Afli

41 уровень

Санкт-Петербург

10.11.2016 17092 7

Уровень 33. Ответы на вопросы к собеседованию по теме уровня

Вопросы/дополнения/критика приветствуются.

1. **Что такое JSON?**

   JSON (JavaScript Object Notation) — простой формат обмена данными, основанный на подмножестве языка программирования JavaScript.

2. **В чем отличия Java и JavaScript?**

   Это 2 разных языка программирования, несмотря на схожесть их названий. Оба они имеют C-подобный синтаксис. Отличия следующие:

   - Java реализует ООП подход, основанный на классах, JavaScript — на прототипах;
   - Java имеет статическую типизацию, JavaScript — динамическую типизацию;
   - Java загружается из скомпилированного байт-кода; JavaScript интерпретируется напрямую из файла.

   Your text to link...

3. **В чем отличия JSON и XML?**

   JSON — формат обмена данными.

Оба они могут быть использованы для передачи данных. Естественно, для работы с обоими стандартами используются различные фреймворки, отличается синтаксис.

4. **Какие фреймворки для работы с JSON вы знаете?**

   33 Уровень познакомил нас с фрейворком jackson. В дополнение я приведу еще 3, и ссылку на статью, в которой они сравниваются:

   0. Jackson от FasterXML
   1. JSON.simple от Yidong Fang
   2. GSON от Google
   3. JSONP от Oracle

   Сравниваем Java-библиотеки для работы с JSON: JSON.simple, GSON, Jackson и JSONP

5. **Какие фреймворки для работы с XML вы знаете?**

   Поскольку XML является форматом представления данных, технологии для работы с ним разнообразнее. Я приведу технологии, используемые для сериализации Java объектов в XML:

   1. JAXB (входит в J в JDK)
   2. Xstream

   Ссылка с кратким обзором различных фреймворков для работы с xml: JAVA + XML

6. **Какие аннотации Jackson вы знаете?**

   Разберем те, которые использовались в лекциях:

   - **@JsonAutoDetect** — ставится перед классом. Сообщает Jackson, что необходимо использовать поля этого класса при записи или чтении. В скобках можно задать параметр (fieldVisibility = JsonAutoDetect.Visibility.ANY), для настройки видимости полей, которые будут использоваться (по умолчанию используются только public поля).
   - **@JsonIgnore** — ставится перед полем. Сообщает Jackson, что данное поле нужно игнорировать при чтении/записи.
   - **@JsonProperty** — Ставится перед полем, getter’ом или setter’ом. Позволяет задать другое имя поля при сериализации.
   - **@JsonWriteNullProperties** — Ставится перед классом. Поля объекта, которые равны null не будет игнорироваться.
   - **@JsonPropertyOrder** — Ставится перед классом. позволяет определить порядок, в котором поля java объекта будут сериализованы в JSON.
   - **@JsonDeserialize** — Ставится перед полем. Позволяет определить класс, в который десериализуется JSON объект. Например из java массивы и списки сериализуются в массивы, и при десериализации можно выбрать, что именно мы хотим получить.

   Вот ссылка на сайт с некоторыми аннотациями: Jackson Annotations

7. **Какие аннотации JAXB вы знаете?**

   Так же разберу только те, которые использовались в лекции:

   - **@XmlRootElement** — Ставится перед классом. Указывает на то, что этот объект может быть, элементом самого верхнего уровня, т.е. все остальные элементы лежат в нем.
   - **@XmlType** — Ставится перед классом. Добавляет в XML-схему дополнительную информацию. Можно указать некоторые атрибуты, например порядок элементов имя и тд.
   - **@XmlElement** — Ставится перед полем. Позволяет задать имя xml-элемента, значение по умолчанию и т.д.

     @XmlAttribute — Ставится перед полем. Поле будет представлено как XML-атрибут.
   - **@XmlElementWrapper** — Ставится перед полем, либо геттером. Позволяет создать обрамляющий тэг для группы элементов.
   - **@XmlJavaTypeAdapter** — Ставится перед классом. В скобках указывается вспомогательный класс-адаптер, необходимый для маршилизации/демаршализации данного класса.
   - **@XmlEnum** — Ставится перед enum. В скобках можно указать тип, в котором будут представлены значения enum.
   - **@XmlEnumValue** — Ставится перед значением enum. Позволяет задать специальное значение для данного значения enum.

https://jaxb.java.net/tutorial/index.html

8. **В чем отличие сериализации и десериализации в JSON?**

   Не понял суть вопроса. Сравнивать 2 взаимообратных процесса смысла не вижу. Возможно имелось ввиду сравнение JSON и XML, на эту тему в следующем вопросе приведена ссылка.

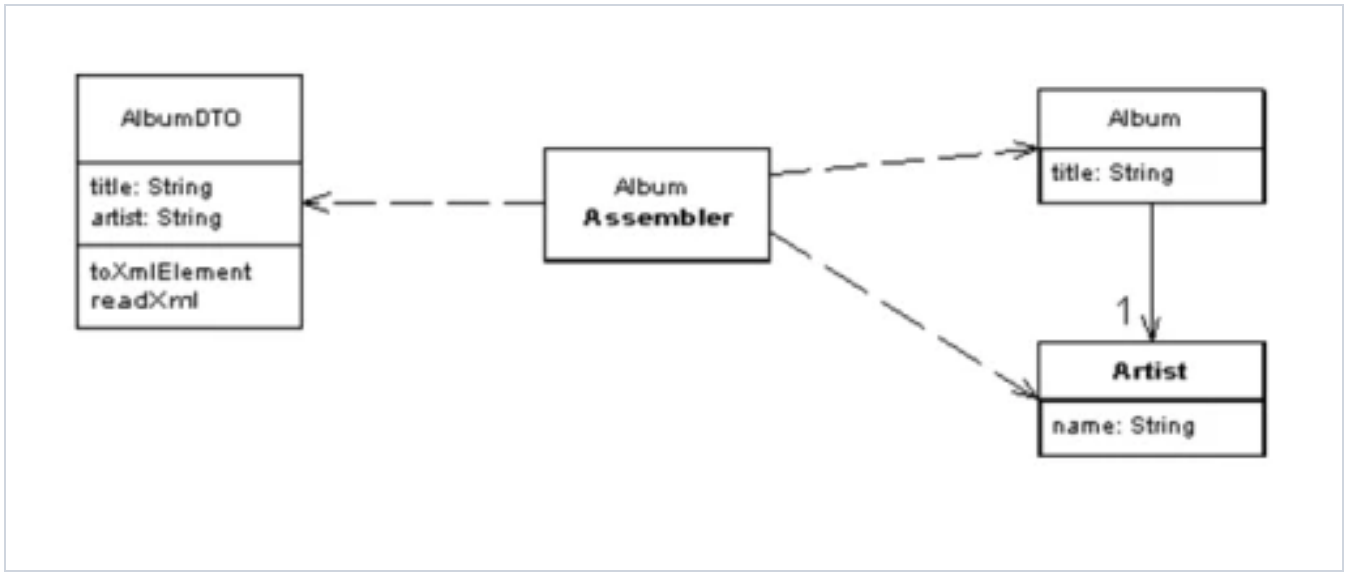
9. **Что лучше JSON или XML? Почему?**

   Вот отличная статья, в которой сравнивается JSON и XML: JSON и XML. Что лучше?

   Пожалуй, нельзя говорить, что что-то лучше. При выборе стоит смотреть на саму задачу и то, что будет эффективнее в использовании. Плюс, выбор может зависеть от личных предпочтений разработчика.

10. **Что такое DTO?**

    DTO (Data Transfer Object) — паттерн проектирования содержащий данные без какой-либо логики для работы с ними. DTO обычно используются для передачи данных между различными приложениями, либо между слоями внутри одного приложения. Их можно рассматривать как хранилище информации, единственная цель которого — передать эту информацию получателю.



Обновлено 10.11.2016 в 15.50

исправлены вопросы №2, №3, согласно замечанию

9. Что лучше JSON или XML? Почему?
Json - для универсальных web-приложений, xml - для корпоративных. Например, системы криптографии заточены именно под XML, напр. чтобы зашифровать отдельные элементы XML и передать на центральный сервер используя WSDL (гос.структуры так обмениваются между собой).

Ответить +4

Роман Кончалов Уровень 28, Россия EXPERT 29 марта, 20:53

Не понятно, кто запрещает шифровать JSON целиком, или отдельные значения.
Разница скорее в более широких инструментах валидации XSD (XML Schema) и DTD, когда у JSON только JSON Schema, которая уступает XSD.

Ответить 0

MrArtemAA Уровень 40, Москва, Россия 7 декабря 2017, 13:55

п.8. Полагаю, подразумевается момент из лекций, который говорит, что местами для сериализации требуется меньше информации, чем для десериализации. Для себя ответил так:

1. Если указан интерфейс, при десереализации требуется указать конкретную реализацию

2. Иерархия классов: если при сериализации в целом ничего критичного может не произойти, десериализация же может пройти не успешно, поэтому требуется указание дополнительного идентификатора и сопоставление типу конкретного класса, который будет использован при сериализации (в JSON появится доп. поле). На него же будет ориентироваться десериализация.

Ответить +8

lichMax Уровень 40, Санкт-Петербург, Россия 4 июля 2017, 21:58

По последнему вопросу немного было сказано на более позднем уровнем тыц, что странно. Для тех, кто ещё не открыл эту лекцию, вот цитата оттуда:

"DTO — Data Transfer Object – объект, который создается с целью быть использованным при транспортировке данных. Обычно к таким объектам два требования: а) уметь хранить данные, б) уметь сериализоваться. Т.е. их используют только для пересылки данных.
Создал объект, записал в него нужные данные из бизнес-логики, сериализовал в JSON/XML и отправил куда-надо. Или наоборот – пришло сообщение – десериализовал его в DTO-объект и вытягивай из него данные."

Ответить +13

Joysi Уровень 41, Россия 10 ноября 2016, 16:29

Так как данная тема весьма важна (особенно для будущего Web/Enterprise трудоустройства) лучше немного больше копнуть для самообразования.
Небольшие дополнения к пунктам:
2) JavaScript уже полноценный язык, который:
— можно транслировать и использовать как язык программирования общего назначения (прочитайте, например, все что связано с Node JS). Java и JavaScript — это просто два разных языка программирования, имеющих общие 4 буквы в начале :) У них есть схожие и различные черты.

3) XML — язык разметки (в котором можно задать синтаксис, структуру, типы данных и вообще их модель — это можно подробнее узнать посмотрев инфу про XSD и его менее популярного брата DTD). Некоторые такие модели XML стандартизованы (например MathML для описания мат формул с соблюдением семантики).
JSON — формат обмена данными, к нему также есть дополнительные средства валидации (например JSON Schema — но они пока не используются достаточно широко).

Вопросы по типу 8 и 9 — они скорее всего для того, чтобы понять наличии практики работы с ними будущего кандидата на трудоустройство.

Ответить +3

Afli Уровень 41, Санкт-Петербург, Россия 10 ноября 2016, 18:56

спасибо за замечания, отредактиовал

Ответить 0

JavaRush — это интерактивный онлайн-курс по изучению Java-программирования с нуля. Он содержит 1200 практических задач с проверкой решения в один клик, необходимый минимум теории по основам Java и мотивирующие фишки, которые помогут пройти курс до конца: игры, опросы, интересные проекты и статьи об эффективном обучении и карьере Java-девелопера.

ПОДПИСЫВАЙТЕСЬ

ЯЗЫК ИНТЕРФЕЙСА

Русский