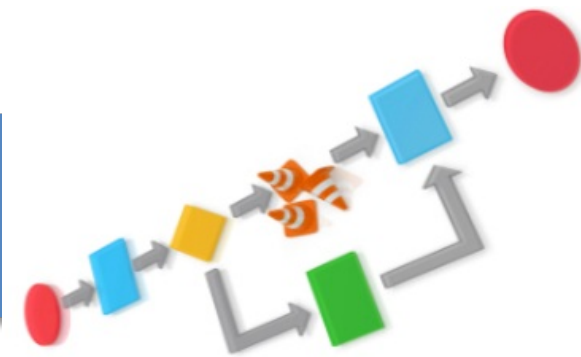


Elasticsearch

Jiří Šebek

b6b36nss



```
public final void onSensorChanged(SensorEvent event)
{
    m_flightIntensity = event.values[0];
    m_etAmblight.setText("" + m_flightIntensity + " lx");
}

private void resume()
{
    m_flightIntensity = SensorEvent.NORMAL;
}
```

Obsah

- Definice
- Instalace
- Implementace a pouziti v projektu

Obsah

- Elasticsearch = fulltextového vyhledávače vycházejícího z Apache Lucene.
- Disponuje RESTful rozhraním a nabízí vysokou dostupnost, rychlost a škálovatelnost.
- Java based technology
- Vhodny pro Big data

Vlastnosti

- Data v reálném čase
- Škálovatelnost
- Vysoká dostupnost
- Fulltextové vyhledávání
- RESTful API

Způsob uložení dat

- Elasticsearch je bezschémovou databází.
- Není proto třeba definovat strukturu databáze, protože ta se vytvoří sama na základě vložených dat.

Instalace/nastaveni

- stahnout z <https://www.elastic.co/downloads/past-releases>
- zvolit Elasticsearch v drop down a vybrat spravnou verzi (5.5.0 pro jdk8) a kliknout na download
- Na dalsi strance stahnout jako Zip pro windows. Tar pro linux
- Otevrit Zip a rozbalit obsah. Ve slozce bin najdete elasticsearch.bat. Spuste. *Spoustejte bat soubor pres cmd ! Uvidite chybove hlasky.*
- POZOR : pokud se objevi hlaska typu :

elasticsearch could not reserve enough space for object heap

`\config\jvm.options`

`-Xms512m`

`-Xmx512m`

Instalace/nastaveni

- Otevře v prohlizeci: `http://localhost:9200/`.
Port 9300 je take povolen jak cluster node
Port 9200 je pro REST komunikaci
- Otestujte pomoci curl prikazu z command line :
 - `curl http://localhost:9200/`

Test

- V insomnii/postmanovi si vytvorte request :
 - <http://localhost:9200/users/employee/1>
 - POST metoda
 - JSON body :

```
{
  "userId" : "1",
  "name" : "Rajesh",
  "userSettings" : {
    "gender" : "male",
    "occupation" : "CA",
    "hobby" : "chess"
  }
}
```


Test

- V insomnii/postmanovi si vytvorite request :
 - `http://localhost:9200/users/employee/1`
 - GET metoda

Test

- Zkusme search :
 - curl -XGET 'http://localhost:9200/users/employee/_search'
 - Pozor windows nema rad ' '.. proto nahradme za « »
 - curl -XGET
« http://localhost:9200/users/employee/_search »

Test

- Zkusme search s query:
 - `curl -XGET "http://localhost:9200/users/employee/_search -H " Content-Type: application/json" -d " {}"`
 - `curl -XGET 'http://localhost:9200/users/employee/_search' -H 'Content-Type: application/json' -d '{"query": { "match": {"name" : "Rajesh" } } }'`

Springboot

- Pom.xml

```
<dependency>
```

```
<groupId>org.elasticsearch</groupId>
```

```
<artifactId>elasticsearch</artifactId>
```

```
</dependency>
```

```
<dependency>
```

```
<groupId>org.elasticsearch.client</groupId>
```

```
<artifactId>transport</artifactId>
```

```
<version>5.0.0</version>
```

```
</dependency>
```

Springboot

- application.properties

```
# Local Elasticsearch config
```

```
elasticsearch.host=localhost
```

```
elasticsearch.port=9300
```

```
# App config
```

```
server.port=8102
```

```
spring.application.name=BootElastic
```

Springboot

- Domain object

```
public class User {  
    private String userId;  
    private String name;  
    private Date creationDate = new Date();  
    private Map<String, String> userSettings = new HashMap<>();  
    -- getter/setter methods  
}
```

Springboot

- config

@Configuration

```
public class config{
```

```
    @Value("${elasticsearch.host:localhost}")
```

```
    public String host;
```

```
    @Value("${elasticsearch.port:9300}")
```

```
    public int port;
```

```
    public String getHost() {
```

```
        return host;
```

```
}
```

```
public int getPort() {
```

```
    return port;
```

```
}
```

@Bean

```
public Client client(){
```

```
    TransportClient client = null;
```

```
    try{
```

```
        System.out.println("host:"+ host+"port:"+port);
```

```
        client = new PreBuiltTransportClient(Settings.EMPTY)
```

```
            .addTransportAddress(new InetSocketTransportAddress(InetAddress.getByName(host), port));
```

```
    } catch (UnknownHostException e) {
```

```
        e.printStackTrace();
```

```
    }
```

```
    return client;
```

```
}
```

```
}
```

Springboot

- Controller

```
@Autowired
```

```
Client client;
```

```
@PostMapping("/create")
```

```
public String create(@RequestBody User user) throws IOException {
```

```
    IndexResponse response = client.prepareIndex("users", "employee", user.getUserId())
```

```
        .setSource(jsonBuilder().startObject().field("name", user.getName())
```

```
            .field("userSettings", user.getUserSettings()).endObject()
```

```
        ).get();
```

```
    System.out.println("response id:"+response.getId());
```

```
    return response.getResult().toString();
```

```
}
```


Springboot

- Controller

```
@GetMapping("/view/{id}")
```

```
public Map<String, Object> view(@PathVariable final String id) {
```

```
    GetResponse getResponse = client.prepareGet("users", "employee", id).get();
```

```
    return getResponse.getSource();
```

```
}
```

Springboot

- Controller

```
@GetMapping("/view/name/{field}")

public Map<String, Object> searchByName(@PathVariable final String field) {

    Map<String, Object> map = null;

    SearchResponse response = client.prepareSearch("users")
    .setTypes("employee")
    .setSearchType(SearchType.QUERY_AND_FETCH).setQuery(QueryBuilders.matchQuery("name",
    field)).get();

    List<SearchHit> searchHits = Arrays.asList(response.getHits().getHits());

    map = searchHits.get(0).getSource();

    return map;

}
```

Springboot

- Controller

```
@GetMapping("/update/{id}")  
  
public String update(@PathVariable final String id) throws IOException {  
  
    UpdateRequest updateRequest = new UpdateRequest();  
  
    updateRequest.index("users").type("employee").id(id)  
  
        .doc(jsonBuilder().startObject().field("name", "Rajesh").endObject());  
  
    try {  
  
        UpdateResponse updateResponse = client.update(updateRequest).get();  
  
        System.out.println(updateResponse.status());  
  
        return updateResponse.status().toString();  
  
    } catch (InterruptedException | ExecutionException e) { System.out.println(e); }  
  
    return "Exception";  
  
}
```

Springboot

- Otestujte :
 - <http://localhost:8102/rest/users/create> ... POST
 - JSON body :

```
{
  "userId": "1",
  "name": "Sumit",
  "userSettings": {
    "gender": "male",
    "occupation": "CA",
    "hobby": "chess"
  }
}
```

Springboot

- Otestujte :
 - `http://localhost:8102/rest/users/view/1 ... GET`
- Otestujte :
 - Podobne delete i update