DJA201 - 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 exploits, such as clickjacking, cross-site scripting, and XML External Entities.

This course also explores how to defend against other exploits using Django settings, how to manage users, and securing your production environment.

Description

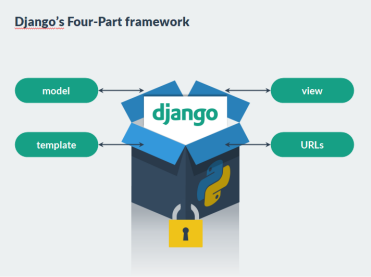
This course was created for developers who already have some experience coding in Python and developing web applications with the Django platform. The purpose of this course is to highlight secure ways of creating web applications, making the best use of Django's built-in security features, and to build on what you already know about using the Django platform.

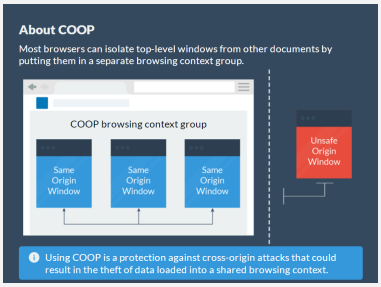
Audience

Python developers
Web application developers

Time Required

Tailored learning - 70 minutes total





DJA201 - DEFENDING DJANGO

Course Outline

1. Setting Up a Django Project

- Django's Four-Part framework
- Django settings.py
- About Django settings.py
- Securing the secret key
- Where to store secret key and other secrets
- How to access secret keys
- Installing Django SecurityMiddleware
- The MIDDLEWARE list
- Securing the Transport Layer
- Configuring HSTS settings
- Referrer policy
- New default policy in Django 3.1+
- Setting up the superuser account
- Environment variables
- Using environment variables

2. Defending Against Exploits 1.

- Blocking clickjacking
- The X-Frame-Options header
- Activating clickjacking protection
- Exempting a view from clickjacking protection
- About XSS
- Protecting HTML code from XSS exploitation
- Temporarily disabling Django's template protection
- About CSRF
- Django's CSRF defense
- Implementing CSRF protection
- Using CSRF protection with single views
- Blocking MIME or content sniffing
- Activating content or MIME sniffing protection
- About XML External Entities (XXE) vulnerability
- Blocking XXE vulnerabilities with the defusedxml package
- Switching from XML to JSON

3. Defending Against Exploits 2.

- About COOP
- Make your website "Cross-Origin Isolated"
- Validating hosts
- Create a list of ALLOWED_HOSTS
- Django Object-Relational Mappers (ORM)
- Parameterize raw SQL queries
- Escaping and sanitizing character strings
- Autoescaping
- Safeguard autoescaping
- HTML sanitation using django-bleach
- Blocking Python code injection
- Use eval(), exec(), and execfile() safely
- Serialization/Input parsing

4. User Management

- Django user authentication
- Custom password validation in Django
- Throttling user authentications
- Configuring django-defender
- Customizing django-defender
- Two-Factor Authentication (2FA)
- Deploying 2FA in Django
- Implementing secure session management
- Session storage options in Django
- Cached_db method
- Setting up session storage
- Configuring session cookies
- User-uploaded Content
- Securing user-uploaded content
- Storing user-uploaded content
- Configuring server for user-uploaded content
- Configuring Django model

5. Securing the Production Environment

- Turn off DEBUG in production
- Production database server
- Update password and privileges for the database admin
- Transferring database particulars to .env file
- Access database information from the .env file
- Create a .gitignore file for .env file
- Protect source code and sensitive data
- Change the default Django admin URL
- Rename the default admin docs directory
- Restrict access to specific IP addresses
- Django-admin-honeypot
- Install django-admin-honeypot
- Test Django modules
- Implement unit tests
- Run manage.py check –deploy
- Use logging and monitoring in Django applications
- Set up Django logging
- Understand your Python dependencies
- Use pip to create a dependencies requirements list
- Update Django

SecurityCompass