

**Modern intelligent IT**  
**Lab 6 (14.11.2016)**  
**Akira Imada**

**STUDENT AVDEY ANDREY**

```
import amqp from 'amqplib/callback_api';
import elasticsearch from 'elasticsearch';
import mongoose from 'mongoose';

import {
  mongoHost,
  rabbitMQHost,
  rabbitMQConfig,
  elasticHost,
  elasticConfig,
} from '../shared/config';

mongoose.connect(mongoHost);
const elasticClient = new elasticsearch.Client({ host: elasticHost });

const maxAttemptCount = 1000;
const modelName = 'message';
const Schema = mongoose.Schema;
const messageSchema = new Schema({
  RecordId: String,
  Classifications: [String],
  Fields: {},
});

mongoose.model(modelName, messageSchema);
const Model = mongoose.model(modelName);

amqp.connect(rabbitMQHost, (err, conn) => {
  conn.createChannel((chErr, ch) => {
    const chName = rabbitMQConfig.channel;

    ch.assertQueue(chName, { durable: false });
    ch.consume(chName, msg => {
      const data = JSON.parse(msg.content);

      switch (data.action) {
        case 'Publish': {
          saveMessage(data);
          return;
        }
      }
    });
  });
});
```

```
function saveMessage(data) {
  saveToMongo(data);
  saveToElastic(data);
}

function returnMessage(data, channel) {
  data.DeliveryAttempt++;
  if (data.DeliveryAttempt > maxAttemptCount) return;

  const jsonMessage = JSON.stringify(data);
  channel.sendToQueue(rabbitMQConfig.channel, Buffer.from(jsonMessage));
}
```

