Springdatajpa

什么是 Java Persistence API?
Java Persistence API 提供了一个规范，用于将数据通过 Java 对象持久化、读取和管理到数据库中的关系表。
什么是 Hibernate 框架?
Hibernate 是 Java 环境的对象关系映射解决方案。对象关系映射或 ORM 是将应用程序域模型对象映射到关系数据库表的编程技术。Hibernate 是一个基于 Java 的 ORM 工具，它提供了一个框架，用于将应用程序域对象映射到关系数据库表，反之亦然。

Hibernate 提供了 Java Persistence API 的参考实现，使其成为具有松散耦合优势的 ORM 工具的绝佳选择。请注意，JPA 是一个规范，Hibernate 是一个 JPA 提供者或实现。
什么是 Spring Data JPA?
Spring Data 是 Spring Framework 的一部分。Spring Data 存储库抽象的目标是显著减少为各种持久性存储实现数据访问层所需的代码量。
Spring Data JPA 不是 JPA 提供者。它是一个库/框架，它在我们的 JPA 提供程序（如 Hibernate）的顶部添加了一个额外的抽象层。

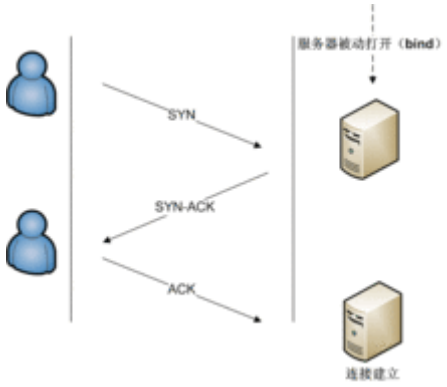
现在，您熟悉 JPA、Hibernate 和 Spring Data JPA 的定义。现在，让我们讨论 Hibernate 和 Spring Data JPA 之间的区别。
Hibernate 和 Spring Data JPA 有什么区别?
Hibernate 是一个 JPA 实现，而 Spring Data JPA 是一个 JPA 数据访问抽象。Spring Data 提供了 GenericDao 自定义实现的解决方案，它还可以通过方法名称约定代表您生成 JPA 查询。通过 Spring Data，您可以使用 Hibernate、Eclipse Link 或任何其他 JPA 提供程序。一个非常有趣的好处是您可以使用@Transactional 注释以声明方式控制事务边界。

Spring Data JPA 不是一个实现或 JPA 提供者，它只是一个抽象，用于显著减少为各种持久性存储实现数据访问层所需的代码量。Hibernate 提供了 Java Persistence API 的参考实现，使其成为具有松散耦合优势的 ORM 工具的绝佳选择。
请记住，Spring Data JPA 始终需要 JPA 提供程序，如 Hibernate 或 Eclipse Link。

TCP 协议

连接建立

TCP 是因特网中的传输层协议，使用三次握手协议建立连接。当主动方发出 SYN 连接请求后，等待对方回答



TCP 的三次握手

SYN+ACK[1]，并最终对对方的 SYN 执行 ACK 确认。这种建立连接的方法可以防止产生错误的连接，TCP 使用的流量控制协议是可变大小的滑动窗口协议。[1]

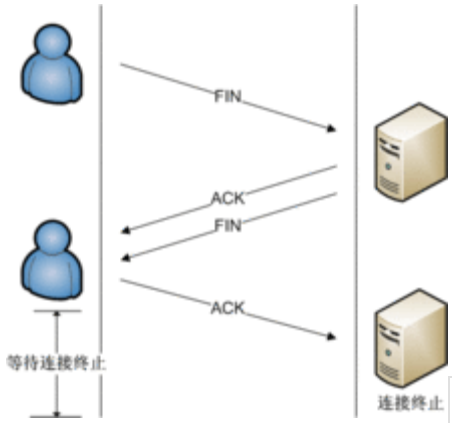
TCP 三次握手的过程如下：

1. 客户端发送 SYN（SEQ=x）报文给服务器端，进入 SYN_SEND 状态。
2. 服务器端收到 SYN 报文，回应一个 SYN （SEQ=y）ACK(ACK=x+1）报文，进入 SYN_RECV 状态。
3. 客户端收到服务器端的 SYN 报文，回应一个 ACK(ACK=y+1）报文，进入 Established 状态。

三次握手完成，TCP 客户端和服务器端成功地建立连接，可以开始传输数据了。

连接终止

建立一个连接需要三次握手，而终止一个连接要经过四次握手，这是由 TCP 的半关闭（half-close）造成的。具体过程如下图所示。[1]



TCP 连接的终止

(1) 某个应用进程首先调用 close，称该端执行“主动关闭”（active close）。该端的 TCP 于是发送一个 FIN 分节，表示数据发送完毕。

(2) 接收到这个 FIN 的对端执行 “被动关闭”（passive close），这个 FIN 由 TCP 确认。

注意：FIN 的接收也作为一个文件结束符（end-of-file）传递给接收端应用进程，放在已排队等候该应用进程接收的任何其他数据之后，因为，FIN 的接收意味着接收端应用进程在相应连接上再无额外数据可接收。

(3) 一段时间后，接收到这个文件结束符的应用进程将调用 close 关闭它的套接字。这导致它的 TCP 也发送一个 FIN。

(4) 接收这个最终 FIN 的原发送端 TCP（即执行主动关闭的那一端）确认这个 FIN。[1]

既然每个方向都需要一个 FIN 和一个 ACK，因此通常需要 4 个分节。

注意：

(1) “通常”是指，某些情况下，步骤 1 的 FIN 随数据一起发送，另外，步骤 2 和步骤 3 发送的分节都出自执行被动关闭那一端，有可能被合并成一个分节。[2]

(2) 在步骤 2 与步骤 3 之间，从执行被动关闭一端到执行主动关闭一端流动数据是可能的，这称为“半关闭”（half-close）。

(3) 当一个 Unix 进程无论自愿地（调用 exit 或从 main 函数返回）还是非自愿地（收到一个终止本进程的信号）终止时，所有打开的描述符都被关闭，这也导致仍然打开的任何 TCP 连接上也发出一个 FIN。

无论是客户还是服务器，任何一端都可以执行主动关闭。通常情况是，客户执行主动关闭，但是某些协议，例如，HTTP/1.0 却由服务器执行主动关闭。

URL 格式

编辑 讨论

在WWW上,每一信息资源都有统一的且在网上唯一的地址,该地址就叫URL(Uniform Resource Locator,统一资源定位符)，它是 WWW 的统一资源定位标志，就是指网络地址。

语法

编辑

URL 由三部分组成：资源类型、存放资源的主机域名、资源文件名。

URL 的一般语法格式为：

(带方括号[]的为可选项):

protocol :// hostname[:port] / path / [;parameters][?query]#fragment

格式说明

编辑

protocol（协议）

指定使用的传输协议，下表列出 protocol 属性的有效方案名称。 最常用的是 HTTP 协议，它也是目前 WWW 中应用最广的协议。

file 资源是本地计算机上的文件。格式 file:///，注意后边应是三个斜杠。

ftp 通过 FTP 访问资源。格式 FTP://

gopher 通过 Gopher 协议访问该资源。

http 通过 HTTP 访问该资源。 格式 HTTP://

https 通过安全的 HTTPS 访问该资源。 格式 HTTPS://

mailto 资源为电子邮件地址，通过 SMTP 访问。 格式 mailto:

MMS 通过 支持 MMS（流媒体）协议的播放该资源。(代表软件：Windows Media Player）格式 MMS://

ed2k 通过 支持 ed2k（专用下载链接）协议的 P2P 软件访问该资源。(代表软件：电驴） 格式 ed2k://

Flashget 通过 支持 Flashget:（专用下载链接）协议的 P2P 软件访问该资源。(代表软件：快车） 格式 Flashget://

thunder 通过 支持 thunder（专用下载链接）协议的 P2P 软件访问该资源。(代表软件：迅雷） 格式 thunder://

news 通过 NNTP 访问该资源。

hostname（主机名）

是指存放资源的服务器的域名系统(DNS) 主机名或 IP 地址。有时，在主机名前也可以包含连接到服务器所需的用户名和密码（格式：username:password@hostname）。

port（端口号）

整数，可选，省略时使用方案的默认端口，各种传输协议都有默认的端口号，如 http 的默认端口为 80。如果输入时省略，则使用默认端口号。有时候出于安全或其他考虑，可以在服务器上对端口进行重定义，即采用非标准端口号，此时，URL 中就不能省略端口号这一项。

path（路径）

由零或多个“/”符号隔开的字符串，一般用来表示主机上的一个目录或文件地址。

parameters（参数）

这是用于指定特殊参数的可选项。

query(查询)

可选，用于给动态网页（如使用 CGI、ISAPI、PHP/JSP/ASP/ASP。NET 等技术制作的网页）传递参数，可有多个参数，用“&”符号隔开，每个参数的名和值用“=”符号隔开。

fragment（信息片断）

字符串，用于指定网络资源中的片断。例如一个网页中有多个名词解释，可使用 fragment 直接定位到某一名词解释。

css 盒子模型

盒子模型是样式表（css）控制页面的很重要的概念。如果理解了盒子模型和其中每个元素的用法，才能熟练使用 css 的定位方法和技巧。所有的页面的元素都可以看成是一个盒子，占据一定的页面空间。占据的空间要比实际使用的空间要大得多。我们可以调整盒子的边框和距离，来调整盒子（页面和页面中的元素）的位置。盒子模型是由内容、边框、间隙（padding）、间隔（margin）组成，他们的关系如下图所示：

图片来自网络

盒子实际宽度（高度）=内容（content）+边框（border）+间隙（padding）+间隔（margin）。对于任何一个元素设置 width 和 height 控制内容大小，也可以分别设置各自的边框（border）、间隙（padding）、间隔（margin）。灵活设置这些盒子的这些属性，可以实现各自排班效果。

border

border 是元素的外围，计算元素的宽和高要把 border 加上特别是特殊网站页面（比如说活动专题页面等）上，这一点是很多新手忽略的地方。border 有三个主要属性，color（颜色）、width（粗细）和 style（样式）。

1、color 主要是指定 border 的颜色，一共有 256 的 3 次方种颜色供我们选择。通常是 16 进制的值，比如红色是“#FF0000”。

2、width 是 border 粗细程度，可以设置为 thin、thick 和 length，length 为具体数值，比如说 border:1px #CCC solid;其中 1px 指的是 border 的 width，默认值是 medium，一般浏览器解析为 2 像素。

3、style 属性可以设为 none、hidden、dotted、dashed、solid、double、groove、ridge、inset 和 outset 等，其中 none 和 hodden 是不显示 border，hidden 可以用来解决边框的冲突问题。对于 groove、inset、outset、rigde、border-style，IE 会出现兼容问题，使用时一定要注意。

padding

padding 用于控制 content 与 border 之间的距离，同时设置上下左右的 padding 时，可以这样写 padding:10px 20px 10px 10px;分别对应上、右、下、左四个方向的 padding，逆时针排序，margin 属性也可以这样书写。

margin

margin 用于控制块与块（可以理解成块级元素）之间的距离。为了便于理解可以把盒子模型想象成一幅画，content 是画本身，padding 是画与画框的留白，border 是画框，margin 是画与画之间距离。需要注意的是 IE 和 firefox 在处理 margin 时有一些差别，倘若设定了父元素的高度 height 值，如果此时子元素的高度超过了父元素的 height 值，二则显示结果完全不同，IE 浏览器会自动扩大，而 firefox（火狐浏览器）就不会，这一点是需要注意的。

关于盒子模型先聊到这，上面总结都是最简单的，一定要熟练掌握。每天学习一个知识点，每日寄语-“不妄求,则心安,不妄做,则身安.”

JSON

JSON(JavaScript Object Notation, JS 对象简谱) 是一种轻量级的数据交换格式。它基于 ECMAScript (欧洲计算机协会制定的 js 规范)的一个子集，采用完全独立于编程语言的文本格式来存储和表示数据。简洁和清晰的层次结构使得 JSON 成为理想的数据交换语言。易于人阅读和编写，同时也易于机器解析和生成，并有效地提升网络传输效率。

简要历史

编辑

JSON(JavaScript Object Notation) 是一种轻量级的数据交换格式。易于人阅读和编写。同时也易于机器解析和生成。它基于 JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999 的一个子集。

JSON 是 Douglas Crockford 在 2001 年开始推广使用的数据格式，在 2005 年-2006 年正式成为主流的数据格式，雅虎和谷歌就在那时候开始广泛地使用 JSON 格式。

JSON 语法

编辑

JSON 语法规则

在 JS 语言中，一切都是对象。因此，任何支持的类型都可以通过 JSON 来表示，例如字符串、数字、对象、数组等。但是对象和数组是比较特殊且常用的两种类型:

- 对象表示为键值对
- 数据由逗号分隔
- 花括号保存对象
- 方括号保存数组

JSON 键/值对

JSON 键值对是用来保存 JS 对象的一种方式，和 JS 对象的写法也大同小异，键/值对组合中的键名写在前面并用双引号 "" 包裹，使用冒号 ：分隔，然后紧接着值:

这很容易理解，等价于这条 JavaScript 语句:

JSON 与 JS 对象的关系

很多人搞不清楚 JSON 和 Js 对象的关系，甚至连谁是谁都不清楚。其实，可以这么理解:

JSON 是 JS 对象的字符串表示法，它使用文本表示一个 JS 对象的信息，本质是一个字符串。

如

JSON 和 JS 对象互转

要实现从 JSON 字符串转换为 JS 对象，使用 JSON.parse() 方法：

要实现从 JS 对象转换为 JSON 字符串，使用 JSON.stringify() 方法：

常用类型

编辑

任何支持的类型都可以通过 JSON 来表示，例如字符串、数字、对象、数组等。但是对象和数组是比较特殊且常用的两种类型。

对象：对象在 JS 中是使用花括号包裹 {} 起来的内容，数据结构为 {key1：value1, key2：value2, ...} 的键值对结构。在面向对象的语言中，key 为对象的属性，value 为对应的值。键名可以使用整数和字符串来表示。值的类型可以是任意类型。

数组：数组在 JS 中是方括号 [] 包裹起来的内容，数据结构为 ["java", "javascript", "vb", ...] 的索引结构。在 JS 中，数组是一种比较特殊的数据类型，它也可以像对象那样使用键值对，但还是索引使用得多。同样，值的类型可以是任意类型。

基础示例

编辑

简单地说[1]，JSON 可以将 JavaScript 对象中表示的一组数据转换为字符串，然后就可以在网络或者程序之间轻松地传递这个字符串，并在需要的时候将它还原为各编程语言所支持的数据格式，例如在 PHP 中，可以将 JSON 还原为数组或者一个基本对象。在用到 AJAX 时，如果需要用到数组传值，这时就需要用 JSON 将数组转化为字符串。

表示对象

JSON 最常用的格式是对象的 键值对。例如下面这样：

表示数组

和普通的 JS 数组一样，JSON 表示数组的方式也是使用方括号 []。

这不难理解。在这个示例中，只有一个名为 people 的变量，值是包含两个条目的数组，每个条目是一个人的记录，其中包含名和姓。上面的示例演示如何用括号将记录组合成一个值。当然，可以使用相同的语法表示更过多的值（每个值包含多个记录）。

在处理 JSON 格式的数据时，没有需要遵守的预定义的约束。所以，在同样的数据结构中，可以改变表示数据的方式，也可以使用不同方式表示同一事物。

如前面所说，除了对象和数组，你也可以简单地使用字符串或者数字等来存储简单的数据，但这样并没有多大意义。

和 XML 的比较

编辑

可读性

JSON 和 XML 的可读性可谓不相上下，一边是简易的语法，一边是规范的标签形式，很难分出胜负。

可扩展性

XML 天生有很好的扩展性，JSON 当然也有，没有什么是 XML 可以扩展而 JSON 却不能扩展的。不过 JSON 在 Javascript 主场作战，可以存储 Javascript 复合对象，有着 xml 不可比拟的优势。

编码难度

XML 有丰富的编码工具，比如 Dom4j、Dom、SAX 等，JSON 也有提供的工具。无工具的情况下，相信熟练的开发人员一样能很快的写出想要的 xml 文档和 JSON 字符串，不过，xml 文档要多很多结构上的字符。

解码难度

XML 的解析方式有两种:

一是通过文档模型解析，也就是通过父标签索引出一组标记。例如：xmlData.getElementsByTagName("tagName")，但是这样是要在预先知道文档结构的情况下使用，无法进行通用的封装。

另外一种方法是遍历节点（document 以及 childNodes）。这个可以通过递归来实现，不过解析出来的数据仍旧是形式各异，往往也不能满足预先的要求。

凡是这样可扩展的结构数据解析起来一定都很困难。

JSON 也同样如此。如果预先知道 JSON 结构的情况下，使用 JSON 进行数据传递简直是太美妙了，可以写出很实用美观可读性强的代码。如果你是纯粹的前台开发人员，一定会非常喜欢 JSON。但是如果你是一个应用开发人员，就不是那么喜欢了，毕竟 xml 才是真正的结构化标记语言，用于进行数据传递。

而如果不知道 JSON 的结构而去解析 JSON 的话，那简直是噩梦。费时费力不说，代码也会变得冗余拖沓，得到的结果也不尽人意。但是这样也不影响众多前台开发人员选择 JSON。因为 json.js 中的 toJSONString()就可以看到 JSON 的字符串结构。当然对于不是经常使用这个字符串的人，这样做仍旧是噩梦。常用 JSON 的人看到这个字符串之后，就对 JSON 的结构很明了了，就更容易的操作 JSON。

以上是在 Javascript 中仅对于数据传递的 xml 与 JSON 的解析。在 Javascript 地盘内，JSON 毕竟是主场作战，其优势当然要远远优越于 xml。如果 JSON 中存储 Javascript 复合对象，而且不知道其结构的话，我相信很多程序员也一样是哭着解析 JSON 的。

除了上述之外，JSON 和 XML 还有另外一个很大的区别在于有效数据率。JSON 作为数据包格式传输的时候具有更高的效率，这是因为 JSON 不像 XML 那样需要有严格的闭合标签，这就让有效数据量与总数据包比大大提升，从而减少同等数据流量的情况下，网络的传输压力[2]。

校验工具

编辑

前言

JSON 格式取代了 xml 给网络传输带来了很大的便利,但是却没有了 xml 的一目了然,尤其是 json 数据很长的时候,我们会陷入繁琐复杂的数据节点查找中。

但是国人的一款在线工具 BeJson 、SoJson[3] 在线工具让众多程序员、新接触 JSON 格式的程序员更快的了解 JSON 的结构，更快的精确定位 JSON 格式错误。

功能

1 JSON 格式化校验

很多人在得到 JSON 数据后,一时没有办法判断 JSON 数据格式是否正确,是否少或多符号而导致程序不能解析,这个功能正好能帮助大家来完成 JSON 格式的校验。

2 JSON 视图

想必很多程序员都会遇到当找一个节点的时候,会发现如果直接对着一行行数据无从下手,就算知道哪个位置,还要一个节点一个节点的往下找,万一一不留神又得从头开始找的麻烦事。

有了这个功能,一切 JSON 数据都会变成视图格式,一目了然,什么对象下有多少数组,一个数组下有多少对象。

这个功能非常实用。不光有视图功能还有格式化、压缩、转义、校验功能。总之很强大。

3 压缩转义

程序员在写 JSON 语句测试用例的时候,很多时候为了方便直接写了个 JSON 字符串做测试,但是又陷入了无止境的双引号转义的麻烦中。这款功能集压缩、转义于一身,让你在写测试用例的时候,如鱼得水。

4 JSON 在线编辑器

如果你现在的电脑刚巧没有装你所熟悉的编辑器,如果你想针对拿到的 JSON 数据的某个节点做数据修改时,这个功能可以满足你的需求。

5 在线发送 JSON 数据

大家都知道,JSON 用的最多的还是 web 项目的开发,那你要测试一个接口是否能准确的接受 JSON 数据,那你就得写一个页面发送 JSON 字符串,重复的做着这件事。随着这个功能的横空出世,你可以摆脱写测试页面了,因为这个功能可以将指定的 JSON 数据发送指定的 url,方便吧。

6 JSON 着色

很多人在写文档时，总希望文档能一目了然,但是面对着白底黑字的 JSON 数据总是提不起精神没关系,使用这个功能,所有的关键字都[4] 会被着色,数据结构一目了然。

7 JSON-XML 互转

顾名思义，将 JSON 格式的数据转化成 XML[4] 格式、或者 XML 格式的数据转化成

JSON 格式，一切都不是问题。

8 JSON-VIEW

JSON 查看实用工具，在开发过程中(windows 平台中)可以对 JSON 数据进行格式化和视图显示。

9 它和 xml 一样都是一种数据交换格式

JSON 的最小化

Eclipse RAP 的一个提交者也是领导者的 Ralf Sternberg，他只用了十个类就把快速的和轻量级的库整合到了一起。显然，使用精益和解析法真正地改善了服务器的性能，因为服务器进程以更高的效率为大量的客户创建了 JSON 信息。在外部 JSON 中不存在依赖关系，代码很容易管理，而且也不会占用很多内存。对于你的全部 JSON 项目来说，这还远远不够，但这确实带来了几件好事。