

A.S. Prakharenka (Francisk Skorina Gomel State University, Gomel),
Scientific adviser **V.D. Liauchuk**, Ph.D. in technics, associate professor

DEVELOPMENT OF DATA CONVERSION SUBSYSTEM IN THE ONLINE BOOKMAKER PLATFORM

Many people are fond of sports. In this segment there are gambling people who bet on the results of matches. It is for them that an online betting platform, which is a web application, is required.

The operation of any web application is impossible without communication between the client and the server through requests. Their processing is resource intensive. Most of the cost of each request is related to the time it passes back and forth between the client and the server. One way to reduce the number of calls is to use a DTO (Data Transfer Object). It combines data that was transmitted by multiple requests into one. In most cases, the data inside the DTO are obtained from more than one business object. DTO has no behavior, so it cannot retrieve data from objects. Also, the DTO is not aware of the business objects themselves, which makes it possible to reuse the DTO in various contexts. Similarly, business objects are not aware of the existence of a DTO, since this change in its logic may lead to the fact that it will require constant changes to the code in the server part, and this in turn will lead to difficulties in the development process. Therefore, the best solution is the use of converters and populators that create DTO from business objects and vice versa.

The goal of the project is to create a subsystem for an online betting platform that transforms server business objects into DTO and vice versa. The developed subsystem consists of a number of converters. They are used to convert server objects to DTO. In turn, each such converter contains a set of populators in which the process of converting certain fields of business objects takes place. It is also possible to expand the set of populators for each converter, which will allow you to customize the DTO for possible changes to its requirements and will not lead to editing previously written code in the event of a subsequent upgrade of the entire system. Moreover, this solution significantly affects the speed of execution of requests between the client and the server and also increases the stability of the entire web-application.