



Como o Drupal Funciona

Neste capítulo, você terá uma visão geral do Drupal. Os detalhes de como cada parte funciona serão fornecidos nos próximos capítulos. Este capítulo mostrará a pilha de tecnologias em cima da qual o Drupal funciona, os formatos (layouts) dos arquivos que compõem o Drupal e os vários termos conceituais que o Drupal utiliza, como nós (nodes), ganchos (hooks), blocos (blocks) e temas (themes).

O que é o Drupal?

Drupal é utilizado para construir websites. É um framework altamente modular, e um gerenciador de conteúdo de websites de código aberto com ênfase em colaboração. É extensível, seguidor de padrões, e trabalha para ter um código limpo e uma base pequena. Drupal vem com funcionalidade básica do núcleo, e funcionalidades adicionais podem ser incluídas, ao habilitar módulos internos ou de terceiros de módulos. O Drupal foi desenhado para ser customizado, mas a customização é feita ao sobrepor o núcleo ou ao adicionar módulos, mas não por meio da modificação do núcleo do código. O formato do Drupal também separa perfeitamente o gerenciamento do conteúdo da apresentação do conteúdo.

Drupal pode ser utilizado, dentre as muitas possibilidades que tem, para construir um portal na Internet, um website pessoal, departamental, ou corporativo; um site de e-commerce, um diretório de recurso, um jornal on-line, uma galeria de imagem e uma intranet. Também pode ser utilizado para cursos de longa distância.

Um dedicado time de segurança trabalha para manter o Drupal seguro, respondendo às ameaças e gerando atualizações de segurança. Uma organização sem fins lucrativos chamada Associação Drupal apoia o Drupal, melhorando a infraestrutura do site www.drupal.org e organizando conferências e eventos sobre o Drupal. E uma forte comunidade on-line de usuários, administradores de site, designers e desenvolvedores de sites trabalham arduamente na contínua melhoria do software. Vide <http://drupal.org> e <http://groups.drupal.org>.

A Pilha de Tecnologias

Os objetivos do projeto do Drupal incluem tanto a capacidade de funcionar bem em sites de hospedagem baratos, quanto a capacidade de escalar até sites maciçamente distribuídos. O primeiro objetivo significa a utilização das mais populares tecnologias; e o último significa uma codificação cuidadosa e ajustada. A pilha de tecnologias do Drupal é ilustrada na Figura 1-1.

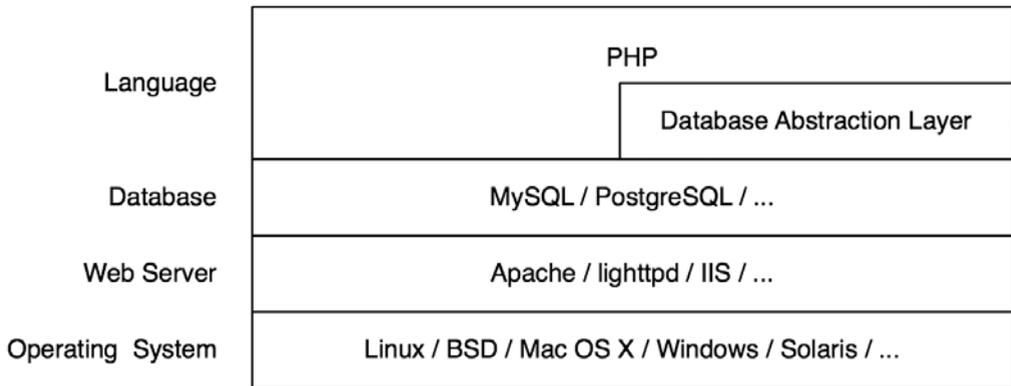


Figura 1-1. A Pilha de Tecnologias do Drupal

O sistema operacional está num nível tão baixo da pilha que não chega a ser tão importante para o Drupal. O Drupal funciona perfeitamente com qualquer sistema operacional que comporte a linguagem PHP.

O servidor web mais utilizado com Dupral é o Apache, embora outros servidores de web (incluindo IIS da Microsoft) possam ser utilizados. Devido ao longo relacionamento do Drupal com o Apache, ele trabalha com arquivos `.htaccess` que asseguram a instalação do Drupal. *URLs Limpas* – isto significa: aquelas com falta de pontos de interrogação (?), ampersands (simbolizados por &), ou outros caracteres estranhos – são realizados utilizando o componente `mod_rewrite` do Apache. Isto é particularmente importante porque, ao migrar de outro sistema de gerenciamento de conteúdo ou de arquivos estáticos, as URLs de conteúdo não precisarão ser alteradas, e não precisar fazer isso é bem melhor, segundo Tim berners-Lee (<http://uwu.u3,opg-Ppotider/Style/URI>). URLs limpas estão disponíveis em outros servidores, por meio da utilização da habilidade reescrever URL do servidor de WEB.

A interface do Drupal com a próxima camada da pilha (o banco de dados) se dá através de uma camada de abstração. Essa camada cuida do saneamento das consultas em SQL e possibilita o uso de bancos de dados de diversos fornecedores sem ter que reescrever o código do Drupal. Os bancos de dados mais amplamente testados são o MySQL e o PostgreSQL, sendo que o suporte ao Microsoft SQL Server e ao Oracle está aumentando.

Drupal é escrito em linguagem PHP. Como a linguagem PHP é fácil de aprender, há muitos programas em PHP feitos por iniciantes. A qualidade de código feito por iniciantes atribuiu à linguagem uma má reputação. Entretanto, a linguagem PHP também pode ser utilizada para escrever código sólido. Todo o núcleo do código do Drupal adere a padrões de codificações estritos (<http://drupal.org/nodes/318>) e passa pela revisão através do processo de modelo de código aberto. Para o Drupal, a curva de aprendizagem fácil da linguagem PHP significa que há uma pequena barreira para a entrada de novos colaboradores, e o processo de revisão assegura que este fácil acesso aconteça sem perda da qualidade final do produto. O feedback que os iniciantes recebem da comunidade ajuda-os a melhorar suas habilidades.

Núcleo

Um framework leve compõe o **núcleo** (*core*) do Drupal. É o que você baixa do site do Drupal (drupal.org). O núcleo é responsável por fornecer a funcionalidade básica que servirá de suporte às outras partes do sistema.

O núcleo inclui o código que permite que o sistema do Drupal faça um bootstrap¹ ao receber uma requisição, uma biblioteca de funções comuns utilizada frequentemente com Drupal, e módulos que fornecem funcionalidade básica, como gerenciamento de usuários, taxonomia, templates como exibidos na figura 1-2.

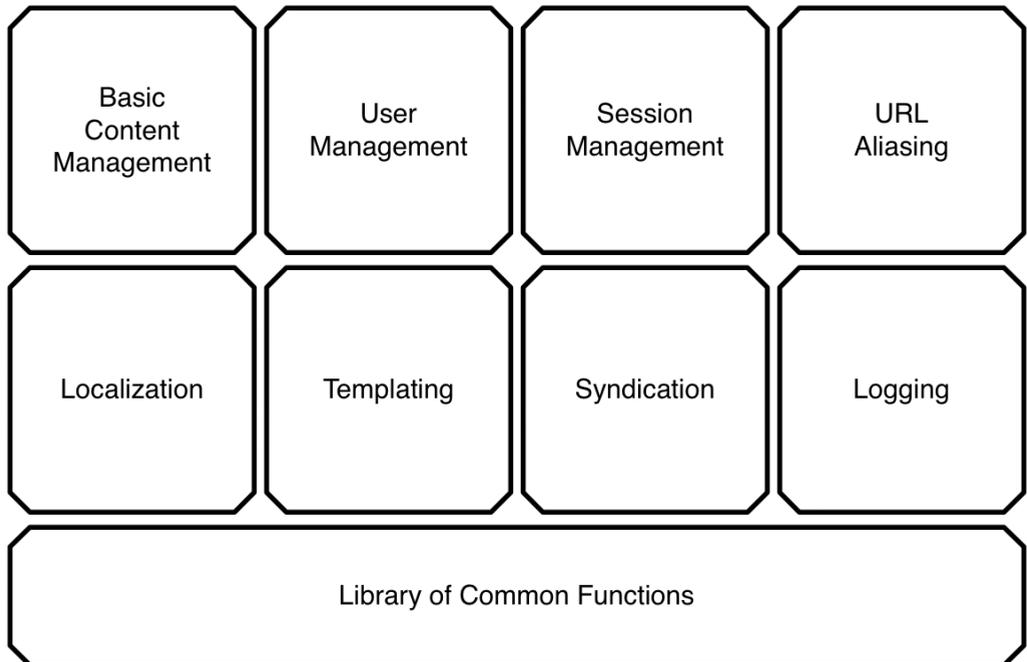


Figura 1-2. Uma visão geral do Núcleo do Drupal (nem todas as funcionalidades do núcleo foram exibidas)

Interface Administrativa

A interface administrativa do Drupal é rigidamente integrada com o resto do site e, por padrão, utiliza o mesmo tema visual. O primeiro usuário, usuário 1, é o superusuário e tem completo acesso ao site. Depois de fazer o login com este usuário, você verá um link **Administer** dentro do seu bloco de usuário (vide seção "Blocos"). Ao clicar no link, você entrará na interface administrativa do Drupal. Cada bloco de usuário terá diferentes links, dependendo do nível de permissão do acesso ao site de cada um.

Módulos

O Drupal é um framework totalmente modular. A funcionalidade está incluída em *módulos*, que podem ser habilitados ou desabilitados (alguns módulos não podem ser desabilitados). Funcionalidades são adicionadas a um site com Drupal habilitando módulos existentes, instalando módulos escritos por membros da comunidade Drupal ou escrevendo novos módulos. Desta forma, os websites que não precisam de certas funções podem funcionar sem recursos, enquanto aqueles websites que precisam de mais funções podem adicionar quantas funcionalidades eles desejarem. Isto é exibido na figura 1-3.

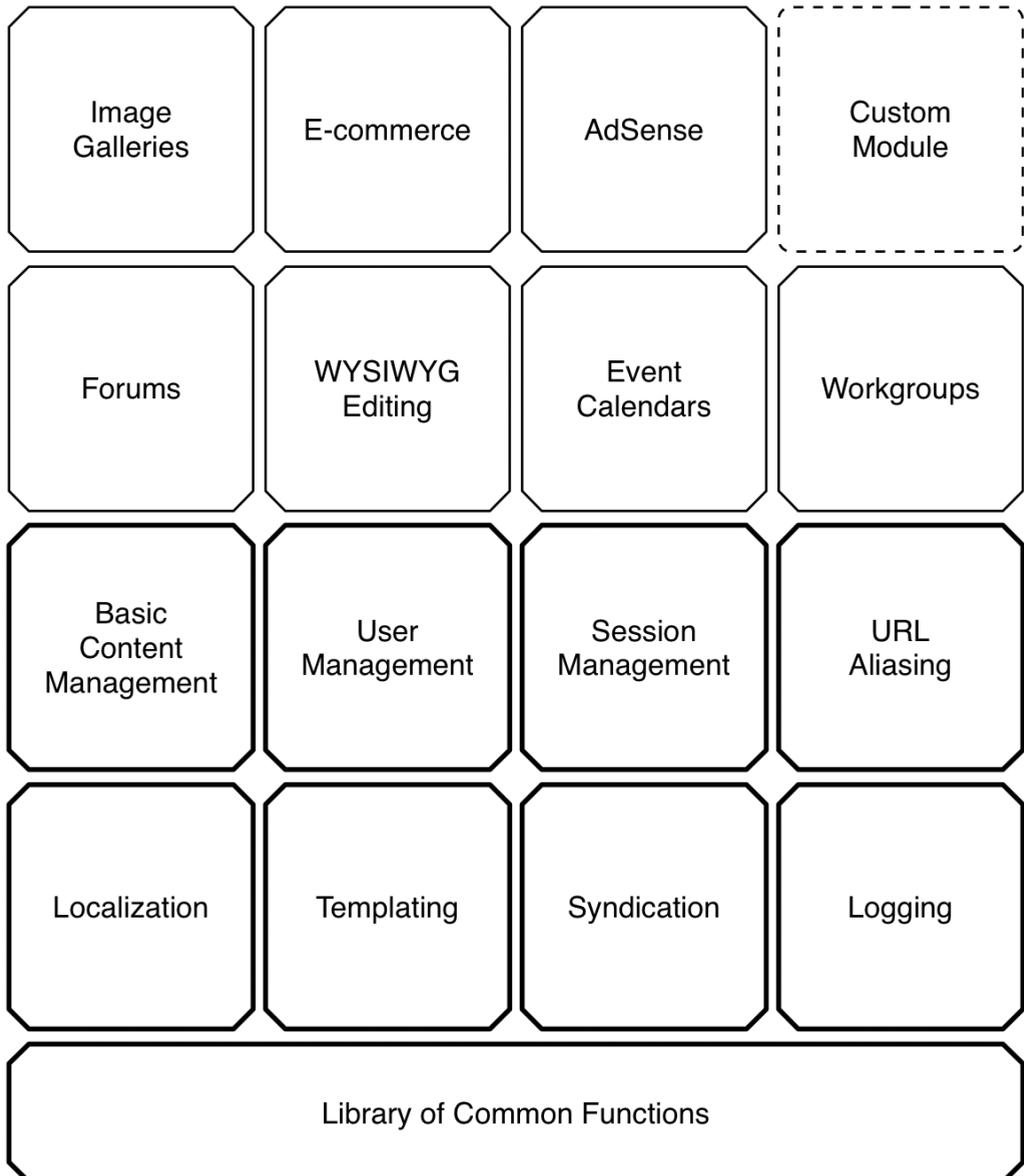


Figura 1-3. Permite incluir módulos adicionais para obter maior funcionalidade.

Tanto a adição de novo conteúdo, como fórmulas, posts em blogs, ou arquivos, quanto a adição de novas funções, como e-mail de notificação, publicação ponto a ponto e agregação são tratadas por módulos. O Drupal faz uso do padrão de projeto *inversão de controle*, no qual a funcionalidade modular é chamada pelo framework no momento apropriado. Essas oportunidades para os módulos fazerem seus trabalhos é o que estamos chamando de ganchos (*hooks* no original em inglês).

Ganchos

Ganchos podem ser vistos como eventos internos do Drupal. Eles são chamados também de *callbacks*, entretanto por serem construídos através de convenções na nomenclatura das funções e não por registro em um ouvinte (*listener*), eles não estão sendo verdadeiramente chamados para o retorno (*called back*). Os ganchos permitem aos módulos “se encaixarem” no que está acontecendo no resto do Drupal.

Suponha que um usuário faça o login em seu website Drupal. Assim que ele fizer isso, o Drupal dispara o gancho *user*. Isso significa que qualquer função nomeada de acordo com a convenção *nome do módulo* mais *nome do gancho* será acionada. Por exemplo, `comment_user()` no módulo de comentário, `locale_user()` no módulo de localização, `node_user()` no módulo de nó e quaisquer outras funções de nomeação similares serão acionadas. Se você fosse escrever um módulo personalizado `spammy.module` e incluísse a função chamada `spammy_user()`, que envia um e-mail ao usuário, esta função seria acionada também, e o usuário receberia um e-mail não solicitado a cada login.

A forma mais comum de explorar a funcionalidade do núcleo do Drupal é pela implementação de ganchos nos módulos.

-
- **Dica** Para mais detalhes sobre o suporte Drupal para ganchos, verifique a documentação on line no site <http://api.drupal.org/api/6>, e procure em Components of Drupal, depois em “Module system (Drupal hooks).”
-

Temas

Ao criar uma página na internet para enviar a um navegador, há duas preocupações principais: a reunião dos dados apropriados e a marcação dos dados para a web. No Drupal, a camada de temas é responsável por criar o HTML (ou JSON, XML, etc.) que o navegador vai receber. O Drupal pode utilizar diversas abordagens de template populares, como Smarty, Template Attribute Language for PHP (PHPTAL) e modelo PHPT.

É importante lembrar que o Drupal incentiva a separação do conteúdo e marcação.

O Drupal disponibiliza várias formas de customizar e sobrepor o visual e jeito do seu website. A forma mais fácil é utilizar CSS (Cascading Style Sheet) para sobrepor a construção das classes e IDs do Drupal. Entretanto, se você fizer mais que isso e tentar customizar a saída atual do HTML, você achará fácil de ser feito. Os arquivos de template do Drupal consistem em HTML padrão e PHP. Além disso, cada parte dinâmica de uma página Drupal (como caixas, listas, ou breadcrumbs) pode ser sobreposta de forma fácil, ao escrever uma função com o nome apropriado. Então, o Drupal utilizará sua função ao invés de criar aquela parte da página.

Nós

Os tipos de conteúdo do Drupal são derivados de um único tipo de base, conhecido como *nó* (*node*). Se for uma entrada de blog, fórmulas, ou até mesmo uma tarefa de um projeto, a estrutura de dados de base é a mesma. A genialidade por trás desta forma de abordagem está na capacidade de extensão dos nós. Os desenvolvedores de módulos podem adicionar funções como classificações, comentários, anexos, informação da localização geográfica e muito mais para os nós, em geral sem se preocupar se o tipo de nós é um blog, fórmula etc. O administrador do site pode, então, misturar e combinar funcionalidade pelo tipo de conteúdo. Por exemplo, o administrador pode deixar a função de comentários de blog ativada, mas não fórmulas, ou somente o upload de arquivo para tarefas de projeto.

Os nós contêm também uma base de propriedades comportamentais que todos os outros tipos de conteúdo herdam. Qualquer nó pode ser promovido a página frontal do web site, publicado ou ser retirado de circulação, ou mesmo buscado. E por causa desta estrutura uniforme, a interface administrativa pode oferecer uma tela de edição em lote (batch editing screen) para trabalhar com nós.

Blocos

Um *bloco* é a informação que pode ser habilitada ou desabilitada em um determinado local do template do seu web site. Por exemplo, um bloco poderia exibir um número de usuários atuais em seu site. Você poderá ter um bloco contendo links para os conteúdos mais acessados do site, ou uma lista de eventos programados. Os blocos são tipicamente colocados na barra lateral (sidebar), cabeçalho (header) ou rodapé (footer) do template, somente na página principal, ou de acordo com outros critérios.

Os blocos são frequentemente utilizados para apresentar informações que são customizadas para os usuários atuais. Por exemplo, o bloco do usuário só tem links para áreas administrativas do site, as quais o usuário tem permissão de acesso, como a página "Minha conta". Regiões onde os blocos podem aparecer (tais como cabeçalho, rodapé, ou barra lateral esquerda ou direita) são definidas no tema do site, localização e visibilidade dos blocos dentro daquelas regiões que são gerenciadas por meio da interface administrativa on-line.

O Layout dos Arquivos

Entender a estrutura do diretório da instalação padrão do Drupal vai lhe ensinar muitas práticas importantes, como onde o download de módulos e temas deveria ficar e como ter diferentes perfis de instalação do Drupal. Veja na figura 1-4 uma instalação padrão do Drupal.

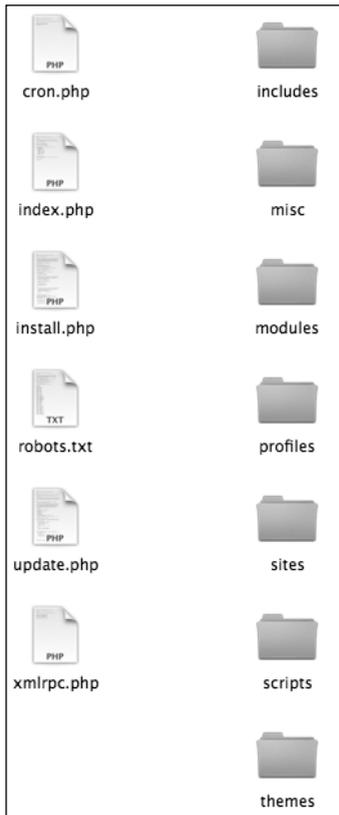


Figura 1-4. O padrão da estrutura do diretório de uma instalação Drupal.

Veja abaixo os detalhes de cada elemento da estrutura do diretório:

A pasta de diretório `includes` contém bibliotecas de funções comuns que o Drupal utiliza.

A pasta de diretório `misc` arquiva JavaScript e diferentes ícones e imagens disponíveis para o estoque de instalação do Drupal.

A pasta de diretório `modules` contém os módulos nucleares, tendo cada módulo a sua própria pasta. É aconselhável não mexer em qualquer uma das pastas (ou qualquer pasta, com exceção das pastas `profiles` e `sites`). Você adiciona módulos extras na pasta de diretório `sites`.

A pasta de diretório `profiles` contém diferentes perfis de instalação para o site. Se houver outros perfis além do perfil padrão neste subdiretório, o Drupal perguntará qual perfil você deseja instalar no momento da primeira instalação do site. O objetivo principal de um perfil de instalação é habilitar automaticamente certos módulos do núcleo e de contribuições. Um exemplo seria um perfil de um site de e-commerce que estabelece automaticamente o Drupal como uma plataforma e-commerce.

A pasta `scripts` contém scripts para checar sintaxe, limpar código, executar o Drupal a partir de linha de comando e lidar com casos especiais usando o cron. Esta pasta não é usada no ciclo de vida de uma requisição; eles são scripts utilitários feitos em shell e Perl.

O diretório `sites` (vide figura 1-5) contém as suas alterações no Drupal, divididas em: configurações, módulos e temas. Quando você adiciona um módulo ao Drupal do repositório de módulos adquiridos

ou ao escrever o seu próprio módulo, eles ficam dentro de `sites/all/modules`. Este arquivo guarda todas as suas alterações do Drupal em uma única pasta. Dentro do diretório `sites` haverá um subdiretório chamado `default`, que armazena o arquivo de configuração padrão do seu site Drupal – `default.settings.php`. O instalador do Drupal modificará estas configurações originais com base na informação dada e vai escrever um arquivo `settings.php` para o seu site. O diretório padrão é copiado e renomeado para a URL do seu site pela pessoa responsável pelo site, então a sua pasta final de configurações seria `sites/www.example.com/settings.php`.

A pasta `sites/default/files` não vem com o Drupal por padrão, mas é necessária para guardar quaisquer arquivos provenientes de upload e que subsequentemente serão servidos. Alguns exemplos são os usos de logos personalizados, possibilitar a criação de avatares, ou fazer o upload de outras mídias associadas ao seu novo site. Este subdiretório requer permissão para leitura e escrita pelo servidor de web que funciona por trás do Drupal. O instalador do Drupal criará este subdiretório, se ele conseguir, e analisará se as permissões corretas foram estabelecidas.

A pasta de diretório `themes` contém mecanismos de template e temas padrão para o Drupal. Temas adicionais que você pode criar ou obter através de download não devem ficar aqui. Eles ficam dentro de `sites/all/themes.cron.php` e são utilizados para a execução de tarefas periódicas, como podas em tabelas de banco de dados e cálculo de estatísticas.

`index.php` é o ponto de entrada principal para atender a requisições.

`install.php` é o ponto de entrada principal para o instalador do Drupal.

`update.php` atualiza o esquema de banco de dados depois de que você fizer um upgrade na versão do Drupal.

`xmlrpc.php` recebe requisições XML-RPC e pode ser removido sem problema de distribuições que não pretendem receber requisições XML-RPC.

`robots.txt` é a implementação padrão do padrão para exclusão de robô (robot exclusion standard).

Outros arquivos não listados aqui são os arquivos de documentação.

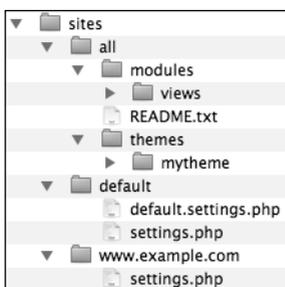


Figura 1-5. As pastas do site podem armazenar todas as suas modificações no Drupal.

Atendendo a Requisições

É útil ter um framework conceitual sobre o que acontece quando uma requisição é recebida pelo Drupal, então esta seção fornecerá uma rápida visão sobre isso. Se você quiser rastrear sozinho os acontecimentos, utilize um bom debugger e comece em `index.php`, lugar onde o Drupal recebe

a maioria de suas requisições. A sequência descrita nesta seção pode parecer complexa apenas para exibir uma página web simples, mas predomina a flexibilidade.

O Papel do Servidor de Web

O Drupal funciona com um servidor de web por trás, geralmente o Apache. Se o servidor de web respeita o arquivo `.htaccess` do Drupal, algumas configurações do PHP são inicializadas e a URL é verificada. Quase todas as chamadas ao Drupal passam por `index.php`. Por exemplo, uma chamada a `http://example.com/foo/bar` segue o seguinte processo:

1. A regra de `mod_rewrite` no arquivo `.htaccess` do Drupal olha a URL que veio, separando a base dessa URL do caminho. No exemplo dado o caminho é `foo/bar`.
2. Este caminho é atribuído ao parâmetro `q` da consulta da URL.
3. A URL resultante é `http://example.com/index.php?q=foo/bar`.
4. O Drupal trata `foo/bar` como um caminho interno e o processo começa com `index.php`.

Como resultado deste processo, o Drupal trata `http://example.com/index.php?q=foo/bar` e `http://example.com/foo/bar` exatamente da mesma forma, porque internamente o caminho é o mesmo em ambos os casos. Isto possibilita que o Drupal utilize URLs sem caracteres estranhos. Estas URLs são conhecidas como as URLs limpas.

Em outros servidores web, como IIS da Microsoft, as URLs limpas podem ser obtidas utilizando um módulo de ISAPI (Windows Internet Server Application Programming Interface), como o ISAPI Rewrite. Versões 7 e superiores do IIS podem dar, diretamente, suporte a reescrita de URL.

O processo de Bootstrap

O Drupal faz um bootstrap (age sozinho) a cada requisição, através de uma série de fases de bootstrap. Estas fases estão definidas em `bootstrap.inc` e ocorrem como descrito nas próximas seções.

Inicializar Configuração

Esta fase preenche a matriz interna de configuração do Drupal e estabelece a base da URL do site (`$base_url`). O arquivo `settings.php` é analisado via `include_once()`, e qualquer ordem para sobrescrever variável ou string que tenha sido estabelecida no arquivo é aplicada. Vide as seções “Variable Overrides” e “String Overrides” do arquivo `sites/all/default/default.settings.php` para mais detalhes.

Cache Inicial de Página

Em situações que requerem um alto nível de escalabilidade, um sistema de cache pode precisar ser invocado antes mesmo que a conexão com o banco de dados seja tentada. O cache inicial de página permite que você inclua (com `include()`) um arquivo PHP que contém uma função chamada `page_cache_fastpath()`, que assume o controle e retorna conteúdo para o navegador. O cache inicial de página é habilitado ao colocar a variável `page_cache_fastpath` para `TRUE`, e o arquivo a ser incluído é definido configurando a variável `cache_inc` com o endereço do arquivo. Vide exemplo no capítulo sobre cache.

Inicializar o Banco de Dados

Durante a fase de banco de dados, o tipo de banco de dados é determinado e uma conexão inicial é feita para ser utilizada com as consultas feitas no banco de dados.

Controle de Acesso Baseado em IP/Hostname

O Drupal permite o banimento de host baseado em endereço IP/hostname. Na fase de controle de acesso, uma verificação rápida é feita para ver se a requisição veio de um host banido; se for o caso, o acesso é negado.

Inicializar Manipulação de Sessão

O Drupal aproveita a implementação de manipulação de sessão do PHP, mas sobrescreve alguns dos manipuladores (handlers) para poder implementar uma manipulação de sessão baseada em banco de dados. Sessões são estabelecidas ou restabelecidas na fase de sessão. O objeto global `$user`, que representa o usuário corrente, é inicializado nessa fase também, entretanto, por uma questão de eficiência, nem todas as propriedades desse objeto ficam disponíveis (elas são adicionadas por uma chamada explícita a função `user_load()` quando forem necessárias).

Cache Final de Página

Na fase de cache final de página, o Drupal carrega código de suporte o suficiente para determinar se servir uma página do cache é apropriado ou não. Isso inclui juntar configurações que estão no banco de dados com a matriz que foi criada durante a fase de configuração e carregar e analisar código do módulo. Se a sessão indicar que a requisição foi feita por um usuário anônimo e o cache de página estiver habilitado, então a página retornada virá do cache e a execução terminará.

Escolha do Idioma

Na fase de escolha do idioma, o suporte multilíngue do Drupal é inicializado, e é decidido qual idioma será utilizado ao servir a atual página. A decisão será baseada em configurações do site e do usuário. O Drupal tem várias alternativas para determinar o suporte de idioma, como prefixo de caminho e negociação de linguagem em nível de domínio.

Caminho

Na fase de caminho, é carregado código que manipula caminhos (paths) e os *alias* de caminho (*path aliasing*). Esta fase habilita a resolução de URLs humanamente legíveis e manipula cache e lookup para os caminhos internos do Drupal.

Completo

Esta fase completa o processo de bootstrap ao carregar uma biblioteca de funções comuns, suporte à temas, suporte a mapeamento de callback, manipulação de arquivos, Unicode, ferramentas do PHP para manipulação de imagens, criação e processamento de formulários, manipulação de mail, tabelas automaticamente ordenáveis e paginação de resultados. O manipulador de erros customizado do Drupal é configurado e todos os módulos habilitados são carregados. Finalmente, o Drupal dispara o gancho `init`, de forma que os módulos dispõem de uma oportunidade de serem notificados antes que o processamento oficial da requisição comece.

Uma vez que o Drupal completou o processo de bootstrap, todos os componentes do framework ficam disponíveis. É a hora de pegar a requisição do navegador e mandá-la para a função que irá manipulá-la. O mapeamento entre URLs e funções que as manipulam é conseguido através do uso de um registro de callback que toma conta tanto do mapeamento de URL, quanto do controle de acesso. Os módulos registram seus callbacks usando o gancho `menu` (para mais detalhes, veja o Capítulo 4).

Quando o Drupal determina que existe um callback que é mapeado perfeitamente pela URL da requisição do navegador e que o usuário tem a permissão para acessar esse callback, então o controle é passado para essa função de callback.

Processando uma Requisição

A função callback faz qualquer trabalho solicitado para processar e acumular os dados necessários para concluir a requisição. Por exemplo, se uma requisição de conteúdo como `http://example.com/q=node/3` for recebida, a URL é mapeada para a função `node_page_view()` em `node.module`. Processos posteriores vão resgatar os dados do banco de dados e colocá-los dentro de uma estrutura de dados. O próximo passo é a utilização dos temas.

Tematizar os dados

Criar um tema para os dados (theming) envolve a transformação dos dados recuperados, manipulados, ou criados em HTML (ou XML e outro formato de saída). O Drupal utilizará os temas que o administrador selecionou para dar à página da web a aparência e formato corretos. O resultado produzido é, então, enviado ao navegador (ou outro cliente HTTP).

Resumo

Depois de ler este capítulo, você entenderá, de forma geral, como o Drupal funciona e terá uma visão geral do que acontece quando o Drupal atende a uma requisição. Os componentes que compõem o processo de execução da página da web serão abordados em detalhes nos próximos capítulos.

