

Задания к курсу

“Разработка прикладных компьютерных систем”

3 курс, 2021-2022 уч. г. (Часть 4)

18. Реализуйте dispenser gRPC-сервер, принимающий запросы на валидацию данных. В запрос могут входить записи, содержащие (возможно, только часть) следующие данные:

- ФИО (макс. 1);
- Номер телефона (может быть несколько);
- Электронная почта (может быть несколько);
- Адрес (может быть несколько);
- Номер паспорта (макс. 1);
- Дата рождения (макс. 1).

Сервис должен:

1. Проверить входные данные на соответствие условиям по количеству компонент в каждой отдельной записи;
2. Присвоить каждой записи уникальный внутренний идентификатор типа System.Guid (с отправкой его вместе с результатами валидации в клиентское приложение);
3. Разобрать каждую запись на составляющие её компоненты и передать их на валидацию серверам, описанным в задании 19; wgatter’ы для gRPC-клиентов необходимо при этом получить из DI-контейнера;
4. При помощи обратных вызовов со стороны серверов валидации, сформировать результат валидации коллекции записей, который отправить клиентскому приложению при помощи RPC со стороны клиентского приложения (проследите, чтобы отправка результатов валидации по одному запросу происходила единожды).

19. Реализуйте gRPC-сервера, обрабатывающие запросы на валидацию следующих данных (каждый сервер обрабатывает свои данные):

- ФИО;
- Номер телефона;
- Электронная почта;
- Адрес;
- Номер паспорта;
- Дата рождения.

Структура каждого сервера:

- При приёме запроса на стороне сервера, данные из запроса помещаются во входную очередь RabbitMQ;
- Запрос из очереди извлекается компонентом [валидации], зарегистрированным в DI-контейнере;
- Компонент валидации делегирует работу по валидации на реализацию интерфейса валидации данных, которая поступает в конструктор компонента валидации данных и которая может быть настроена в `deployment time behaviour` при помощи конфигурационного файла `appsettings.json`;
- По завершении валидации данных, результаты валидации с привязкой к конкретной записи необходимо отправить в выходную очередь RabbitMQ от компонента валидации к серверу;
- Сервер, получая результаты валидации из выходной очереди RabbitMQ, отправляет их dispatcher-серверу.

При реализации серверов, допускается использование хранилищ данных (баз данных), на основе которых может строиться логика валидации. При этом:

- строка подключения к базе данных должна быть размещена в конфигурационном файле `appsettings.json`;
- Объекты, связанные с обращением к контексту БД, должны порождаться фабрикой, получаемой из DI-контейнера.

20. Реализуйте оконное приложение на WPF, позволяющее составлять запросы на валидацию, отправлять их, получать результаты валидации и отображать результаты клиенту, при помощи элементов управления. Пользовательский интерфейс должен быть интуитивно понятен “для домохозяек”. Структуру пользовательского интерфейса, необходимых для взаимодействия с backend компонент и т. д., определите самостоятельно. Приложение должно быть реализовано на базе архитектуры MVVM.