomic operations
- Easy CLI-based testing