

Opracowano na podstawie materiałów znajdujących się na stronach: <https://hazelcast.org> oraz <https://hazelcast.com>

1 Przykładowa aplikacja

1.1 Wstęp

Aplikacja wykorzystująca Hazelcast wymaga do działania dodania do ścieżki CLASSPATH odpowiedniego archiwum jar, które może zostać pobrane ze strony <https://hazelcast.org/download/>. Do celów laboratoryjnych wystarczające jest archiwum `hazelcast-all-NUMER-WERSJI.jar`. Na podanej powyżej stronie można znaleźć również sposób konfiguracji projektu m.in. dla narzędzi Maven i Gradle. Hazelcast można łączyć Hazelcast z innymi technologiami. Ich lista jest podstępna pod adresem: <https://hazelcast.org/hub/>. Dokumentacja do Hazelcast znajduje się pod adresem <https://hazelcast.org/documentation/>.

Hazelcast można używać w dwóch trybach:

- *member* - Spowoduje to dodanie nowej instancji do klastra. Na jednym węźle może działać kilka instancji. Stworzenie nowej instancji następuje poprzez: `Hazelcast.newHazelcastInstance()`. Poza utworzeniem nowej instancji, w tym trybie możliwe jest również wykonywanie innych operacji, takich jak dodawanie obiektów.
- *client* - Nowy klient dla Javy tworzony jest poprzez wywołanie: `HazelcastClient.newHazelcastClient()`.

1.2 Memeber HListener

Aplikacja działające w trybie *member* i dodająca następujące listener'y: *DistributedObjectListener*, *MembershipListener*, *MigrationListener*, *EntryAddedListener*.

```
1 import java.net.UnknownHostException;
2
3 import com.hazelcast.config.Config;
4 import com.hazelcast.core.DistributedObjectEvent;
5 import com.hazelcast.core.DistributedObjectListener;
6 import com.hazelcast.core.EntryEvent;
7 import com.hazelcast.core.Hazelcast;
8 import com.hazelcast.core.HazelcastInstance;
9 import com.hazelcast.map.IMap;
10 import com.hazelcast.cluster.MembershipEvent;
11 import com.hazelcast.cluster.MembershipListener;
12 import com.hazelcast.partition.MigrationListener;
13 import com.hazelcast.partition.MigrationState;
14 import com.hazelcast.partition.PartitionService;
15 import com.hazelcast.partition.ReplicaMigrationEvent;
16 import com.hazelcast.map.listener.EntryAddedListener;
17
18 public class HListener {
19     public static void main(String[] args) throws UnknownHostException {
20         Config config = HConfig.getConfig();
21
22         HazelcastInstance instance = Hazelcast.newHazelcastInstance(config);
23
24         instance.addDistributedObjectListener(new DistributedObjectListener() {
25
26             @Override
27             public void distributedObjectDestroyed(DistributedObjectEvent e) {
28                 System.out.println(e);
29             }
30
31             @Override
```

```

32         public void distributedObjectCreated(DistributedObjectEvent e) {
33             System.out.println(e);
34         }
35     });
36
37     instance.getCluster().addMembershipListener(new MembershipListener() {
38
39         @Override
40         public void memberRemoved(MembershipEvent e) {
41             System.out.println(e);
42         }
43
44         @Override
45         public void memberAdded(MembershipEvent e) {
46             System.out.println(e);
47         }
48     });
49
50     PartitionService partitionService = instance.getPartitionService();
51     partitionService.addMigrationListener(new MigrationListener() {
52
53         @Override
54         public void replicaMigrationFailed(ReplicaMigrationEvent e) {
55             System.out.println(e);
56         }
57
58         @Override
59         public void replicaMigrationCompleted(ReplicaMigrationEvent e) {
60             System.out.println(e);
61         }
62
63         @Override
64         public void migrationStarted(MigrationState s) {
65             System.out.println(s);
66         }
67
68         @Override
69         public void migrationFinished(MigrationState s) {
70             System.out.println(s);
71         }
72     });
73
74     IMap<Long, Student> students = instance.getMap("students");
75
76     students.addEntryListener(new EntryAddedListener<Long, Student>() {
77
78         @Override
79         public void entryAdded(EntryEvent<Long, Student> e) {
80             System.out.println(e);
81         }
82     }, true);
83
84 }
85
86 }

```

Przykład 1: `src/LabHazelcast/src/main/java/HListener.java` {link}

Kompilacja:

`mvn clean compile`

Uruchomienie:

```
mvn exec:java -Dexec.mainClass="HListener"
```

1.3 Memeber HMapPut

Aplikacja działająca w trybie *memeber* i dodająca do mapy *students* dwa obiekty należące do klasy *Student*. Klucze wyznaczone są na podstawie identyfikatora procesu *pid* oraz liczby pobranej generatora liczb pseudolosowych. Użycie *pid* powinno spowodować, że przy uruchomieniu na jednym komputerze aplikacji kilka razy klucze będą miały inne wartości.

```
1 import java.io.Serializable;
2
3 public class Student implements Serializable {
4
5     private static final long serialVersionUID = 1L;
6
7     private String name;
8     private int birthyear;
9
10    public Student(String name, int birthyear) {
11        this.name = name;
12        this.birthyear = birthyear;
13    }
14
15    public String getName() {
16        return name;
17    }
18
19    public void setName(String name) {
20        this.name = name;
21    }
22
23    public int getBirthyear() {
24        return birthyear;
25    }
26
27    public void setBirthyear(int birthyear) {
28        this.birthyear = birthyear;
29    }
30
31    @Override
32    public String toString(){
33        return "Student " + name + " born in " + birthyear;
34    }
35 }
```

Przykład 2: `src/LabHazelcast/src/main/java/Student.java` {link}

```
1 import java.net.UnknownHostException;
2 import java.util.Map;
3 import java.util.Random;
4
5 import com.hazelcast.config.Config;
6 import com.hazelcast.core.Hazelcast;
7 import com.hazelcast.core.HazelcastInstance;
8
9 public class HMapPut {
10
11     final private static Random r = new Random(System.currentTimeMillis());
12
13     public static void main(String[] args) throws UnknownHostException {
14         Config config = HConfig.getConfig();
```

```

15     HazelcastInstance instance = Hazelcast.newHazelcastInstance(config);
16     Map<Long, Student> students = instance.getMap("students");
17     Long key1 = (long) Math.abs(r.nextInt());
18     Student student1 = new Student("Kowalski", 1999);
19     System.out.println("PUT " + key1 + " => " + student1);
20     students.put(key1, student1);
21     Long key2 = (long) Math.abs(r.nextInt());
22     Student student2 = new Student("Nowak", 2002);
23     students.put(key2, student2);
24     System.out.println("PUT " + key2 + " => " + student2);
25 }
26 }

```

Przykład 3: `src/LabHazelcast/src/main/java/HMapPut.java` {link}

Kompilacja:

```
mvn clean compile
```

Uruchomienie:

```
mvn exec:java -Dexec.mainClass="HMapPut"
```

1.4 Client HMapGet

Aplikacja działająca w trybie *client* i wyświetlająca zawartość mapy *students*.

```

1  import com.hazelcast.client.config.ClientConfig;
2
3  import java.net.UnknownHostException;
4  import java.util.Map.Entry;
5
6  import com.hazelcast.client.HazelcastClient;
7  import com.hazelcast.core.HazelcastInstance;
8  import com.hazelcast.map.IMap;
9
10 public class HMapGet {
11     public static void main( String[] args ) throws UnknownHostException {
12         ClientConfig clientConfig = HConfig.getClientConfig();
13         HazelcastInstance client = HazelcastClient.newHazelcastClient( clientConfig );
14         IMap<Long, Student> students = client.getMap( "students" );
15         System.out.println("All student: ");
16         for(Entry<Long, Student> e : students.entrySet()){
17             System.out.println(e.getKey() + " => " + e.getValue());
18         }
19     }
20 }

```

Przykład 4: `src/LabHazelcast/src/main/java/HMapGet.java` {link}

Kompilacja:

```
mvn clean compile
```

Uruchomienie:

```
mvn exec:java -Dexec.mainClass="HMapGet"
```

1.5 Client HExecutorService

Aplikacja działająca w trybie *client* i wyświetlająca zawartość lokalnych kubeków przy pomocy *IExecutorService*.

```

1  import java.io.Serializable;
2  import java.net.UnknownHostException;
3  import java.util.Set;
4  import java.util.concurrent.Callable;
5
6  import com.hazelcast.client.HazelcastClient;
7  import com.hazelcast.client.config.ClientConfig;
8  import com.hazelcast.core.HazelcastInstance;
9  import com.hazelcast.core.HazelcastInstanceAware;
10 import com.hazelcast.core.IExecutorService;
11 import com.hazelcast.map.IMap;
12
13 public class HExecutorService {
14
15     public static void main( String[] args ) throws UnknownHostException {
16         ClientConfig clientConfig = HConfig.getClientConfig();
17         final HazelcastInstance client = HazelcastClient.newHazelcastClient( clientConfig );
18
19         IExecutorService executorService = client.getExecutorService("exec");
20         executorService.submitToAllMembers(new HCallable());
21     }
22 }
23
24 class HCallable implements Callable<Integer>, Serializable, HazelcastInstanceAware {
25     private static final long serialVersionUID = 1L;
26     private transient HazelcastInstance instance;
27
28     @Override
29     public Integer call() {
30         IMap<Long, Student> students = instance.getMap("students");
31         Set<Long> keys = students.localKeySet();
32         for (Long k : keys) {
33             System.out.println("Instance " + instance + " " + k + " => " + students.get(k));
34         }
35         return keys.size();
36     }
37
38     @Override
39     public void setHazelcastInstance(HazelcastInstance instance) {
40         this.instance = instance;
41     }
42 }

```

Przykład 5: src/LabHazelcast/src/main/java/HExecutorService.java {link}

Kompilacja:

```
mvn clean compile
```

Uruchomienie:

```
mvn exec:java -Dexec.mainClass="HExecutorService"
```

1.6 Client HEntryProcessor

Aplikacja działająca w trybie *client* i zmieniająca wielkość liter w nazwiskach studentów przy pomocy *EntryProcessor*.

```

1  import java.io.Serializable;
2  import java.net.UnknownHostException;
3  import java.util.Map.Entry;
4

```

```

5 import com.hazelcast.client.HazelcastClient;
6 import com.hazelcast.client.config.ClientConfig;
7 import com.hazelcast.core.HazelcastInstance;
8 import com.hazelcast.map.IMap;
9 import com.hazelcast.map.EntryProcessor;
10
11 public class HExecuteOnEntries {
12
13     public static void main( String[] args ) throws UnknownHostException {
14         ClientConfig clientConfig = HConfig.getClientConfig();
15         final HazelcastInstance client = HazelcastClient.newHazelcastClient(clientConfig);
16
17         IMap<Long, Student> students = client.getMap("students");
18         students.executeOnEntries(new HEntryProcessor());
19
20         for (Entry<Long, Student> e : students.entrySet()) {
21             System.out.println(e.getKey() + " => " + e.getValue());
22         }
23     }
24 }
25
26 class HEntryProcessor implements EntryProcessor<Long, Student, String>, Serializable {
27     private static final long serialVersionUID = 1L;
28
29     @Override
30     public String process(Entry<Long, Student> e) {
31         Student student = e.getValue();
32         String name = student.getName();
33         if (name.equals(name.toLowerCase())) {
34             name = name.toUpperCase();
35             student.setName(name);
36         } else {
37             name = name.toLowerCase();
38             student.setName(name);
39         }
40
41         System.out.println("Processing = " + student);
42         e.setValue(student);
43
44         return name;
45     }
46 }

```

Przykład 6: `src/LabHazelcast/src/main/java/HExecuteOnEntries.java` {link}

Kompilacja:

```
mvn clean compile
```

Uruchomienie:

```
mvn exec:java -Dexec.mainClass="HExecuteOnEntries"
```

1.7 Client HPredicate

Aplikacja działająca w trybie *client* i wyszukująca studentów o nazwisku "Kowalski" oraz urodzonych przed 2000 rokiem przy pomocy *Predicate*.

```

1 import java.net.UnknownHostException;
2 import java.util.Collection;
3
4 import com.hazelcast.client.HazelcastClient;
5 import com.hazelcast.client.config.ClientConfig;

```

```

6 import com.hazelcast.core.HazelcastInstance;
7 import com.hazelcast.map.IMap;
8 import com.hazelcast.query.Predicate;
9 import com.hazelcast.query.Predicates;
10
11 public class HPredicate {
12
13     public static void main( String[] args ) throws UnknownHostException {
14         ClientConfig clientConfig = HConfig.getClientConfig();
15         final HazelcastInstance client = HazelcastClient.newHazelcastClient(clientConfig);
16         IMap<Long, Student> students = client.getMap("students");
17
18         Predicate<?,?> namePredicate = Predicates.equal( "name", "KOWALSKI" );
19         Predicate<?,?> birthyearPredicate = Predicates.lessThan("birthyear", 2000);
20
21         Collection<Student> kowalski = students.values(Predicates.and(namePredicate, ↵
22             birthyearPredicate));
23         for (Student s : kowalski) {
24             System.out.println(s);
25         }
26     }
27 }

```

Przykład 7: src/LabHazelcast/src/main/java/HPredicate.java {link}

Kompilacja:

```
mvn clean compile
```

Uruchomienie:

```
mvn exec:java -Dexec.mainClass="HPredicate"
```