



Laboratory 3

Microservices

During third laboratory students should get familiar with basic concepts of decomposition monolithic application into stand-alone modules based on microservices architecture. Projects should be based on the code developed during previous laboratories.

The following tasks must be completed:

1. Original Spring Boot project should be divided into two separate projects representing two stand-alone applications. One for category management and second for elements management. Each of the applications should make an use of private in-memory h2 database. Category management application should only contain mechanisms for category management. Elements management application should contain mechanisms for elements management as well as simplified mechanism for category management (in order to maintain relationships and hierarchy). (3 point).
2. Implementation of inter-applications event-based communication. When removing existing or adding new category the elements management application should be notified in order to remove elements or create new simplified category record in the database. As event communication REST services can be used. (3 point).
3. New Spring Boot application based on Spring Cloud Gateway should be created. The application should contain routing rules for category and elements management applications. (1 point).

Attention: It is required to show that HTTP requests are working as assumed. The task does not required building view layer. It is enough to use HTTP request testing tool, for example HTTP Client delivered with IntelliJ IDEA.



Example data model – RPG game heroes representation (access modifiers, accessors, constructors and other methods omitted for the sake of simplification):

```
class Profession {          class Character {
    String name;             String name;
    double moveSpeed;        Profession profession;
    int baseArmor;           int level;
}                             }
```

Example model for category management application:

```
class Profession {
    String name;
    double moveSpeed;
    int baseArmor;
}
```

Example model for element management application:

```
class Profession {          class Character {
    String name;             String name;
}                             Profession profession;
                             int level;
                             }
```