

Universities in Nepal - Complete API Payload Documentation

Base paths:

- Admin staff APIs: `api/v1/admin/`
 - Directory admin APIs: `api/v1/admin/directory/`
 - Public directory APIs: `api/v1/directory/`
-

Auth & Permissions

- `api/v1/admin/*` -> `IsAdminUser`
 - `api/v1/admin/directory/*` -> `IsStaffOrAdminUser`
 - `api/v1/directory/*` -> public (no auth)
-

1) Admin APIs (staff /user management)

1.1 Staff account create

`POST api/v1/admin/staff/create`

Payload:

```
{
  "full_name": "Staff Name",
  "email": "staff@example.com",
  "password": "your-password",
  "user_role": "staff"
}
```

1.2 Staff account update

`PATCH api/v1/admin/staff/<uuid:user_id>/update/`

Payload (partial):

```
{
  "full_name": "Updated Name",
  "email": "staff@example.com"
}
```

```
"email": "new@example.com",  
"password": "new-password"  
}
```

1.3 User list

GET api/v1/admin/users/list/

Query params:

- ?active=true|false
- ?role=student|staff

1.4 User active toggle

PATCH api/v1/admin/user/<uuid:user_id>/status-toggle/

1.5 Account delete

DELETE api/v1/admin/staff/<uuid:user_id>/delete/

1.6 Student account create

POST api/v1/admin/student/create

Payload:

```
{  
  "full_name": "Student Name",  
  "email": "student@example.com",  
  "user_role": "student"  
}
```

Handling:

- Admin does **not** send password for student create.
- Server auto-generates a temporary password and sends it by email.
- `user_role` defaults to `student`.

2) Directory Admin APIs

Shared rules

- Requirement `type: file | text`

- Requirement fields: `label`, `type`, `description`, `is_mandatory`
- Scholarship scopes: `university` | `course` | `offering`
- Scholarship payload item:

```
{
  "title": "Merit Scholarship",
  "description": "Given to top entrance scorers.",
  "eligibility": "Proof of rank required.",
  "seats": 3,
  "amount_npr": 50000,
  "coverage_percent": 100,
  "is_published": false
}
```

- Scholarship validation: at least one of `amount_npr` / `coverage_percent` ; coverage must be `0..100` .
- Course `code` optional (auto-derived from name if not provided).
- University `institution_type` : `public` | `private` | `state` (default `public`).
- University now includes `country` .

2.1 Dashboard

D1. Admin dashboard stats

`GET api/v1/admin/directory/dashboard/stats/`

Response:

```
{
  "universities": 4,
  "colleges": 15,
  "courses": 10,
  "offerings": 29,
  "scholarships": 9,
  "students": 3,
  "applications": 0
}
```

2.2 Discipline APIs

1. `POST api/v1/admin/directory/discipline/create/`
2. `PATCH api/v1/admin/directory/discipline/<str:slug>/update/`

3. `DELETE api/v1/admin/directory/discipline/<str:slug>/delete/`
4. `GET api/v1/admin/directory/discipline/list/ ( ?search= )`

Payload:

```
{
  "name": "discipline-name"
}
```

2.3 University APIs

5. University create

`POST api/v1/admin/directory/university/create/`

Supports multipart for `logo` , `cover_image` .

```
{
  "name": "Tribhuvan University",
  "established_year": 1959,
  "institution_type": "public",
  "country": "Nepal",
  "city": "Kirtipur",
  "province": "Bagmati",
  "overview": "text...",
  "salient_features": "text...",
  "logo": "<file>",
  "cover_image": "<file>",
  "address": "Kirtipur, Kathmandu",
  "website": "https://tu.edu.np",
  "email": "info@tu.edu.np",
  "phone": "+9771234567",
  "extra_sections": [{"achievements": "any format"}],
  "is_published": false,
  "scholarships": []
}
```

Handling:

- Requires either `website` or `email` .
- Nested `scholarships` replace university-scope set.
- Supports gallery rows directly in main create API (multipart):
 - `gallery[0][image_file]`
 - `gallery[0][caption]`
 - `gallery[0][sort_order]`

- Can also send existing IDs on update/create f bw:
 - `gallery[0][id]`

6. University update

`PATCH api/v1/admin/directory/university/<str:slug>/update/`

Partial update. `scholarships` list replaces full university-scope set. Also supports gallery rows in the same `gallery[index][...]` format.

7. University list

`GET api/v1/admin/directory/university/list/`

Query params:

- `search`
- `published`
- `institution_type`
- `country`
- `city`
- `province`
- `address`
- `established_year`
- `established_year__gte`
- `established_year__lte`

8. University detail

`GET api/v1/admin/directory/university/<str:slug>/detail/`

Includes:

- base info
- `country`, `city`, `province`, `location`
- `gallery`
- `scholarships`

9. University delete

`DELETE api/v1/admin/directory/university/<str:slug>/delete/`

10. University publish toggle

`PATCH api/v1/admin/directory/university/<str:slug>/publish-toggle/`

```
PATCH api/v1/admin/directory/university/<str:slug>/publish-logger/
```

Handling:

- On unpublish, cascades unpublish to university courses and of ferings.

10a. University scholarship create

```
POST api/v1/admin/directory/university/<str:slug>/scholarship/create/
```

Scope auto-set to `university`.

10b. University gallery update (new)

```
GET api/v1/admin/directory/university/<str:slug>/gallery/update/ PATCH  
api/v1/admin/directory/university/<str:slug>/gallery/update/
```

PATCH multipart format:

- `gallery[0][id]` (optional existing image UUID)
- `gallery[0][image_file]` (required for new row, optional for replacement)
- `gallery[0][caption]`
- `gallery[0][sort_order]`

Repeat for index 1..N.

Handling semantics:

- Replace-style update.
- Only rows sent in request are kept.
- Existing gallery items missing from request are deleted.

2.4 Course APIs

11. Course create

```
POST api/v1/admin/directory/course/create/
```

Supports multipart for `curriculum_file` (PDF).

```
{  
  "university": "university-slug",  
  "discipline": "discipline-slug",  
  "name": "BSc Computer Science",  
  "code": "BSC CSIT",  
  "level": "bachelor",  
  "duration_years": 4.0,
```

```
"description": "text...",
"career_prospects": "text...",
"eligibility": "text...",
"curriculum_text": "text...",
"curriculum_file": "<file>",
"requirements": [],
"scholarships": [],
"is_published": false
}
```

Rules:

- `level` : `bachelor|master|diploma|phd`
- duration bounds enforced by level.

12. Course update

`PATCH api/v1/admin/directory/course/<str:slug>/update/`

`requirements` or `scholarships` replace full list when provided.

13. Course list

`GET api/v1/admin/directory/course/list/`

Query:

- `search` , `published` , `university` , `discipline` , `level`

14. Course detail

`GET api/v1/admin/directory/course/<str:slug>/detail/`

15. Course publish toggle

`PATCH api/v1/admin/directory/course/<str:slug>/publish-toggle/`

16. Course delete

`DELETE api/v1/admin/directory/course/<str:slug>/delete/`

16a. Course requirement create

`POST api/v1/admin/directory/course/<str:slug>/requirement/create/`

16b. Course requirement list

`GET api/v1/admin/directory/course/<str:slug>/requirement/list/`

16c. Course requirement update

```
PATCH api/v1/admin/directory/course-requirement/<uuid:requirement_id>/update/
```

16d. Course requirement delete

```
DELETE api/v1/admin/directory/course-requirement/<uuid:requirement_id>/delete/
```

16e. Course scholarship create

```
POST api/v1/admin/directory/course/<str:slug>/scholarship/create/
```

2.5 College APIs

17. College create

```
POST api/v1/admin/directory/college/create/
```

Supports multipart for `logo` , `cover_image` .

`universities` may be JSON string in multipart. Also supports gallery rows in the same request:

- `gallery[0][image_file]`
- `gallery[0][caption]`
- `gallery[0][sort_order]`
- optional `gallery[0][id]` for existing row updates/replacements

18. College update

```
PATCH api/v1/admin/directory/college/<str:slug>/update/
```

Handling:

- `universities` replaces full affiliation list.
- Gallery is handled in main update API using `gallery[index][...]` multipart keys.

19. College list

```
GET api/v1/admin/directory/college/list/
```

Query:

- `search` , `published` , `university` , `address` , `district` , `city` , `province` , year filters

20. College detail

```
GET api/v1/admin/directory/college/<str:slug>/detail/
```

21. College publish toggle

```
PATCH api/v1/admin/directory/college/<str:slug>/publish-toggle/
```

22. College delete

```
DELETE api/v1/admin/directory/college/<str:slug>/delete/
```

2.6 Affiliation APIs

22a. Create

```
POST api/v1/admin/directory/affiliation/create/
```

22b. List

```
GET api/v1/admin/directory/affiliation/list/
```

Query:

- university
- college
- is_active

22c. Detail

```
GET api/v1/admin/directory/affiliation/<int:affiliation_id>/detail/
```

22d. Update

```
PATCH api/v1/admin/directory/affiliation/<int:affiliation_id>/update/
```

22e. Delete

```
DELETE api/v1/admin/directory/affiliation/<int:affiliation_id>/delete/
```

Handling:

- Cannot deactivate/delete while of étrings exist for that university-college pair.
-

2.7 Course Of étring APIs

Managed under `college/<slug>/...`.

23. Create

POST `api/v1/admin/directory/college/<str:slug>/offering/create/`

24. Requirements update

- PATCH
`api/v1/admin/directory/college/<str:slug>/offering/<uuid:offering_id>/requirements/`
- PATCH
`api/v1/admin/directory/college/<str:slug>/course/<str:course_slug>/requirements/`

25. Scholarships update

- PATCH
`api/v1/admin/directory/college/<str:slug>/offering/<uuid:offering_id>/scholarships/`
- PATCH
`api/v1/admin/directory/college/<str:slug>/course/<str:course_slug>/scholarships/`

26. Update

- PATCH
`api/v1/admin/directory/college/<str:slug>/offering/<uuid:offering_id>/update/`
- PATCH
`api/v1/admin/directory/college/<str:slug>/course/<str:course_slug>/update/`

27. College of offering list

GET `api/v1/admin/directory/college/<str:slug>/offering/list/`

Query:

- `course` , `university` , `published` , `is_open` , `search`

28. Detail

- GET
`api/v1/admin/directory/college/<str:slug>/offering/<uuid:offering_id>/detail/`
- GET `api/v1/admin/directory/college/<str:slug>/course/<str:course_slug>/detail/`

Includes:

- `course + college info`

- pricing and admission fields
- `description_override` , `eligibility_override`
- `course_requirements` , `offering_requirements` , merged `requirements`
- scholarship breakdown + merged scholarships
- publish/open flags

29. Publish toggle

- PATCH
`api/v1/admin/directory/college/<str:slug>/offering/<uuid:offering_id>/publish-toggle/`
- PATCH
`api/v1/admin/directory/college/<str:slug>/course/<str:course_slug>/publish-toggle/`

30. Delete

- DELETE
`api/v1/admin/directory/college/<str:slug>/offering/<uuid:offering_id>/delete/`
- DELETE
`api/v1/admin/directory/college/<str:slug>/course/<str:course_slug>/delete/`

31. Global offering list

`GET api/v1/admin/directory/offering/list/`

Query:

- `college` , `course` , `university` , `published` , `is_open` , `search`

Current response fields:

- `offering_id` , `id` , `title`
- `course_slug` , `course_name`
- `college_slug` , `college_name`
- `university_slug` , `university_name`
- `annual_fee_npr` , `seats` , `intake_period` , `application_deadline`
- `is_open` , `is_published` , `status`

32. Offering requirement create

- POST
`api/v1/admin/directory/college/<str:slug>/offering/<uuid:offering_id>/requirements/create/`

```
l/create/
```

- POST
api/v1/admin/directory/college/<str:slug>/course/<str:course_slug>/requirement/create/

33. Of fering requirement update

```
PATCH api/v1/admin/directory/college/<str:slug>/offering-requirement/<uuid:requirement_id>/update/
```

34. Of fering requirement delete

```
DELETE api/v1/admin/directory/college/<str:slug>/offering-requirement/<uuid:requirement_id>/delete/
```

35. Of fering scholarship create

- POST
api/v1/admin/directory/college/<str:slug>/offering/<uuid:offering_id>/scholarship/create/
- POST
api/v1/admin/directory/college/<str:slug>/course/<str:course_slug>/scholarship/create/

2.8 Scholarship APIs (global)

36. Create (explicit scope)

```
POST api/v1/admin/directory/scholarship/create/
```

37. List

```
GET api/v1/admin/directory/scholarship/list/
```

Query:

- scope=university|course|offering

38. Update

```
PATCH api/v1/admin/directory/scholarship/<uuid:scholarship_id>/update/
```

39. Delete

```
DELETE api/v1/admin/directory/scholarship/<uuid:scholarship_id>/delete/
```

3) Public Directory APIs

Only published data is returned.

P1. Discipline list

```
GET api/v1/directory/discipline/list/
```

P2. University list

```
GET api/v1/directory/university/list/
```

Query:

- `search` , `institution_type` , `country` , `city` , `province` , `address` , `year_filters`

Card fields:

- `institution_type`
- `country` , `city` , `province` , `location`
- `short_overview` , `college_count`
- `logo` , `cover_image`

P3. University detail

```
GET api/v1/directory/university/<str:slug>/detail/
```

P4. Course list

```
GET api/v1/directory/course/list/
```

Query:

- `search` , `university` , `discipline` , `level`

P5. Course detail

```
GET api/v1/directory/course/<str:slug>/detail/
```

P5a. Related courses

```
GET api/v1/directory/course/<str:slug>/related-courses/
```

P6. Course affiliated colleges

```
GET api/v1/directory/course/<str:slug>/affiliated-colleges/
```

Query:

- `is_open`

P7. College list

```
GET api/v1/directory/college/list/
```

Query:

- `search`, `city`, `province`, `district`, `university`, `published`, `address`, `year` filters

P8. College detail

```
GET api/v1/directory/college/<str:slug>/detail/
```

P9. College affiliated courses

```
GET api/v1/directory/college/<str:slug>/affiliated-courses/
```

Query:

- `is_open`

P10. College offering detail

- `GET api/v1/directory/college/<str:slug>/course/<str:course_slug>/detail/`
(recommended)
- `GET api/v1/directory/college/<str:slug>/offering/<uuid:offering_id>/detail/`
(legacy)

Includes:

- of offering core fields
- course + university + college nested info
- requirements breakdown and merged requirements
- merged published scholarships (university + course + of offering)
- fee/admission fields

4) Gallery Handling Summary

Gallery in main create/update APIs (latest)

Handled directly inside serializer create/update for:

- University create/update
- College create/update

Payload key format:

- `gallery[0][id]` (optional existing image UUID)
- `gallery[0][image_file]` (required for new row)
- `gallery[0][caption]`
- `gallery[0][sort_order]`

Replace behavior:

- Only gallery rows sent in payload are kept.
- Existing rows not included are removed.

Dedicated University gallery API (admin)

Endpoint:

- `GET/PATCH api/v1/admin/directory/university/<str:slug>/gallery/update/`

Payload key format:

- `gallery[0][id]`
- `gallery[0][image_file]`
- `gallery[0][caption]`
- `gallery[0][sort_order]`

Replace behavior:

- Same replacement behavior as main create/update APIs.

5) Notes for frontend integration

- Use course-slug URL form for of `fring` detail/edit/delete where possible.
- For multipart forms with nested JSON arrays, stringify array fields where needed (`universities` , etc.).
- For university location display, use `country + city + province` fallback logic from API response `location`.