DJA101 - DEFENDING DJANGO

Course Learning Objectives

In this course, you'll learn what you need to do to take advantage of Django's built-in security features and provide other layers of protection to your app. You'll learn how to set up your projects securely to prevent attacks at run-time and how to secure the admin console. You will also learn how to identify secure and insecure practices to protect your application against common attacks.

Description

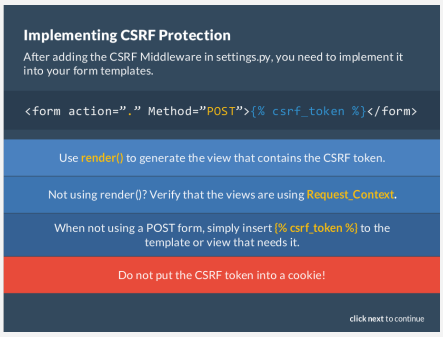
This course was created for those who already have some experience coding in Python and developing web applications with the Django platform. The purpose of this course is to show you secure ways of creating web applications using the Django platform, and is intended to build on what you already know about building web apps in this platform.

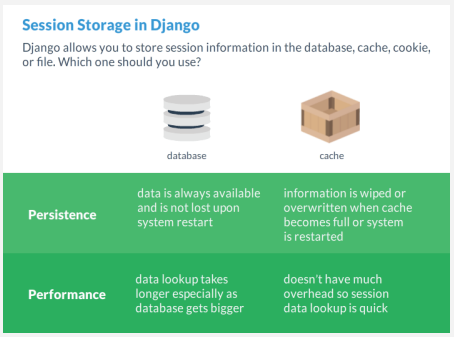
Audience

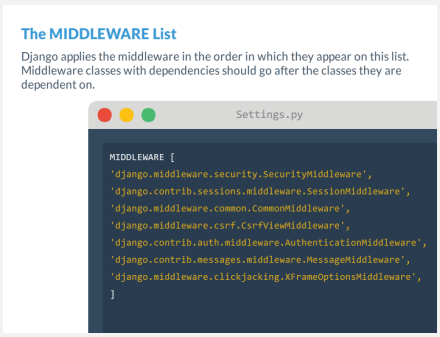
Python developers
Web application developers

Time Required

Tailored learning - 40 minutes total







Security Compass

COPYRIGHT 2017

DJA101 - DEFENDING DJANGO

Course Outline

1. Setting Your Project Up Securely

- Securing the Servers
- Securing the Secret Key
- Installing Django SecurityMiddleware
- The MIDDLEWARE list
- About Clickjacking
- The X-Frame-Options header
- Activating Clickjacking Protection
- About Cross-Site Scripting
- Activating XSS Protection
- About Content or MIME Sniffing
- Activating Content or MIME Sniffing Protection
- Enabling SSL Redirects
- Implementing HTTPS Strict Transport Security
- Configuring HSTS Settings
- Validating Hosts
- Implementing Secure Session Management
- Session Storage in Django
- Cached_db Method
- Setting Up Your Session Storage
- Configuring Your Cookies
- Subdomains and Cookies
- Securing User-Uploaded Content
- Storing User-Uploaded Content
- Configuring Django Server to Secure User-Uploaded Content
- Configuring your Django Model

2. Developing Your Web Apps Securely

- The Django Object-Relational Mappers
- Parameterizing Raw SQL Queries
- Escaping and Sanitizing Character Strings
- Autoescaping
- About CSRF
- Django's CSRF Defense
- Implementing CSRF Protection
- Using CSRF Protection With Single Views
- Exempting a View from Clickjacking Protection
- Settings and Functionalities to Avoid
- Safeguarding Autoescaping
- Safeguarding HTML Sanitizing
- Including and Excluding Fields When Using ModelForm
- Blocking Python Code Injection
- Using eval(), exec(), and execfile() Safely
- Serialization/Input Parsing

3. Securing Your Django Admin and Server

- Changing the Default Admin URL
- Changing the Default Admin Docs URL
- Creating the Superuser Account
- Restricting Access to Specific IP Addresses
- Updating Django

Security Compass

COPYRIGHT 2017