开发人员专业技术丛书

HZ BOOKS

Dojo

Using the Dojo Javascript Library to Build Ajax Applications

Dojo构建Ajax应用程序

（美）James E. Harmon 著

张 龙 译

机械工业出版社
China Machine Press

开发人员专业技术丛书

Dojo

Using the Dojo Javascript Library to Build Ajax Applications

Dojo构建Ajax应用程序

（美）James E. Harmon 著

张 龙 译

机械工业出版社
China Machine Press

本书系统论述了利用 Dojo 构建 Ajax 应用程序的方法和实践。全书内容主要由三部分组成：Dojo 教程、Dojo Widget、Dojo 详解。其中各部分自成一个知识模块，相互之间又恰当衔接。

本书编写体系完整，撰写风格生动、配合大量的代码示例和操作步骤，非常适合 Dojo 开发者学习参考。

本书版权登记号：图字：01-2009-1346

图书在版编目（CIP）数据

Dojo 构建 Ajax 应用程序/(美)哈蒙(Harmon, J. E.)著；张龙译 . —北京：机械工业出版社，2009. 5

(开发人员专业技术丛书)

书名原文：Dojo：Using the Dojo JavaScript Library to Build Ajax Applications

ISBN 978-7-111-26664-8

Ⅰ. D…　Ⅱ. ①哈…　②张…　Ⅲ. 计算机网络 - 程序设计　Ⅳ. TP393. 09

中国版本图书馆 CIP 数据核字(2009)第 043664 号

机械工业出版社(北京市西城区百万庄大街 22 号　邮政编码　100037)

责任编辑：盛东亮

北京京北印刷有限公司印刷

2009 年 5 月第 1 版第 1 次印刷

186mm × 240mm · 15. 5 印张

标准书号：ISBN 978-7-111-26664-8

定价：45. 00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

本社购书热线：(010)68326294

译 者 序

Dojo、ExtJS、DWR、YUI、GWT、Prototype、ZK……面对这么多的 Ajax 框架(或库),初学者难免不知所措,我们到底该学习哪个?我想说的是,首先学习 JavaScript,因为不管哪个 Ajax 框架,其根基始终离不开 JavaScript,只有将 JavaScript 掌握好了才可能走得更远,更快。那么,学习 JavaScript 哪些内容呢?面向对象、闭包、事件模型、DOM 等都是我们必须掌握的,只有这样才不会被一个又一个新的 Ajax 框架所累,才能真正凌驾于框架之上达到灵活运用的目的。

我从 2006 年就已经开始使用 Dojo,那时的版本为 0.43(这也是 Dojo 的一个经典版本),当时关于 Dojo 的资料非常少,甚至连官方网站的帮助文档也不够完善,迫不得已,只有阅读 Dojo 的源代码。从 Dojo 1.0 开始,Dojo 的核心已经发生了重大的变化并且与之前的版本不再兼容;但万变不离其宗,其架构、思想还是有相通之处的。本书讲述的 Dojo 版本为 1.1,我相信只要读者掌握好了该版本,就可以轻松应对以后的版本升级了。

本书共分为三大部分,共十七章。其中第一部分提供了 Dojo 的初学者指南,帮助大家尽快走进 Dojo 的世界。第二部分重点介绍了 Dojo Widget,这也是 Dojo 最重要的组成部分,期间以表格的形式给出了各 Widget 的用法示例及相关属性、方法与事件等,读者也可以将这部分内容当作参考手册,随用随查。第三部分深入讲解了 Dojo 的方方面面,从对象与类到测试与调试,通过这一部分的学习将极大提高读者应用 Dojo 和 JavaScript 的技能。

从接手本书的翻译到交付出版经过了 4 个多月的时间,在这期间我集中了所有精力完成这本专著的翻译,不敢懈怠。

感谢我的父母!感谢你们的培养和教育。感谢我的女友张明辉!在翻译此书的这段日子里,是你无微不至的关怀让我忘却了生活中的琐事,专心于译作。

不得不提的还有华章公司的陈冀康先生,是你的不断鼓励与帮助,给了我完成此书翻译的信心和勇气,谢谢。

限于译者水平,书中难免有不妥之处,如读者在阅读本书的过程中发现了任何问题或是有任何建议,望与我联系。联系方式如下:

E-mail:zhanglong217@yahoo.com.cn

博 客:http://blog.csdn.net/ricohzhanglong

序

学习 Dojo Toolkit 时至少要清楚一点——你想要的是什么！在刚开始使用 Dojo 时，我的基本目标就是创建一个有用的 JavaScript 工具集而不必让专业的 JavaScript 开发者重复发明轮子。术语 Ajax 的出现掀起了一阵旋风，我们很快就发现自己创建的这个工具集被数以千计的开发者和上百万的用户所使用。

就像那些发展速度远远超出想象的项目和公司一样，痛苦始终是伴我们左右的。Dojo 团队花费了将近 18 个月的时间来解决其快速发展所带来的问题：性能、范围、易用性及文档。从市场和文档的角度来看，开源项目的名声并不好，最开始 Dojo 也无法摆脱这个宿命。在发布了 Dojo 0.9、1.0 及 1.1 之后，我们对文档和 API 查看工具进行了极大的改进，现在我们可以自豪地说这些内容已经不会再给 Dojo 抹黑了。

除了源代码文档外，精彩的示例也是一本好书不可或缺的组成部分。在学习新东西时，最困难之处在于你不知道如何提出问题。Dojo 是非常强大和高效的，但这也会导致一些 Dojo 新手无从下手。无论是狭义的 Dojo 还是广义的 Ajax 都会有一些学习曲线，你需要具有宽广的技术背景——从服务器端的编程语言到 JavaScript、CSS、HTML 及 DOM 都需要掌握，外加上浏览器之间的一些不兼容特性。类似于 Dojo 这样的工具集会尽最大努力帮助开发者摆脱这么多问题的困扰，但当开发者面对新问题时还是不可避免地会遇到很多麻烦。

Dojo 开发者和用户可以采取很多方式去解决问题并加快脚步，可以阅读本书，也可以求助于社区，甚至还可以寻求 SitePen 这样的商业公司的帮助。

Dojo 已经获得了成功并在茁壮成长，这是由其透明和开放的进程所决定的。所有代码都基于 AFL 和 BSD 协议，这两个协议的目的在于让广大用户能更好地使用而不是限制其使用。

很多个人与公司都对 Dojo 做出了巨大的贡献，如 AOL、Google、IBM、Nexaweb、Renkoo、SitePen、Sun 及 WaveMaker 等。我们有一个严格但低门槛的贡献政策(contribution policy)，它要求所有贡献的源代码都要通过一个贡献协议协定(Contributor License Agreement)，这样就能保证现在和将来对 Dojo 地使用不会导致法律或专利上的问题。

相对于其他工具集，我们进行了更多的创新和试验，在 DojoX 中引入的新特性已经遥遥领先于其他工具集。

我与 James Harmon 相识在一次会议上，那时他正在进行关于 Dojo 的讲座。James 对 Dojo 深入浅出地介绍非常精彩。Alex Russell 和我总是想面面俱到地将 Dojo 介绍给别人，而 James 却能将复杂的主题分解为易于理解的概念，这样人们就能快速掌握 Dojo 了。

本书同样以清晰明了的方式介绍了如何使用 Dojo 轻松创建 Web 应用和 Web 站点，即使开发者不是 JavaScript 专家也没有关系，凭借 Dojo Toolkit，他们可以快速且高效地进行开发。

Dylan Schiemann
CEO，SitePen
联合创始人，Dojo Toolkit

目　录

第三部分 Dojo 详解

第一部分　Dojo 教程

第 1 章　理解 Dojo：教程

耳听为虚，眼见为实。

——中国谚语

灵感来源于本章开头所引用的谚语，我相信讲解新技术最好的方式之一就是给出一个简单的示例。因此我将以一个教程作为本书的开始，该教程将使用 Dojo Toolkit 来改进一个基本的 HTML 表单。从本章开始一直到第 5 章都将介绍该教程，同时这五章也构成了本书的第一部分——Dojo教程。

1.1　教程简介

假如你是一个 Web 开发者（如果你正在阅读这本书，那么对此就没什么好奇怪的了），有人建议你向站点中增加一些 Ajax 特性。也许提出该需求的人正是你的老板，甚至是老板的老板，但他可能连 Ajax 是什么都不知道，更不用说哪些特性有用了。也许你有点不太自信。想象一下，你之前的经验几乎都在服务器端，使用 Java 或者其他服务器端技术，你对 HTML 和 JavaScript 的经验非常有限。这就是接下来几章中你开始了解 Dojo Toolkit 时所处的场景。

为了进一步说明该场景，假设你已经听说过 JavaScript 编程语言非常强大，有很多 JavaScript 库和框架可以协助你充分利用其强大的功能。你已经决定使用 Dojo Toolkit，因为你经常访问的一些 Web 站点和论坛推荐它。同时你也准备将应用中访问最频繁的一个页面“Ajax 化”。

该教程将通过几个步骤向你介绍如何使用 Ajax 特性来更新页面。我们将采用小步前进的方式来改进该页面，每一小步都会重点说明一个具体的问题。通过这种方式，我们将会看到 Ajax 允许我们往 Web 页面中增加的各种特性，同时我们也会看到如何使用 Dojo Toolkit 来实现这些特性。

1.1.1 本教程的目标

本教程的首要目标就是向你展示如何使用 Dojo Toolkit 将一些常见的 Ajax 特性增加到 Web 页面中。该教程从简单处着手。换句话说，它专注于容易实现且对可用性提升大有裨益的一些特性。它不会对 Dojo 特性进行面面俱到的讲解，也不会涵盖我们实现的所有特性。你可以将它当作改进 Web 站点的第一步。

该教程的另一个主要目标就是以最简单的方式实现一些特性。尽管 Dojo 的大多数特性既可以声明方式（通过 HTML 标记），也可以编程方式（通过 JavaScript）实现，但我们首先以声明方式实现，因为大多数的服务器端 Web 开发者更熟悉 HTML 标记，相比之下，对 JavaScript 就差一些了。当然了，我们也会使用一些 JavaScript 来作为各种东西的粘合剂。

1.1.2 使用 Dojo 的目的

我们使用 Dojo 想要得到什么呢？首要的是——想让页面的可用性更好。我们在很多地方都会这样想。页面的执行应该更快点。看起来更漂亮点。操作起来更简单点。可以帮助用户准确地获取其想要的信息，这就要求页面的导航更简单。但同时在用户访问 Web 页面时，我们不应改变其操作习惯。我们在不断改进可用性的同时，不能以牺牲用户已有地体验为代价。

我们该如何改进可用性呢？Dojo 对已有的 HTML 表单元素进行了改进，提供了额外的功能，这样会使当前的表单元素可用性更好。

我们可以通过如下两种方式来改进性能：让系统运行地更快或者是让系统看起来运行地更快。让一个过程看起来运行地更快的理想方式就是在其运行的同时，用户可以做一些其他事情而不是在那干等着。Ajax 使用了非常棒的机制来支持该技术。我们将使用 Dojo 让页面向服务器端发送异步请求，同时用户的操作可以继续而不被中断。对用户来说，该页面看起来运行地更快，响应也更快了。

我们可以在用户输入数据时就进行验证，以此来改进数据验证过程。Dojo 可以将小部分验证请求发送到服务器端而无需提交整个表单。如果合适的话，我们甚至想采用桌面应用那种数据验证方式——根据用户的按键进行验证。

我们还希望这些特性容易实现。我们想利用对于 HTML 的开发经验，在使用 JavaScript 时，让编程模型保持一致而强大。我们希望相比于完全由自己来开发功能，所需编写的代码量更少。代码少意味着出错的机会小。当你学习 Dojo 时，你会发现你所掌握的内容对进一步探索 Dojo 会大有裨益。当事情变得复杂时，你可以使用工具以辅助调试。简而言之，Dojo

提供了非常棒的编程环境，绝对值得你期待。

最后，你可以看到我们会一直惊讶于 Dojo 带给我们的好处，而这些好处不需要任何额外的工作就可以得到。例如，我们会看到增加的任何特性都可以在各种浏览器中以一致的方式呈现出来。同时我们还会看到用来支持 Web 的易用性（Accessibility）及国际化（Internationalization）标准的一些可视化元素。

我们已经给 Dojo 带上了高帽。我们要求不少，但却不想费太多劲。Dojo 真的行吗？这需要我们去探索。我们首先以一个页面作为改进的基础，通过该页面确定好我们希望解决的各种问题。

1.2　一个标准的 HTML 数据输入表单

首先，从我们的应用中选出一个页面作为 Dojo 改进的目标（图 1-1）。该页面来自于一个假想的客户服务应用，该应用面向全国的电报公司，它允许客户创建账号并请求服务。该教程对于我们“业务”操作的定义很含糊，因为你可能会猜到，该表单就是用来突出很多业务应用都会拥有的一些特定类型的功能。因此如果你能先将怀疑放在一边的话，那就让我们来检查一下该表单。

该页面的设计很一般——几乎一点设计都没有。它仅使用了一点样式，就这点样式也是很简单的。你的页面肯定比它强多了，但我们使用该简单设计以让示例保持足够简单。

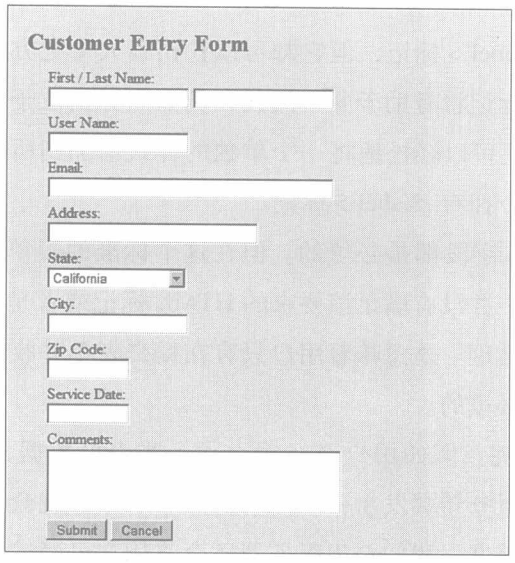
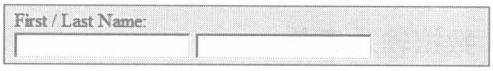


图 1-1　标准的 HTML 客户输入表单

让我们将表单中的每个字段都过一遍，然后看看有没有可用性问题。接下来会谈到 Dojo 是如何解决这些问题的。

1.2.1 First 与 Last Name

第一个数据输入字段用来保存客户的 first name。这足够直接，但我们已经有问题了。该字段的标签内容是："First / Last Name:"，同时紧跟两个文本输入框。尽管用户可能明白该页面需要什么，但对于屏幕阅读器（有视觉障碍的人用其读取屏幕上的内容）来说还是困难些。



你可能会从可用性角度提出质疑：这让人感到不明就理。页面上的其他标签都仅指一个单独的文本框，而该标签却指两个。当将一个名字分为两部分时，last name 就应该放在 first name 前吗？没有其他方式吗？问的好，但我们现在暂且不进行回答。记住，我们现在仅仅是找出问题。在后面的章节中将给出解决方案。

现在让我们看看这些字段的 HTML 标记。

```
<label for="firstName">First / Last Name:</label>
<input type="text" id="firstName" name="firstName" />
<input type="text" id="lastName" name="lastName" />
```

你以前可能没用过 <label> 标记，但它却可以让伤残人士更方便地访问你的站点。当标签不在输入字段前时，该标记将有助于屏幕阅读，例如当标签位于不同的单元格中时。当使用级联样式表（CSS）时还可以轻松地将一个单独的样式增加到所有的标签上。另一个问题就是这两个字段中只有一个拥有 <label> 标记。

First name 和 last name 字段都是必填的。但在这个标准的表单中并没有使用 JavaScript，那我们如何做到这一点呢？并没有满足该要求的 HTML 标记或者是属性存在，因此我们准备将这个验证交给服务器端处理。这意味着用户只有在提交表单并收到来自服务器端的错误消息后才能知道哪些字段是必填的。

该如何显示错误消息呢？假如用户已经输入了一些表单数据，然后按下了"提交"按钮。接下来浏览器就会向服务器端发出一个请求，然后服务器就会对数据进行验证并将带有错误消息的表单返回给浏览器。我们希望服务器还会将用户已经输入的数据也返回来，这样用户就无需重新输入了。错误页面通常在页面上方显示所有错误消息。带有错误消息的页面

看起来如图 1-2 所示。

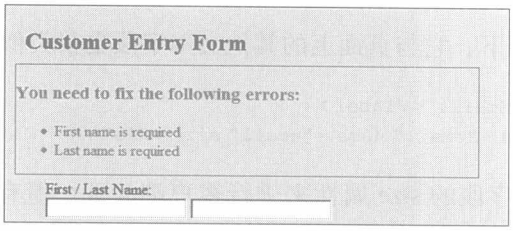


图 1-2　典型的表单错误消息

1.2.2　用户名

我们的应用允许用户登录并管理自己的账号，因此我们要求每个用户都要创建一个用户名。我们应该为用户提供一些指导以帮助其创建合适的名字，但这需要往页面中增加大量的文本，因此我们决定不这样做了。该表单就简简单单地要求一个用户名并提供一个文本输入框而已。

完成该功能的 HTML 标记非常类似于“First / Last Name:”字段，就一个 < label > 和 < input type = " text" > 标记，如下所示。

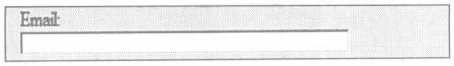

```
<label for="userName">User Name: </label>
<input type="text" id="userName" name="userName" size="20" />
```

我们通过设定 size = " 20" 属性已经增加了一点儿客户端验证，以保证用户无法输入超过 20 个字符的名字。

该字段的一个问题也出现在验证上。用户可能想创建一个易记的简短用户名，但这也是其他用户的想法，这样该用户名就很有可能已经被使用了。那么我们该如何通知用户呢？请注意，直到用户提交页面时才能进行验证。服务器会检查用户名以确认是否已被使用了，如果被使用了，就会返回一个页面来重新显示表单并给出错误消息（同时还有其他字段的错误消息）。如果能给用户一些建议就更好了，这样他就不会再次输入已被使用了的用户名了。该建议应该基于用户想要注册的那个用户名。

1.2.3　Email 地址

我们想通过 Email 与用户交流，因此需要一个 Email 地址。可以使用一个简单的文本输入框来存放用户输入的 Email 地址。



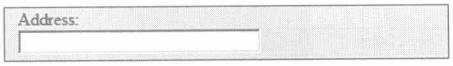
其 HTML 标记显示如下，它与页面上的其他文本字段也很类似。

```
<label for="email">Email: </label>
<input type="text" id="email" name="email" size="45" />
```

我们又一次通过设定字段的 size 属性来进行客户端验证。但有没有办法验证 Email 地址的有效性呢？我们可以使用两类验证。首先，Email 地址的格式正确么？例如，是否包含“@”符号？是否以 TLD 如“.com”结尾？其次，它是一个真实的 Email 地址吗？遗憾的是，如果不实际创建并发送一个 Email 是没法验证后者的。尽管向用户发送一个密码并让他自己来验证用户名是一个好主意，但这已经超出了我们讨论的范围了。因此我们仅仅确认该 Email 地址的格式正确就行了。

1.2.4　地址

我们需要用户家庭地址的首行，同时使用一个文本框保存它。



其 HTML 与上面的字段都很类似。

```
<label for="address">Address: </label>
<input type="text" id="address" name="address"/>
```

该字段应该包含客户账单地址的首行，因此我们需要确保用户填写了该项。它是一个必填字段，但我们还是通过服务器端进行验证。

1.2.5　州

我们将用户所在的州作为其账单地址的一部分。因为州名有限，所以可以使用 <SELECT> 来提供州名的一个下拉列表，用户可以选择其中一个。下图显示了使用下拉列表的州名示例。

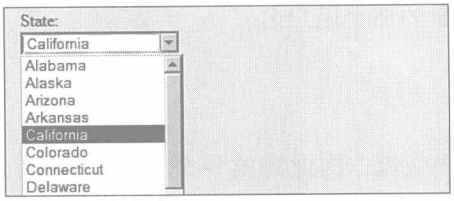


图 1-3　州名的下拉列表

HTML 提供了 < SELECT > 表单元素，它能提供一系列值的列表。下面展示了创建该字段所需的一些标记片段。

```
<select name = "state" >
        <option value = "AL" > Alabama < /option >
        <option value = "AK" > Alaska < /option >
        <option value = "AS" > American Samoa < /option >
        <option value = "AZ" > Arizona < /option >
       ... additional state values not shown ...
< /select >
```

由于州名有限，所以它们的值都可以列出来。对于这个字段，验证不是问题，问题在于行为。我住在伊利诺斯州（Illinois），经常网购，所以我常常要输入账单地址。当从表单中选择州名时，我首先会敲一个“i”，接下来“Idaho”就会跳出来，因为它是第一个以“i”开头的州名。这太棒了——尽管我不住在爱达荷州（Idaho）。接下来我敲一个“l”（小写的“L”），“Louisiana”又会跳出来。现在很多有成就的人都住在路易斯安那州（Louisiana），但我不是。问题在于 < SELECT > 标记将我的输入解释为两种不同的情况——每种都是一个单词的首字母，它没有将其当作一个单词的前两个字母。当我敲“il”时，我想看到所有以“il”开头的州名，只有 Illinois 符合这种情况。遗憾的是，这不是 < SELECT > 标记的工作方式——当我敲“l”时，它显示的是“Louisiana”，这是因为它觉得我又一次敲了一个州名的首字母。

该问题并不是无法解决。有些浏览器就会按照我们期望的方式工作（将整个“il”当作州名的前两个字母），但我们想让页面在各种浏览器中都有一致的行为。

1.2.6　城市

这是需要的另一个字段。我们使用一个文本框存储用户的输入值。

其 HTML 与其他的文本字段一样。

```
<label for = "city" > City: < /label >
<input id = "city" name = "city"/ >
```

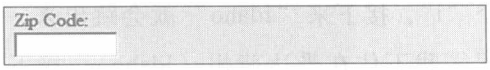
基本的 HTML 表单不会对该字段进行任何验证。但我们能否向用户提供一个城市的下拉列表呢，就像选择州名那样？每个州只有确定数量的城市，但其数量倒不少。整个美国大约有 30 000 多个城市。因此简单的在页面中列出这些值会增加页面的大小，使其加载速

度变慢。另外列出所有城市也不太对头；我们需要列出用户所选择的州下的城市。这就需要我们使用一些 JavaScript 逻辑来进行处理，但在这个简单的表单中我们尽量避免使用 JavaScript。

下拉列表的可用性也是一个问题。由于城市数量太多，导致很多城市都以相同的字母开头。敲入城市的首字母只会让用户看到一个长长的列表的开头。用户不得不向下拉动以寻找正确的值——这也太无聊了吧。

1.2.7　邮政编码

邮政编码是账单地址所需的最后一个字段。我们使用一个文本框来存储用户的输入。



其 HTML 与其他文本框的一样。

```
<label for="zipCode">Zip Code:</label>
<input type="text" id="zipCode" name="zipCode" size="10" /></br>
```

这需要验证。我们还是使用服务器端验证来保证该字段已被输入。服务器将返回一个包含表单、用户已输入的数据及所有验证错误消息的页面。该字段除了必填以外还需要执行什么验证呢？就像 Email 地址一样，它需要执行两类验证，数据格式是否正确和数据是否有效。

在美国邮政编码有两种形式。既可以是5 位数字，又可以是5 位数字后加一个破折号和4 位数字。这意味着输入的数据既可以是 5 个字符长，也可以是 10 个字符长。HTML 确实可以通过 size 属性来限制最大长度。然而它却无法限制最小长度。同时也没法限定是否需要使用一个破折号将邮编的两部分数字分开。这些检查服务器端都可以做，但只有在用户提交表单后才行。

还可以再进一步。就像州和城市一样，美国的邮编数量也是确定的。那我们能否使用 <SELECT> 将其列举出来呢？既然我们已经知道了州和城市的用法，可否将其应用于邮编呢？事实上这个逻辑比你想象的要复杂多了——一些城市有多个邮编，而一些邮编也可能被多个城市所用。当我们将地理范围扩大到美国以外，你会发现情况还要更复杂。尽管如此，为了保持该教程的简单性，我们还是仅讨论美国的情况吧。

关于该字段我们已经谈到了很多问题。记住，在教程的后面会给出解决方案的。

1.2.8　服务日期

我们的客户还想确定其电报服务的开始时间，因此我们提供了一个文本框用来存放其输

入的开始服务日期。

其 HTML 与其他字段的一样。

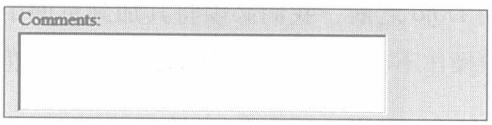
```
<label for="serviceDate">Service Date:</label>
<input type="text" id="serviceDate" name="serviceDate" size="10"/>
```

对于该服务日期需要什么验证呢？当然，它必须是一个有效的日期，但格式呢？我们不会给用户任何提示。这显然是个问题。除此以外，日期应该是尚未到来的，而不是过去的。甚至还有一些日期应该禁止用户输入，比如非工作日。

该字段的另一个可用性问题就是用户不太可能轻松算出未来的日期。2 周后是几号？难道我们向当前日期加 14 就行了吗？如果那天已经超出了当前的月份就不对了。3 周后的第一个星期一是几号？如果没有日历，用户很难计算出想要的日期。

1.2.9　评论

我们来看看表单中最后一个字段——评论。用户可以随意输入一些评论来谈论他对我们服务的看法以及他喜欢的东西——或者是任何他想说的东西。



这是一个多行文本框，用户可以随心所欲地进行输入。其 HTML 如下所示。

```
<label for="comments">Comments:</label>
<textarea name="comments" rows="3" cols="35" id="comments">
</textarea>
```

这不是一个必填字段，所以无需进行验证。HTML 表单元素 <textarea> 提供了一些基本的文字编辑功能。当用户输入到每行末尾时它会自动对单词进行续排。当用户输入的文本超出了文本域的表示范围，就会自动在右边出现一个滚动条，用户可以拖拽它进行滚动。这些是 <textarea> 的特性，但它缺少格式化功能。

至此我们完成了对该原始 HTML 表单的介绍。既然我们已经将该表单的诸多问题进行了分类，那我们就按照计划使用 Dojo 来逐一解决它们。