Katarina Smolikova, 1528686

Advanced services engineering, 2017

Log-based e-commerce recommendation system

Mini project documentation

Basic idea

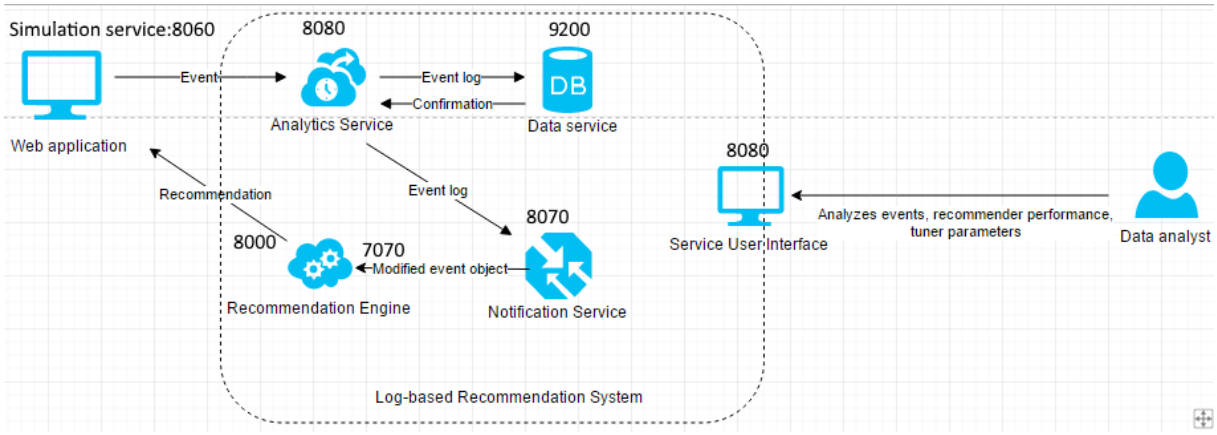
The Log-based e-commerce recommendation system is a service designed especially for e-shops but also for any website where there is some content recommendation required. The service uses the data about user interaction which is saved for analytic and safety purposes to train a machine-learning recommendation engine. This way, the data from the website is only sent into one service which can be then used for both analysis and recommendations. That is not only more efficient, but also more convenient for the website owner.

Demo

The service can be build using Docker. The service itself consists of 5 smaller services each running on its own port:

- **Analytic service (8080):** Core service responsible for the data analysis and sending the information of an event to the data service and possiblity to notification service. This service also hosts user interface for website administrator.
- **Data service (9200):** Data service where events can be saved either directly or through analytic service. The service is based on Elasticsearch. The data from data service can be analyzed using Kibana on port 5601.
- **Notification service (8070):** Resends data from analytic service to recommendation service.
- **Recommendation service (9000):** The service consists of Event server on port 7070, Event engine on port 8000. The service is based on PredictionIO and uses machine learning to get user-based recommendations based on users previous interactions.
- **Simulation service (8060):** Only for purposes of this demo we created the simulation service, which uses data from Expedia.com and feeds them to our Log-based e-commerce recommendation system at fast random intervals.

How the system works can be seen in following picture:

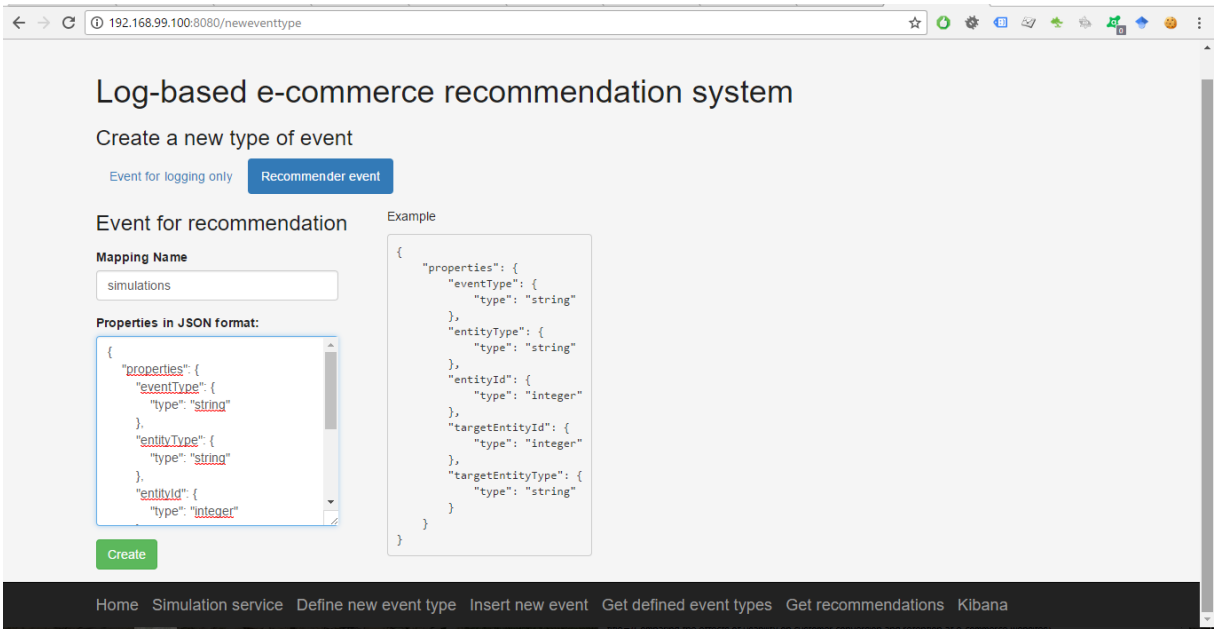


Installing the service

1) Open docker console
2) Make sure it runs on IP http://192.168.99.100 (if not, change it in the docker config): docker-machine ip
3) Go to analytic-service directory and run: mvn package docker:build
4) Go to notification-service directory and run: mvn package docker:build
5) Go to simulation-service directory and run: mvn package docker:build
6) Go to recommendation-service directory and run: docker build . -t predictionio
7) Go back to project root folder
8) Run: docker-compose up
9) Wait untill all services start (it might take few minutes)
10) The service interface can be found at: http://192.168.99.100:8080

Using the service

1) Before sending any event to the service, types of events that will be sent need to be defined. The event can be either only for logging purposes or also for recommendation purporses.
   If you wisht to use simulation service for feeding the events, open http://192.168.99.100:8080/neweventtype . Create mapping called „simulations“ and copy the properties from the example. Properties in the example are required for recommender events. However, more properties can be added at your convenience.



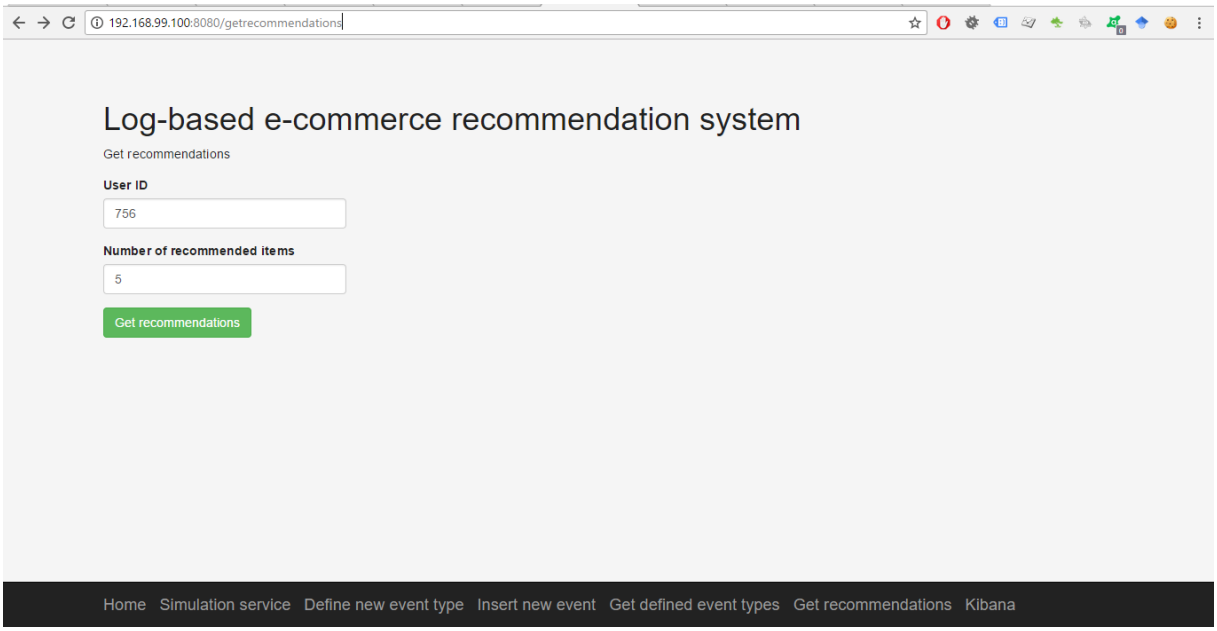
Creating a new mapping for recommender event

2) You can check that the mapping was created here: http://192.168.99.100:8080/eventtypes There can be more event types sent to the recommender. For example different event type for bookmarking, booking, rating etc. However, all the event types which should are used in recommender have to be saved in *recommender_events* index.

3) You can now send the events which you defined to the service through REST http://192.168.99.100:8080/insertEvent where request body is the defined event in JSON form. If you want to use the simulation service for inserting events, you can start the simulation service here: http://192.168.99.100:8060/

4) After sufficient amount of new data has been sent to the recommendation engine, the recommendations can be retrieved. However, if there is a lot of data which is was previosly unknown to the recommender, in order to be able to get the recommendations we first need to retrain the recommender. In the docker we need to run: *docker exec predictionio-container /bin/sh -c "cd /quickstartapp/MyRecommendation && pio train && pio deploy --ip 0.0.0.0&"*

5) After the training is complete we can get the recommendations here: http://192.168.99.100:8080/getrecommendations



We can also access the notification throught REST call http://192.168.99.100:8070/getrecommendation/[USER_ID]/[NB_OF_RECOMMENDED_ITEMS]