

Introdução ao Django

Introdução

Neste primeiro artigo do Django, respondemos a pergunta "O que é o Django?" e daremos uma visão geral do que torna este framework web especial. Vamos descrever os principais recursos, incluindo algumas das funcionalidades avançadas que não teremos tempo para abordar detalhadamente neste módulo. Também mostraremos alguns dos principais blocos de construção de um aplicativo Django (embora neste momento você ainda não tenha um ambiente de desenvolvimento para testá-lo).

O que é Django?

Django é um framework web Python de alto nível que permite o rápido desenvolvimento de sites seguros e de fácil manutenção. Construído por desenvolvedores experientes, o Django cuida de grande parte do trabalho de desenvolvimento web, para que você possa se concentrar em escrever seu aplicativo sem precisar reinventar a roda. É gratuito e de código aberto, tem uma comunidade próspera e ativa, ótima documentação e muitas opções de suporte gratuito e pago.

Django ajuda você a escrever programas que são:

Completo

Django segue a filosofia de "baterias incluídas" e fornece quase tudo que desenvolvedores possam querer fazer "fora da caixa". Como tudo o que você precisa é parte de um "produto", tudo funciona perfeitamente junto, seguindo princípios de design consistentes, contando uma extensa e atualizada documentação.

Versátil

Django pode ser (e tem sido) utilizado para construir quase todo tipo de website - desde sistema de gestão de conteúdo e wikis, passando por redes sociais e sites de notícias. Ele pode trabalhar com qualquer framework do lado do cliente, e pode entregar conteúdo em praticamente qualquer formato (incluindo HTML, feeds RSS, JSON, XML, etc). Esse site que você está lendo agora é baseado em Django.

À medida em que, internamente, fornece opções para quase todo tipo de funcionalidade que você possa querer (por exemplo: vários banco de dados que são populares, motores de template, etc), ele pode também ser estendido para utilizar outros componentes, caso seja necessário.

Seguro

Django ajuda os desenvolvedores a evitar os erros de segurança mais comuns, fornecendo um framework que foi desenhado para "fazer as coisas certas", de modo a proteger o website automaticamente. Por exemplo, Django fornece uma maneira segura de gerenciar as contas dos usuários e suas senhas, evitando erros comuns, tais como colocar informações da sessão em cookies, onde ficam vulneráveis (ao invés disso os cookies contêm apenas uma chave e os dados são armazenados no banco de dados), ou armazenar as senhas de forma direta, ao invés de gravar um hash para essas senhas.

Um hash de senha é um valor fixed-length (tamanho-fixado) criado mandando a senha por uma cryptographic hash function (função hash criptográfica). Django pode checar se uma senha inserida está correta executando ela pela função hash e comparando a saída com o valor hash armazenado. Porém devido a natureza "one-way" ("um-caminho") da função, mesmo que o valor hash armazenado estiver comprometido, é difícil para uma pessoa comentando um ataque resolver a senha original.

O Django ativa a proteção contra muitas vulnerabilidades por padrão, incluindo SQL injection (injeção de SQL), cross-site scripting, cross-site request forgery (Falsificação de

solicitação entre sites), e clickjacking ("furto de click") (veja Segurança de sites para mais detalhes de tais ataques).

Escalável

Django usa uma arquitetura baseada em componentes "shared-nothing" ("nada-compartilhado") (cada parte da arquitetura é independente das outras, e consequentemente podem ser substituídas ou mudadas caso necessário). Ter uma separação clara entre as partes diferentes significa que pode se escalar para um tráfego aumentado adicionando hardware em qualquer nível: servidores de cache, servidores de banco de dados ou servidores de aplicação. Alguns dos sites mais ocupados escalaram o Django com sucesso para cumprir com as suas demandas (ex: Instagram e Disqus).

Sustentável

O código do Django é escrito usando princípios de design e padrões que encorajam a criação de código sustentável (que facilita a manutenção) e reusável. Em particular, isso utiliza o princípio DRY - Don't Repeat Yourself (Não Repita a Si Mesmo) para que não haja duplicações desnecessárias, reduzindo a quantidade de código. O Django também promove o agrupamento de funcionalidades relacionadas para aplicativos reusáveis e, em um nível mais baixo, grupos de código relacionados para módulos (juntamente as linhas do padrão MVC - Model View Controller).

Portável

Django é escrito em Python, que executa em muitas plataformas. Isso significa que você não está preso em nenhuma plataforma de servidor em particular, e pode executar seus aplicativos em muitas distribuições do Linux, Windows e Mac OS X. Além disso, o Django tem um bom suporte em muitos provedores de servidores de web, que muitas vezes provem infraestrutura específica e documentação para hospedar sites feitos com Django.

De onde o Django veio?

Django foi inicialmente desenvolvido entre 2003 e 2005 por um time de web que era responsável por criar e manter sites de jornal. Depois de criar um número de sites, o time começou a fatorar e reutilizar muitos de seus códigos comuns e padrões de design. Esse código comum evoluiu para um framework genérico de desenvolvimento web, que foi lançado como um projeto de código aberto nomeado "Django" em Julho de 2005.

Django continuou a crescer e aprimorar, desde seu lançamento (1.0) em Setembro de 2008 até a versão recentemente lançada 2.0 em 2017. Cada lançamento adicionou novas funcionalidades e consertou falhas, variando entre suportar novos tipos de banco de dados, mecanismos de template e caches, até a adição de funções view "genéricas" e classes (que reduzem a quantidade de código que os desenvolvedores tem que escrever para um número de tarefas de programação).

Nota: Cheque as notas de lançamento no site do Django para ver o que mudou nas versões mais recentes, e quanto trabalho está sendo feito para tornar o Django melhor.

Django é um projeto de código aberto, colaborativo e próspero, com milhares de usuários contribuindo. Embora ainda tenha alguns recursos que refletem sua origem, Django evoluiu para um framework versátil que é capaz de desenvolver qualquer tipo de website.