

架，多数功能只是对 HTTP 协议中常用操作的封装，更多的功能需要插件或者整合其他模块来完成。

下面用 Express 重新实现前面的例子：

```
var express = require('express');

var app = express.createServer();
app.use(express.bodyParser());
app.all('/', function(req, res) {
  res.send(req.body.title + req.body.text);
});

app.listen(3000);
```

可以看到，我们不需要手动编写 req 的事件监听器了，只需加载 `express.bodyParser()` 就能直接通过 `req.body` 获取 POST 的数据了。

5.2 快速开始

在上一小节我们已经介绍了 Web 开发的典型架构，我们选择了用 Express 作为开发框架来开发一个网站，从现在开始我们就要真正动手实践了。

5.2.1 安装 Express

首先我们要安装 Express。如果一个包是某个工程依赖，那么我们需要在工程的目录下使用本地模式安装这个包，如果要通过命令行调用这个包中的命令，则需要用全局模式安装（关于本地模式和全局模式，参见 3.3.4 节），因此按理说我们使用本地模式安装 Express 即可。但是 Express 像很多框架一样都提供了 Quick Start（快速开始）工具，这个工具的功能通常是建立一个网站最小的基础框架，在此基础上完成开发。当然你可以完全自己动手，但我还是推荐使用这个工具更快速地建立网站。为了使用这个工具，我们需要用全局模式安装 Express，因为只有这样才能在命令行中使用它。运行以下命令：

```
$ npm install -g express
```

我遇到 express 不是内部或外部命令

等待数秒后安装完成，我们就可以在命令行下通过 `express` 命令快速创建一个项目了。在这之前先使用 `express --help` 查看帮助信息：

```
Usage: express [options] [path]
```

查看版本: `express -V`

Options:

注意 `express -V` 中的 V 要大写，不然很多版本中会不识别

```
-s, --sessions          add session support
```

```
-t, --template <engine> add template <engine> support (jade|ejs). default=jade
```

卸载: 命令 `npm uninstall -g express`

最新 express4.0 版本 安装命令工具 `npm install -g express-generator`

```
-c, --css <engine>    add stylesheet <engine> support (stylus). default=plain css
-v, --version          output framework version
-h, --help             output help information
```

Express 在初始化一个项目的时候需要指定模板引擎，默认支持Jade和ejs，为了降低学习难度我们推荐使用 ejs^①，同时暂时不添加 CSS 引擎和会话支持。

5.2.2 建立工程 **首先要 / 到目录里面去这里**

通过以下命令建立网站基本结构：

```
express -t ejs microblog
```

当前目录下出现了子目录 microblog，并且产生了一些文件：

```
create : microblog
create : microblog/package.json
create : microblog/app.js
create : microblog/public
create : microblog/public/javascripts
create : microblog/public/images
create : microblog/public/stylesheets
create : microblog/public/stylesheets/style.css
create : microblog/routes
create : microblog/routes/index.js
create : microblog/views
create : microblog/views/layout.ejs
create : microblog/views/index.ejs
```

dont forget to install dependencies:
\$ **cd** microblog && npm install

这一步很重要，漏掉了 不可以
类似 spm build
会自动加载 package.json

它还提示我们要进入其中运行 **npm install**，我们依照指示，结果如下：

```
ejs@0.6.1 ./node_modules/ejs
express@2.5.8 ./node_modules/express
-- qs@0.4.2
-- mime@1.2.4
-- mkdirp@0.3.0
-- connect@1.8.5
```

根据package.json 里面的东西

来下载 资源

它自动安装了依赖 ejs 和 express。这是为什么呢？检查目录中的 package.json 文件，内容是：

① ejs (Embedded JavaScript) 是一个标签替换引擎，其语法与 ASP、PHP 相似，易于学习，目前被广泛应用。Express 默认提供的引擎是 jade，它颠覆了传统的模板引擎，制定了一套完整的语法用来生成 HTML 的每个标签结构，功能强大但不易学习。

```
{
  "name": "microblog"
, "version": "0.0.1"
, "private": true
, "dependencies": {
    "express": "2.5.8"
  , "ejs": ">= 0.0.1"
  }
}
```

其中 `dependencies` 属性中有 `express` 和 `ejs`。无参数的 `npm install` 的功能就是检查当前目录下的 `package.json`，并自动安装所有指定的依赖。

5.2.3 启动服务器

用 Express 实现的网站实际上就是一个 Node.js 程序，因此可以直接运行。我们运行 `node app.js`，看到 Express server listening on port 3000 in development mode。

接下来，打开浏览器，输入地址 `http://localhost:3000`，你就可以看到一个简单的 Welcome to Express 页面了。如果你能看到如图5-2所示的页面，那么说明你的设定正确无误。



图5-2 Express 初始欢迎页面

要关闭服务器的话，在终端中按 `Ctrl + C`。注意，如果你对代码做了修改，要想看到修改后的效果必须重启服务器，也就是说你需要关闭服务器并再次运行才会有效果。如果觉得有些麻烦，可以使用 `supervisor` 实现监视代码修改和自动重启，具体使用方法参见 3.1.3 节。