API Planning for the News Application

1. Purpose

- Provide a RESTful interface for **third-party clients** to retrieve articles.
- Articles returned should **respect subscriptions**:
  - Only articles from **publishers/journalists the client is "subscribed" to**.
- Output format: **JSON** (optionally XML if required).

2. Endpoints

Endpoint	Method	Purpose	Access
/api/articles/	GET	Retrieve all approved articles	Authenticated / Subscription-based
/api/articles/<id>/	GET	Retrieve a specific article	Authenticated
/api/publishers/<id>/articles/	GET	Retrieve articles from a specific publisher	Authenticated
/api/journalists/<id>/articles/	GET	Retrieve articles from a specific journalist	Authenticated

3. Serializers

- **ArticleSerializer**: title, content, author/journalist, publisher, created_at, is_approved
- **NewsletterSerializer**: title, content, author/journalist, publisher, created_at

Converts Django models to JSON for API responses.

4. Views

- **APIView / ViewSet** (Django REST Framework):
  - Filter by subscriptions of requesting user/client

    - Only return is_approved=True articles
- Can implement **pagination** for performance

5. Permissions

- Use **DRF permissions**:
    - Only authenticated users can access
    - Ensure clients can **only see articles they're "subscribed" to**
- Optional: token-based auth for third-party clients

6. Testing Plan

- Unit tests to ensure:
    - Subscribers get **only their allowed articles**
    - Non-subscribers cannot see restricted content
    - API returns correct JSON structure
- Can use **Postman** for manual testing

7. Integration Notes

- API must **interact with existing models**:
    - Article
    - Publisher
    - Journalist
    - Subscription tables
- Must respect **role-based access**:
    - Reader-like access for API clients
    - Only approved articles returned