Программирование web-сервисов

ПОИТ-4, ПОИБМС-4

Коллоквиум

1. Поясните понятие SOA

**SOA: Service-oriented architecture**, **сервис-ориентированная архитектура** – парадигма разработки программного обеспечения, основанная на применении распределенных слабосвязанных компонентов, обеспечивающих стандартные интерфейсы.

2. Перечислите типовые программные компоненты SOA

**SOA:** компоненты распределенной системы SOA – узлы - сервисы.

**SOA:** **сервис** – это видимый ресурс, выполняющий повторяющуюся задачу и описанный внешней инструкцией.

3. Поясните понятие и назначение ESB SOA.

**ESB – Enterprise Service Bus:** программный компонент обеспечивающий обмен сообщениями между различными информационными системами, имеющих сервис-ориентированную структуру.

4. Поясните понятие и назначение Registry SOA

**SOA Registry –** реестр сервисов, информация о сервисах и их интерфейсах.

5. Поясните понятие и назначение Service broker SOA.

**Service broker** –программный компонент, позволяющий трансформировать запрос пользователя в системе в запуск и координированную работу.

6. Поясните понятие и назначение ESB SOA.

Смотреть 3 вопрос

7. Поясните понятие и назначение Workflow Engine SOA.

**Workflow Engine –** программный компонент, предназначенный для построить (построить модель бизнес процесса) и выполнить бизнес процесс на основе группы сервисов. Другими словами разработать новый сервис на основе последовательного выполнения нескольких сервисов.

8. Поясните понятие и назначение SOA Supervisor.

**SOA Supervisor** – служебный сервис, предназначенный для управления и мониторинга других служебных сервисов.

9. Поясните понятие web-сервис.

Веб сервис - это приложение которое предоставляет открытый интерфейс для использования другим приложением в веб

10. Перечислите известные вам способы клиент-серверного взаимодействия.

Дуплекс, полудуплекс, однонаправленная, потоковая, издатель-подписчик

11. Перечислите два типа сервисов (по типу интерфейсов).

**REST, RPC**

12. Что такое REST?

**REST**: **Representational State Transfer** - передача состояния представления.

**REST**: представление – это URI (идентификатор ресурса), управление ресурсом с помощью глаголов (GET, POST, PUT, DELETE).

**REST**: Архитектурный стиль взаимодействия компонентов распределенного приложения.

13. Каким образом представляется REST-сервис пользователю(разработчику)?

**REST**: представление – это URI (идентификатор ресурса), управление ресурсом с помощью глаголов (GET, POST, PUT, DELETE).

14. Перечислите применяемые http-методы REST-сервисами и поясните их назначение.

**REST:** GET (select), POST(insert), PUT (update), DELETE (delete).

15. Перечислите основные ограничения, накладываемые на REST-сервисов.

- модель клиент-сервер;

- отсутствие состояния на стороне сервера, сохранение состояния допускается на стороне клиента, допускается сохранение состояния в другом сервисе (например, в БД);

- кэширование на стороне клиента, сервер явно управляет кэшированием;

- единообразие интерфейсов (идентификация ресурсов, манипуляция ресурсами через представления, самодостаточные сообщения, HATEOAS);

- для клиента сервер должен представляться конечным;

- код по требованию: допускается (необязательно) выгрузка на клиент апплетов или сценариев для расширения его функциональности.

16. Поясните понятие HATEOAS, какой получает эффект ло применения HATEOAS?

**HATEOAS: Hypermedia As The Engine Of Application State** – гипермедиа в качестве управления состоянием.

**Гипермедиа**: технология обработки, структурирования информации и произвольного доступа к ее элементам с помощью гиперсвязей (Тед Нильсон, 1965), WWW – реализация гипермедиа.

17. Перечислите основные общепринятые правила, применяемые при разработке REST-сервисов.

REST: общепринятые правила

Общий префикс для всех ресурсов сервиса … /API/…,

Два типа ресурсов: коллекция (users, students, …), элемент коллекции /api/users/238, /api/students/ef3d26.

Иерархическая связь …/api/users/238/cars/aah4899

Ограничить количество HTTP-статусов, сопроводить сообщение дополнительным кодом (например 20003, 404001,…),сделать отдельный ресурс (HATEOAS link) для пояснения ошибок http://ccc/api/errors/20003.

Подавление статуса ответа …/api/students/ef3d26?status\_code=200.

Версионность /api/students/ef3d26?v=7.

Постраничное получение данных: параметры limit, offset.

Сортировка: параметр sort.

Все фильтры вынести за знак вопроса: …/api/students?minbday=1998101&maxbday=20001231&gender=m.

Пользователь получает только то, что хочет: …/api/students?field=bday,surname,gender.

Обозначать в запросе формат сообщений (желательна поддержка нескольких форматов):

Глобальный поиск: ../api/search?q=19600107+Иванов.

Документация.

18. Основные недостатки REST-архитектуры.

* Нет общепринятого стандарта RESTful API.
* Не все браузеры поддерживают полный словарь REST-методов (PUT, DELETE); на практике используется только GET и POST (insert, delete, update).
* Не однозначны коды состояний.

19. Что такое RPC?

Remote Procedure Call — вызов удалённых процедур

20. Каким образом представляется RPC-сервис пользователю(разработчику)?

RPC: адрес SOAP-node; имя процедуры или метода; идентификаторы и значения параметров; шаблон передачи сообщений;

21. Перечислите известные вам технологии реализации RPC-сервисов.

CORBA (OMG), COM. DCOM, COM+, RMI, XML-RPC, SOAP, EbXML

22. Что такое SOAP?

**SOAP:** Simple Object Access Protocol – простой протокол доступа к объекту

Транспорт: HTTP, SMTP, FTP

23. Что такое WSDL?

**WSDL: Web Services Description Language**

**WSDL:** разработанаIBM (NASSL), Microsoft (SCL), webMethods (WIDL).

**WSDL:** средство XML-описания интерфейса для доступа к web-службе; платформонезависимое описание;

24. Что такое UDDI?

UDDI (Universal Description Discovery & Integration) стандарт хранения WSDL –файлов для поиска другими организациями для поиска и внедрения в свои организации.

25. Перечислите секции SOAP Envelop-документа.

**SOAP: Header block** (необязательный)

**SOAP: Body block** (обязательный)

**SOAP Fault** (может в Header или Body): env:Сode – код сообщения; env:Reason – пояснение к коду; env:Node – узел назначения; env: Role – роль назначения; env: Detail – детальная информация.

26. Каким образом протокол SOAP может быть расширен?

27. Перечислите секции WSDL-документа.

<!-- структура wsdl-документа -->

<definition>

<types>

<!-- оперделение типов, используемых web-сервисом -->

</types>

<message>

<!-- сообщения, используемые web-сервисом -->

</message>

<portType>

<!-- методы, предоставляемые web-сервисом -->

<!-- определены операции web-сервиса и используемые сообщения -->

</portType>

<binding>

<!-- протоколы связи, используемые web-сервисом -->

<!-- форматы сообщений и детали протокола для каждого порта -->

</binding>

<service>

<!— набор портов связанных с сервисом -->

</service>

</definition>

28. Перечислите основные свойства ASMX-сервиса.

* **ASMX: Active Server Method Extended;** технология Microsoft для разработки web-сервисов(2002, 2007), основанная на XML, WSDL, SOAP; официальное название в MSDN «XML Web Services»; легкая технология: для работы с ней не обязательно знать XML, SOAP и WSDL
* - Не нужно хостить
* - Модель взаимодействия только полудуплекс
* - ASMX - это старый способ делать веб-службы, а WCF - это текущий способ работы с веб-службами.

Сеня пидор ты Смелику сервисы сдавал?Нет ну и иди нахуй.

* asmx-сервис можно размещать только на iis-сервере
* используется только полудуплексная модель взаимодействия
* asmx-сервис работает только по протоколу http в качестве транспорта
* asmx-сервис может использовать только одну конечную точку (сетевой ресурс, которому можно отсылать сообщение)

29. Что такое WCF?

**WCF:** Windows Communication Foundation; технология основанная на .NET FRAMEWORK для разработки приложений SOA-архитектуры, первая версия 2006 (Indigo).

**WCF:** Windows Communication Foundation - технология для построения SOA-приложений.

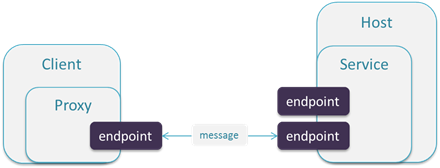
30. Перечислите основные свойства WCF-сервиса.

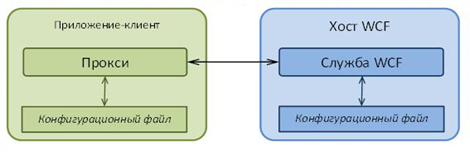
* **В качестве хоста может выступать консольное или графическое Net приложение(селфхостинг)или iis**
* **Для взаимодействия используют полудуплекную, дуплексную, однонаправленную,потоковую, издатель-подписчик модели взаимодействия**
* **В качестве транспорта могут выступать протоколы HTTP, TCP/IP, NAMED PIPE, MSMQ**
* **Количество endpoint равно количеству протоколов используемых сервисом**

**(Сеня насрал ниже)**

**WCF:** WCF-служба, представляет собой класс; этот класс не может существовать самостоятельно, а должен находиться под управлением некоторого процесса Windows, называемого ***хостовым*** процессом. В качестве хоста может выступать консольное или графическое NET-приложение (автохостинг), Windows-служба (Windows Service), IIS, WAS.

**WCF:** архитектура

****

****

**Service:** dll-библиотека.

**Host:** программный модуль, содержащий в себе Service.

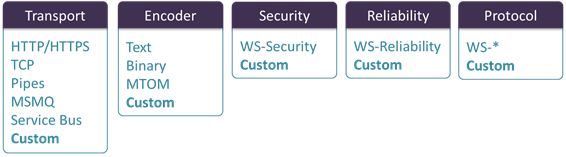
**Client:** приложение, использующее Service.

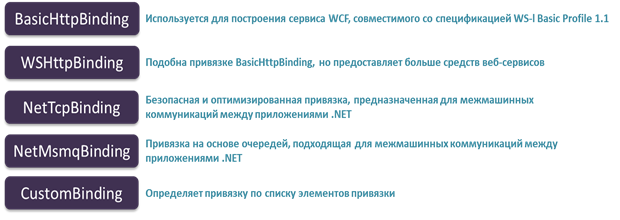
**Proxy:** промежуточная dll-библиотека эмулирующая работу с Service, как с локальным объектом.

**Endpoint:** конечная точка – сетевой ресурс, которому можно отсылать message.

**Address:** сетевой адрес сервиса.

**Binding:** способ взаимодействия клиента с сервисом.





**Contract:** описание интерфейса сервиса (что может сервис).



**WCF:** для передачи параметров имеющих известный тип (Known Types) используется сериализатор DataContractSerializer.

**WCF:** поддерживает и другой тип сериализатора NetDataContractSerializer, который является полной аналогией DataContractSerializer, но содержит полное имя типа (это сводит на нет всю кросс-платформенность).

31. Поясните понятие OData.

**Open Data Protocol –** открытый web-протокол; позволяет выполнять операции с ресурсами и получать ответы в форматах XML, JSON.

32. Что такое WCF Data Service?

WCF Data Services (прежнее название — "службы данных ADO.NET") — это компонент .NET Framework, который позволяет создавать службы, использующие Open Data Protocol (OData) для предоставления и использования данных в Интернете или интрасети с помощью семантики [передачи состояния представления (остальное)](https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm). OData предоставляет доступ к данным в виде ресурсов, которые адресуются по URI. Доступ и изменение данных производится с помощью таких стандартных команд HTTP, как GET, PUT, POST и DELETE. OData использует соглашения об отношениях отношений сущностей [EDM](https://docs.microsoft.com/ru-ru/dotnet/framework/data/adonet/entity-data-model) для предоставления ресурсов в виде наборов сущностей, связанных ассоциациями.

33. Что такое RSS?

**RSS:** семейство XML-форматов, предназначенного для описания новостных лент, анонсов и статей (199) .

34. Что такое ATOM?

**ATOM**: формат описания web-ресурсов и протокола для их публикации (2005г.).

35. Что такое WCF Syndication Service?

Этот шаблон содержит ссылку на библиотеку System.ServiceModel и предусматривает возможность добавления конфигурационного файла приложения для определения конечной точки WCF.

36. Что такое JSON-RPC?

JSON-RPC: протокол удаленного вызова процедур, использующий формат JSON для передачи сообщений.

37. Структура JSON-RPC-запроса.

**jsonrpc** – версия json-rpc протокола.

**method** — строка с именем вызываемого метода.

**params** — массив данных, которые должны быть переданы методу, как параметры.

**id** — значение любого типа, которое используется для установки соответствия между запросом и ответом.

38. Структура JSON-RPC-ответа.

**jsonrpc** – версия json-rpc протокола.

**method** — строка с именем вызываемого метода.

**params** — массив данных, которые должны быть переданы методу, как параметры.

**id** — значение любого типа, которое используется для установки соответствия между запросом и ответом.

39. Структура JSON-RPC пакетного запроса.

[

{

"jsonrpc": "2.0",

"method": "",

"params": {},

"id": 1

}

]

40. Структура JSON-RPC пакетного ответа.

[

{

"jsonrpc": "2.0",

"result": {},

"id": 1

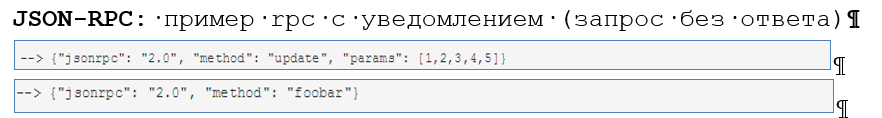
error если есть

}

]

41. Структура JSON-RPC-уведомления.

Отсутствует id



42. Что такое микросервис?

Микросервис – сервис, выполняющий одну элементарную функцию; основной принцип разбиения – изменение сервиса не затрагивает другие сервисы.

43. Что такое DevOps?

**DevOps (DEVelopment** **OPeration)** – это набор практик для повышения эффективности процессов разработки **(Development)** и эксплуатации **(Operation)** программного обеспечения (ПО) за счет их непрерывной интеграции и активного взаимодействия профильных специалистов с помощью инструментов автоматизации.

Девопс позиционируется как [Agile](https://www.bigdataschool.ru/wiki/agile)-подход для устранения организационных и временных барьеров между командами разработчиков и других участников жизненного цикла ПО (тестировщиками, администраторами, техподдержкой), чтобы они могли быстрее и надежнее собирать, тестировать и выпускать релизы программных продуктов

44. Что такое CI/CD?

CI – Continuous Integration (непрерывная интеграция), CD – Continuous Delivery, CD – Continuous Development. TFS – Team Foundation Server – продукт Microsoft для реализации CI/CD.

45. Что такое ASP.NET Nancy?

**Nancy:** Framework для разработки self-hosting приложений в .NET и Core.

46. Укажите место Nancy в структуре OWIN.

Nancy работает в ASP.Net CORE, который придерживается структуре OWIN

47. Что такое DDD?

**1. DDD:** Domain-Driven Design - предметно(проблемно)-ориентированное проектирование: набор принципов направленных на создание систем объектов со сложной предметной областью.

**2. DDD:** цель – ускорить процесс проектирование программного обеспечения.

**3. DDD:** введен Э. Эвансом в книге «Domain-Driven Design», 2004г.

48. Что такое CQRS?

**CQRS**: разделение ответственности на команды и запросы;

CQS – command-query separation(Б. Майер) = принцип императивного программирования (команда-память-команда-память, ассемблер – типичный императивный язык). Основной принцип – команда-чтение не вызывает побочного эффекта(не изменяет состояние объекта).

**CQRS**: контрактное программирование (Б.Майер): спецификация методов (интерфейсы), предусловие, постусловие, инвариант (обязательные соглашения в методе), исключения.

49. Что такое Event Sourcing?

Event Sourcing: подход к хранению данных, при котором вместо конечного результата хранится череда событий произошедших с некоторой сущностью.

**ES:** каждое событие имеет имя.

**ES:** произошедшие события неизменны.

**ES:** обычно применяется в архитектуре CQRS для реализации WriteDB.

**ES:** проекция - вычисленные результаты для UI (аналог view).

**ES:** ориентация на процесс (workflow driven), а не на данные (data-input based).

50. Что такое Event Store?

EventStore: функциональная база данных с комплексной обработкой событий в JavaScript.