МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Автоматизированные системы управления»

ОТЧЕТ

о прохождении производственной практики

на ИООО «ЭПАМ СИСТЕМЗ»

Руководитель от университета Тимашкова Л. А.

(подпись) (дата)

Руководитель от предприятия Iryna Kastsevich

(подпись) (дата)

Студент гр. АСОИ – 181 Остапенко А. К.

(подпись) (дата)

Могилев, 2021 г.

Содержание

[Введение 3](#_Toc80633236)

[О компании 4](#_Toc80633237)

[Индивидуальное задание 7](#_Toc80633238)

[Алгоритм синхронизации 7](#_Toc80633239)

[Введение в Salesforce 7](#_Toc80633240)

[Подключение к Salesforce 9](#_Toc80633241)

[Получение данных из Salesforce 10](#_Toc80633242)

[Обновление данных в Salesforce 11](#_Toc80633243)

[Создание данных в Salesforce 12](#_Toc80633244)

[Удаление данных в Salesforce 12](#_Toc80633245)

[Получение удаленных данных из Salesforce 13](#_Toc80633246)

[Работа с базой данных 14](#_Toc80633247)

[Заключение 16](#_Toc80633248)

[Список использованной литературы 17](#_Toc80633249)

# Введение

Технологическая практика играет огромную роль в учебном процессе.

Целью технологической практики является изучение организации и управления производством, проверка возможностей будущего специалиста самостоятельно выполнять профессиональные функции.

Задачей технологической практики является приобретение учащимися профессиональных умений и навыков по специальности; закрепление, углубление и систематизация знаний по специальным предметам; изучение технологии и организации производства; приобретение практического опыта, развитие профессионального мышления.

Практика проходила на ООО «Техартгруп» в период с 08.06.2021 г. по 05.07.2021 г..

В ходе практики было выдано индивидуальное задание, которое связано с разработкой приложения на платформе .NET Core для синхронизации данных между базой данных и приложением на платформе Salesforce.

Отчет состоит из основных разделов: содержание, введение, информация о компании, индивидуальное задание, заключение и список использованной литературы.

# О компании

История компании берёт начало в 2002 году в США. Тогда компанию уместнее было бы назвать стартапом. Она состояла из нескольких человек, в основном, близких друзей и знакомых. Но уже тогда в команде были уверены: предлагать качественные технические решения - подлинное искусство. Так у компании появилось имя – iTechArt означает Information, Technologies, Art.

В 2008 году компания официально стала ООО «Техартгруп». Поначалу основатели не только развивали бизнес, но и сами писали клиентский код. Всё изменилось с получением статуса Microsoft Gold Certified Partner. Перспективное партнерство запустило «сарафанное радио», которое привело новых клиентов.

В 2009 году дочерняя компании Google доверила iTechArt крупный проект на Java. Для такого заказа нужно было больше разработчиков. В итоге штат компании вырос до 40 человек. Постепенно расширяется присутствие iTechArt в США: в компании появились первые американские менеджеры по продажам.

За несколько месяцев в 2010 году в США сформировалась полноценная sales-команда. Список клиентов пополнялся авторитетными именами, например, Gain Capital (Forex.com). Эксперт финансового рынка высоко оценил нашу экспертизу в мобильной разработке. Так началось сотрудничество iTechArt с FinTech-компаниями.

Производительность компьютеров и браузеров растёт, компании понадобилось ещё больше Front-End-инженеров. Поэтому появились собственные курсы для опытных разработчиков. Так в 2011 году в iTechArt начала формироваться экспертиза в Front-End.

В 2012 году среди клиентов стало больше быстрорастущих компаний. ITechArt решил и далее фокусироваться на работе с амбициозными стартапами. Сформировался HR департамент, появились программы обучения сотрудников, запустились курсы Students Lab по .NET, Java, PHP, Mobile и QA.

В 2013 году компания насчитывала уже 300 человек, количество проектов тоже увеличилось. За iTechArt на международном рынке закрепился имидж компании со штатом высококлассных инженеров по разработке и тестированию ПО.

Летом 2014 года компания инициировала внутренний хакатон: 14 команд за 24 часа написали приложения для улучшения жизни в офисе.

В 2015 году компания начала формировать внутреннюю базу знаний компании. За основу взяли материалы с внутренних обучающих сессий Dev2Dev. Прошел первый iTechForum – масштабная техническая конференция компании и соревнования по программированию. Команда разрослась до 470 человек.

Летом 2016 года прошел iTechForum – техническая конференция на 300 участников. В хакатоне свои проекты разработали 13 команд (тема Smart City). В конце года компания выступила спонсором профессионального события New York Nasdaq Startup CTO Summit. К тому времени количество сотрудников уже превысило 600 человек.

В 2017 году iTechArt открыл новые офисы: в мае – в Нью-Йорке для sales-команды, в июле – первый белорусский региональный офис iTechArt в Могилеве. К концу года появились офисы в Бресте, Витебске, Гродно. Dev2Dev-обучение перешло в формат вебинаров. А летом снова состоялась конференция iTechForum с 450 участниками, гостем из Microsoft и хакатоном по теме Smart Office на 15 команд.

В 2018 году открылся офис компании в Гомеле, а также ещё один минский офис. В этому году компания состаяла уже из 1200 сотрудников. Количество клиентов выросло до 200+, пришли новые, именитые компании: PayPal, Forex и Yale University. В пятый раз прошла двухдневная конференция iTechForum и приуроченный к ней хакатон в Минске. Итог: более 600 участников и 20 спикеров.

В 2019 году компания вошла в список самых быстрорастущих компаний США (Inc5000) и в топ IT-компаний (Clutch, GoodFirms, AppFutura, Wadline). Провели 28 iTechMeetups для разработчиков и тестировщиков в 6 городах. А ещё 20 курсов в Students Lab. Число активных клиентов выросло до 300+, а штат компании – до 1655 человек.

В 2020 году открылись офисы в Кремниевой долине и Польше. Компания стала Select Consulting Partner для Amazon Web Services (AWS), была признана Fastest Growing IT Company of the Year на 2020 IT World Awards. IAOP добавили ее в список Global Outsourcing 100® companies for 2020. А так же снова оказалась в списке Deloitte’s Technology Fast 500.

# Индивидуальное задание

Темой индивидуального задания была выбрана тема «Разработка приложения на платформе .NET Core для синхронизации данных между базой данных и приложением на платформе Salesforce».

Для полного раскрытия данной темы необходимо для начала разобраться с платформой Salesforce и принципами работы с ней, алгоритмом синхронизации и работой с базой данных.

## Алгоритм синхронизации

В обеих системах записи могут быть добавлены, удалены или обновлены.

Алгоритм синхронизациия будет состоять из нескольких шагов:

1. Получение данных из обеих систем.

2. Сравнение данных для выяснения приоритетной информации.

3. Добавление новых записей в систему

4. Обновление измененных записей

5. Поиск удаленных записей и удаление их в другой системе

## Введение в Salesforce

Salesforce – популярная облачная CRM (Customer Relationship Management), которой владеет одноимённая компания из США. Gartner в 2018 году назвал Salesforce лучшим PaaS-решением (платформа как услуга, PaaS — это один из способов предоставления клиенту готовой программной среды) для бизнеса, а в 2019 Salesforce признали лидером на рынке CRM-систем за последние 6 лет. Salesforce уже использует более 150 тысяч компаний, среди которых – Sony, Financial Times, Toyota и многие другие известные мировые бренды.

Компания Salesforce динамично развивает не только Customer Relationship Management, но и экосистему для разработки под эту платформу. По данным исследования международной консалтинговой компании IDC, уже к 2022 году платформа Salesforce создаст в мире около 3,3 миллионов рабочих мест, включая позиции для разработчиков.

Salesforce представляет собой облачную платформу для управления бизнес-процессами в сфере продаж, клиентского сервиса, цифрового маркетинга. И ключевые продукты для этих целей – Sales Cloud, Service Cloud, Marketing Cloud. Также среди востребованных решений для бизнеса – Commerce Cloud (для электронной коммерции), Financial Services Cloud (для компаний из финансовой сферы), Health Cloud (для медицинских организаций). Salesforce предлагает и дополнительные компоненты, которые можно внедрять в их продукты. Например, CPQ (Configure, Price, Quote) – инструмент для автоматической генерации ценовых предложений, Chatter – корпоративная социальная сеть.

Работает Salesforce CRM по модели SaaS (форме вычислений в облаке). Пользователям CRM-системы предоставляется готовое программное обеспечение, доступ к которому открытый через браузер или мобильное приложение. CRM-система ориентирована на малый, средний и крупный бизнес. Предлагает все необходимое для поиска, удержания клиентской базы, совершения сделки и прочее.

Salesforcе можно адаптировать под потребности бизнеса двумя способами – с помощью конфигурации и кастомизации. Разработчики преимущественно занимаются кастомизацией Salesforce, но частично могут выполнять и конфигурацию.

При конфигурации функционал CRM настраивается через пользовательский интерфейс администратора Salesforcе без использования кода. Так можно создать таблицы, объекты, поля, сформировать между ними взаимосвязи, установить правила для автоматизации рабочих процессов. Крупный бизнес обычно для конфигурации СRM нанимает в штат отдельного специалиста – Salesforcе-администратора.

Кастомизация нужна, если возможностей конфигурации не хватает, чтобы автоматизировать сложную бизнес-логику.

## Подключение к Salesforce

Для доступа к salesforce приложению будет использоваться REST API.

REST API — это надежный и удобный интерфейс веб-служб на основе REST для взаимодействия с системой Salesforce. Преимущества данного интерфейса заключаются в удобстве интеграции и развертывания, поэтому он идеально подходит для использования в веб-проектах и приложениях для мобильных устройств.

Подключение к Salesforce осуществлялось по протоколу OAuth 2.0, поток – Username-Password.

Для подключения к приложению было созданно Сonnected Аpp – это фреймворк, который позволяет внешнему приложению интегрироваться с Salesforce с помощью API-интерфейсов и стандартных протоколов, таких как SAML, OAuth и OpenID Connect.

Пример, запроса на получение токена доступа рисунок 1. Client\_id и clinet\_secret данные созданного на стороне Salesforce приложения для интеграции.

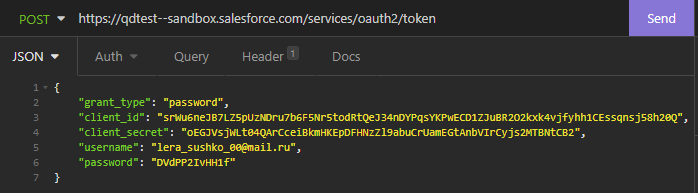


Рисунок 1 – Пример запроса для получения токена доступа

После успешного подключения будет получен ответ рисунок 2.

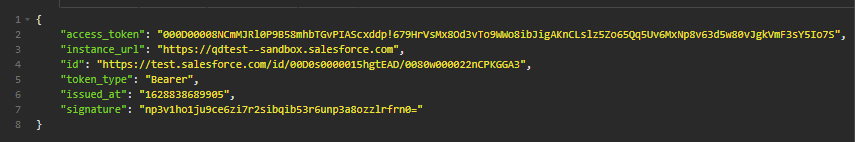


Рисунок 2 – Пример ответа с токеном доступа от Salesforce

Далее токен доступа, полученный в ответе от Salesforce, будет использоваться в HTTP запросах в заголовке рисунок 3.

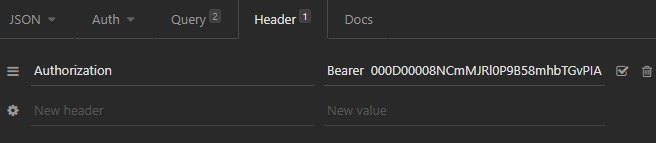


Рисунок 3 – Пример использования токена доступа в HTTP заголовке

## Получение данных из Salesforce

Для получения данных из Salesforce через REST API был использован SOQL(Salesforce Object Query Language) запрос.

Это язык запросов Salesforce, разработанный для работы с базой данных SFDC. Он может искать запись по заданному критерию только в одном sObject.

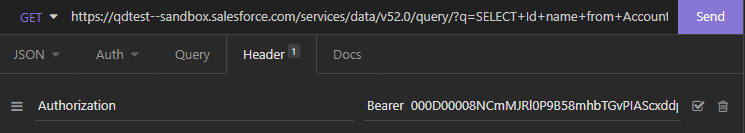


Рисунок 4 – Пример запроса на получение данных с использование SOQL запроса

Такой запрос возвращает все результаты в одном ответе или, если необходимо, возвращает часть результатов и идентификатор, который может использоваться для получения оставшихся результатов рисунок 5.



Рисунок 5 – Пример ответа на запрос с SOQL запросом

## Обновление данных в Salesforce

Для обновления записей в Salesforce использовался метод PATCH с коллекциями sObject, что позволяет одним запросом обновить 200 записей.

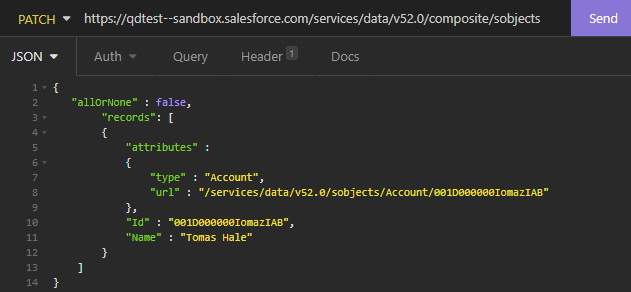


Рисунок 6 – Пример запроса на обновление данных в Salesforce

## Создание данных в Salesforce

Запрос на создание данных в Salesforce похож на запрос на обновление, только вместо метода PATCH используется метод POST. Один запрос может включать в себя до 200 объектов.

Наличие в запросе информации о типе объекта обязательное условие.

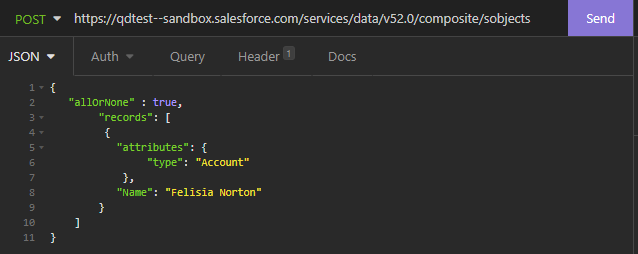


Рисунок 7 – Пример запроса на добавление данных в Salesforce

## Удаление данных в Salesforce

Для удаления данных из Salesforce используется запрос DELETE с параметрами. Этот запрос относиться к категории сложных и позволяет удалять записи разных типов данных в Salesforce за один запрос.

Параметер ids – обязательный, содержит уникальные идентификаторы записей в Salesforce.

Параметр allOrNone – необязательный, позволяет откатить полностью операцию в случае ошибки.

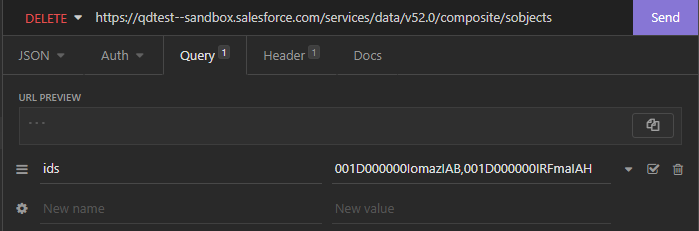


Рисунок 8 – Пример запроса на удаление данных из Salesforce

## Получение удаленных данных из Salesforce

После удаление Salesforce перестает возвращать запись через REST API. Удаленные данные помещаются в корзину на 15 дней. В течение этого времени запись может быть востановлена администратором.

Для получение удаленных записей используется запрос рисунок 9. Получить удаленные записи можно только для одного объекта за один запрос.

Параметры start и end – обязательны и обозначают период времени, в течение которого записи были удалены.

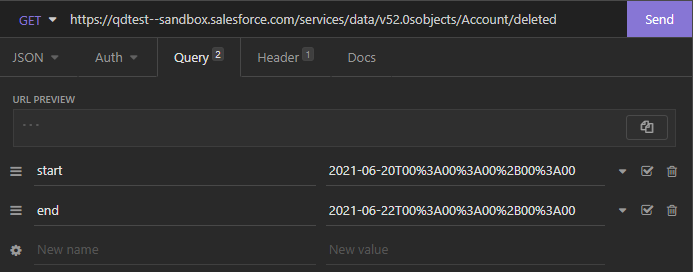


Рисунок 9 – Пример запроса на получение удаленных данных в Salesforce

На рисунке 10 показан пример с информацией об удаленных данных за указанный промежуток времени.

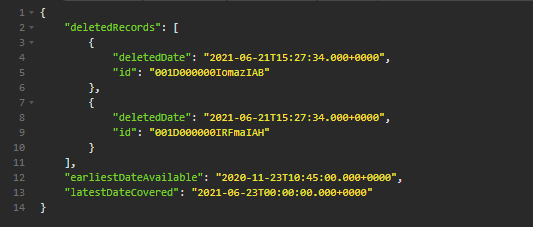


Рисунок 10 – Пример ответа на запроса для получение удаленных данных в Salesforce

## Работа с базой данных

Для работы с базой данных был выбран Entity Framework Core.

Entity Framework Core (EF Core) представляет собой объектно-ориентированную, легковесную и расширяемую технологию от компании Microsoft для доступа к данным. EF Core является ORM-инструментом (object-relational mapping - отображения данных на реальные объекты). То есть EF Core позволяет работать базами данных, но представляет собой более высокий уровень абстракции: EF Core позволяет абстрагироваться от самой базы данных и ее таблиц и работать с данными независимо от типа хранилища.

Взаимодействие с базой данных в Entity Framework Core происходит посредством специального класса – контекста данных. Он включает в себя DbSet/DbSet<TEntity>, которые представляет набор объектов, хранящихся в базе данных.

Для добавления объекта используется метод Add, определенный у класса DbSet, в который передается добавляемый объект. Метод Add устанавливает значение Added в качестве состояния нового объекта. А метод SaveChanges сгенерирует выражение INSERT для вставки модели в таблицу.

Удаление производится с помощью метода Remove. Данный метод установит статус объекта в Deleted, благодаря чему Entity Framework при выполнении метода SaveChanges сгенерирует SQL-выражение DELETE.

При изменении объекта Entity Framework сам отслеживает все изменения, и когда вызывается метод SaveChanges, будет сформировано SQL-выражение UPDATE для данного объекта, которое обновит объект в базе данных. Но если объект не отслеживается контектом, то используется метод Update, который установит значение Modified.

# Заключение

За период прохождения технологической практики на ООО «Техартгруп» были приобретены знания, навыки и умения, в дальнейшем необходимые для активной работы в освоении будущей специальности, а именно:

1. Были получены необходимые знания для работы с платформой Salesforce.

2. Были получены навыки работы с Entity Framework Core.

Индивидуальное задание, связанное с синхронизации между базой данных и приложением на платформе Salesforce, было успешно выполнено.

Также были получены базовые знания о компании ООО «Техартгруп», в частности, информация об истории развития компании, целях и задачах функционирования, выпускаемой продукции, организационной структуре, а также освещены вопросы, касающиеся охраны труда.

# Список использованной литературы

1. Metanit Entity Framework Core [Электронный ресурс] – Режим доступа: https://metanit.com/sharp/entityframeworkcore/1.1.php
2. Microsoft Docs Entity Framework Core [Электронный ресурс] – Режим доступа: https://docs.microsoft.com/en-us/ef/core/
3. Salesforce REST API Developer Guide [Электронный ресурс] – Режим доступа: https://developer.salesforce.com/docs/atlas.en-us.api\_rest.meta/api\_rest/ intro\_what\_is\_rest\_api.htm
4. Salesforce Help [Электронный ресурс] – Режим доступа: https://help.salesforce.com/s/?language=ru
5. iTechArt [Электронный ресурс] – Режим доступа: https://www.itechart.by