

Comprehensive Django REST Framework (DRF) Syllabus

Here's a detailed syllabus that includes both **predefined features** and **user-defined extensions** to DRF, covering topics like **payment integrations**, **map integration**, **email/phone verification**, and more.

1. Introduction to DRF

- What is Django REST Framework (DRF)?
 - Difference between Django and DRF.
 - REST principles and architecture.
 - Installing and configuring DRF.
 - Creating a basic DRF project.
-

2. Core Concepts of DRF

- DRF project structure and settings.
 - HTTP Methods: GET, POST, PUT, PATCH, DELETE.
 - Status codes and their importance.
 - Serializers:
 - `serializers.Serializer` vs. `serializers.ModelSerializer`.
 - Field-level and object-level validation.
 - Custom fields in serializers.
 - Views:
 - Function-Based Views (FBVs).
 - Class-Based Views (CBVs).
 - `APIView`.
 - Generic views and mixins.
 - `ViewSet` and `Routers`.
-

3. Predefined Features in DRF

- **Authentication:**
 - Session, Token, Basic Authentication.
 - JWT Authentication using `djangorestframework-simplejwt`.
- **Permissions:**
 - `AllowAny`, `IsAuthenticated`, `IsAdminUser`.

- Custom permission classes.
 - **Filtering and Pagination:**
 - Filtering with query parameters.
 - `django-filter` integration.
 - Pagination classes: `PageNumberPagination`, `LimitOffsetPagination`, `CursorPagination`.
 - **Throttling:**
 - `AnonRateThrottle` and `UserRateThrottle`.
 - **Versioning APIs.**
 - Content negotiation and response rendering.
-

4. User-Defined Extensions in DRF

Email Verification

- Sending OTP for user registration using `django.core.mail`.
- Implementing email verification using tokens (e.g., `itsdangerous` or JWT).

Phone Number Verification

- SMS-based OTP verification using APIs like Twilio, Nexmo, or Fast2SMS.
- Storing and validating OTP in the database.

Maps Integration

- Google Maps API integration for geocoding and location-based services.
- Displaying and querying locations using latitude and longitude.

Payment Integration

- Using payment gateways like Razorpay, Stripe, or PayPal.
- Implementing payment workflows:
 - Generating payment links.
 - Verifying payment status.
 - Handling webhooks for payment updates.

Social Authentication

- Google, Facebook, and other OAuth integrations using `django-allauth`.
 - JWT integration for social logins.
-

5. Testing APIs

- Unit testing using Django's TestCase.
 - Writing tests for DRF APIs using [APITestCase](#) and [APIClient](#).
 - Mocking external APIs in tests.
-

6. Deployment

- Preparing DRF apps for production.
 - Setting up a server with Gunicorn/NGINX.
 - Deploying using Docker.
 - Using CI/CD pipelines for automated deployments.
-

7. Advanced Topics

- Asynchronous programming with async views in DRF.
 - File upload/download handling.
 - WebSocket integration with Django Channels for real-time APIs.
 - Building GraphQL APIs alongside DRF using libraries like [graphene-django](#).
-

8. Capstone Projects

1. **E-Commerce API:**
 - Product catalog.
 - Cart and checkout system.
 - Payment integration.
 - User authentication.
2. **Food Delivery App API:**
 - Restaurant listings with geolocation.
 - Order placement and tracking.
 - Payment processing.
3. **Real-Time Chat API:**
 - WebSocket-based communication.
 - User authentication and chat history.
4. **Event Management API:**
 - CRUD operations for events.
 - User registration and ticketing.
 - Map-based location display.

Predefined vs. User-Defined Features in DRF

Feature Type	Examples
Predefined	Authentication (Session/Token/JWT), Permissions, Pagination, Serializers, Throttling.
User-Defined	Email/Phone verification, Payment integrations, Maps, Custom validation rules, GraphQL.

9. Tools and Libraries

- **Predefined Tools:**
 - DRF for serializers, views, and authentication.
 - `django-filter` for advanced querying.
 - `django-rest-framework-simplejwt` for JWT authentication.
 - **User-Defined Libraries:**
 - `twilio` for phone verification.
 - `googlemaps` for location services.
 - `razorpay` or `stripe` for payment processing.
 - `celery` for background tasks like email notifications.
-

Custom DRF Extensions

1. **File Upload Handling:** Upload images, files, and videos using DRF.
 2. **Background Tasks:** Use Celery for sending emails, OTPs, or processing large tasks asynchronously.
 3. **Realtime Notifications:** Use Django Channels for WebSocket-based notifications.
-

10. Future Enhancements

- **AI Integration:** Integrating DRF APIs with AI services like OpenAI or TensorFlow.
- **IoT Applications:** Using DRF with MQTT protocols for IoT systems.
- **Scalable Systems:** Best practices for building scalable and distributed systems using DRF.

If you'd like, I can create a **PDF** for this syllabus or provide more detailed examples on any specific topic! Let me know.