

Аннотация рабочей программы дисциплины (модуля) Б1.В.ДВ.06.02 «Интеграция информационных систем»

Цель и задачи дисциплины

Цель дисциплины

Целью изучения дисциплины является приобретение знаний о сетевых протоколах, используемых при интеграции систем, инструментах и нотации для аналитики в области интеграции, инструментах для проектирования интеграционных сервисов, инструментах для тестирования интеграционных сервисов, архитектурных подходах и шаблонах к проектированию интеграционных сервисов.

Задачи дисциплины

Основными задачами изучения дисциплины являются:

- изучение основных принципов интеграции информационных систем;
- знакомство с протоколами и инструментами интегрирования информационных систем;
- знакомство с архитектурными подходами и шаблонами к проектированию API;
- знакомство с процессами тестирования, прохождения контрольных испытаний;
- выработка практических навыков аналитического и экспериментального исследования основных методов и средств, используемых в области, изучаемой в рамках данной дисциплины.

Формируемые компетенции и индикаторы их достижения по дисциплине

Код компетенции	Содержание компетенции	Код и наименование индикатора достижения компетенции
ПКС-4	Способен проектировать программные интерфейсы	ПКС-4.1. Знать основные принципы проектирования программных интерфейсов. ПКС-4.2. Уметь использовать принципы проектирования программных интерфейсов. ПКС-4.3. Иметь навыки проектирования программных интерфейсов.

Содержание разделов дисциплины

Тема 1. Введение в интеграцию информационных систем

Интеграция систем как процесс настройки «связей» между различными ИС для получения единого информационного пространства и упрощения работы с бизнес-процессами, находящимися на стыке работы систем. Интеграционный процесс включает в себя: интеграцию необходимых приложений; интеграцию данных.

Тема 2. Основы протокола прикладного уровня передачи данных HTTP 1.1

Работа с открытым API с помощью Postman: авторизация через http заголовки, составление запроса (заголовки и тело), разбор ответов, создание коллекции, работа коллекции запросов с помощью переменных окружения, создание автотестов, подключение набора данных для работы автотестов.

Тема 3. Протокол REST (передача состояния представления) как способ создания API с помощью протокола HTTP

Подключение и работа к REST API: выполнение операций CRUD(L), разбор http ошибок сервера, разбор path к ресурсам, разбор best best practices в проектировании.

Тема 4. Знакомство с нотациями UML и BPMN

Введение в UML. Определение и назначение. История UML. Способы применения UML. Классификация диаграмм. История создания BPMN как стандарта графического языка моделирования бизнес-процессов. Обзор программных средств с поддержкой BPMN. Основные элементы BPMN.

Тема 5. Протокол SOAP для обмена структурированными сообщениями в распределённой вычислительной среде

Появление, развитие и актуальность SOAP API. Протокол SOAP как стандарт веб-служб для взаимодействия приложений. Особенности SOAP API. Работа с открытым SOAP сервисом: разбор WSDL описания, разбор типов в XSD для составления тела запроса, указание заголовков SOAPAction, составление коллекции для всех операций сервиса, подстановка значений из data-файла для автоматизации тестирования операций.

Тема 6. Протокол gRPC - высокопроизводительный фреймворк для вызов удаленных процедур (RPC)

gRPC (Remote Procedure Calls) как система удалённого вызова процедур (RPC) с открытым исходным кодом. Функции как аутентификация, двунаправленная потоковая передача и управление потоком, блокирующие или неблокирующие привязки, отмена и тайм-ауты. Подключение к gRPC сервисам: импорт в Postman описания gRPC сервиса, составление запросов для обращения к сервису, отправка запросов в unary сервис и в client streaming сервис, обращение к server streaming сервис, Составление собственного proto описания gRPC сервиса.

Тема 7. Протокол GraphQL, как язык и механизм обработки запросов клиентов к программным интерфейсам приложений

Протокол GraphQL язык запросов к графам. Язык не для прямого взаимодействия с базой данных, а для определения контракта, через который клиент коммуницирует с API-сервером. Спецификация GraphQL.