**Как работает метод конвейера MapFallbackToController.**

Метод конвейера MapFallbackToController – добавялет в список маршрутов роутинга ендпойнт с наименьшим приоритетом, т.е. на этот эндпойнт будут перенаправляться все запросы, которым не найдено соответствие по имеющимся правилам роутинга.

Удобно использовать этот функционал для SPA-приложений - он разруливает все запросы, которые не попадают на WebApi и переадресует их в index.html -> в react.

Метод MapFallbackToController необходим в этом случае, т.к. необходима переадресация на файл index.html, т.е. тогда, когда на одном веб-сервере работает и webapi, и сам реакт. В этом случае любой запрос обрабатывается веб-сервером по конвейеру запросов и в зависимости от указанного url ищется подходящий энд-пойнт. Если идет обращение к webapi – такой метод находится и возвращается требуемый ответ. Но если идет обращение к страничке реакта – точка доступа не находится и получаем 404 NotFound. Или, если совпадают наименования энд-пойнтов, но не совпадают типы – получаем 405 BadRequest.

А нам нужно, что если метод не найден – передаем управление к реакту, т.е. просто возвращаем index.html, а там уже реакт разруливает роутинг.

Таким образом, этот метод служит для возврата файла index.html если не найден ни один из методов webapi.

В основном блоке проекта Microservice мы не используем MapFallbackToController в контроллере MainApi, так как там нет нужды переадресовывать конвейер – у нему обращаются ТОЛЬКО по имеющимся методам webapi, а сам реакт – обслуживает отдельный сервер. Если потребуется перенести реакт на один из webapi – то да, придется доавлять в этот webapi и метод MapFallbackToController.

Для запуска в режиме отладки – для разработки:

Используется файл .env – конфигуратор отладочного сервера react, в нем конфигурируется другой порт – 5011 – чтобы не совпадал с портом webapi – 5010. Также настраиваем CORS в webapi для порта 5011.

Запускаем из каталога -> D:\Projects\MicroServices\Identity\client-app приложение Реакт командой npm start

И запускаем также webapi -> можно отлаживать по адресу URL -> <https://192.168.1.10:5011/login> или <https://192.168.1.10:5011/newaccount>

Если нам нужно запустить Identity в составе проекта – нужно сделать релиз nom run build -> тогда будет обновлена папка wwwroot, которую будет использовать проект Identity в режиме отладки.

Для создания релиза -> npm run build -> создается (или обновляется) папка wwwroot в каталоге Identity, ее копируем в папку с exe-ником, например D:\Projects\MicroServices\Identity\bin\Debug\net7.0

И там запускаем exe, в этом случае реакт и webapi работают по одному адресу – <https://192.168.1.10:5010>, в этом случае не запускается отладочный веб-сервер.