

TheKiranAcademy

Job Sphere API Workflow

The workflows are designed to:

- Cover all endpoints from the 12 controllers (CategoryController, CompanyController, EmployerProfileController, JobApplicationController, JobController, JobSeekerProfileController, JobSkillController, LocationController, NotificationController, SkillController, UserController, UserSkillController).

Employers Workflow

The Employers Workflow details the sequence of API calls for employers to interact with the Job Portal API, covering account setup, company and profile management, job posting, application review, and analytics. All endpoints are accessed at <http://localhost:8091/api> and can be tested using the Postman collection "Job Portal API." The steps are ordered to ensure dependencies are met (e.g., registering a user before creating a job). Each step includes the API, Postman request, and expected outcome.

Workflow Steps

- Register as an Employer**
 - Goal:** Create an employer account to access hiring features.
 - API:** `POST /api/users/register`
 - Controller:** `UserController`
 - Postman Request:** `Users > Register Employer`
 - Details:**
 - Body: `{ "email": "employer@example.com", "password": "password", "role": "EMPLOYER", "firstName": "John", "lastName": "Doe" }`
 - Note the `userId` (e.g., `2`) from the response.
 - Expected Outcome:** Employer account created with `role: EMPLOYER`.
- Log In as Employer**
 - Goal:** Authenticate to obtain user details.
 - API:** `POST /api/users/login`
 - Controller:** `UserController`
 - Postman Request:** `Users > Login Employer`
 - Details:**
 - Body: `{ "email": "employer@example.com", "password": "password" }`
 - Confirms `userId` (e.g., `2`) and `role`.
 - Expected Outcome:** User details returned, confirming authentication.
- Get User Details**
 - Goal:** Verify employer account details.
 - API:** `GET /api/users/{id}`
 - Controller:** `UserController`
 - Postman Request:** `Users > Get User`
 - Details:**
 - URL: `{{baseUrl}}/users/2`
 - Response includes `email`, `role`, etc.
 - Expected Outcome:** Employer details retrieved.
- Create Employer Profile**
 - Goal:** Set up a detailed employer profile.
 - API:** `POST /api/employer-profiles`
 - Controller:** `EmployerProfileController`
 - Postman Request:** `Employer Profiles > Create Profile`
 - Details:**
 - Body: `{ "userId": 2, "companyName": "Tech Corp", "contactInfo": "contact@techcorp.com" }`
 - Note the `profileId` (e.g., `1`).
 - Expected Outcome:** Employer profile created.
- Get Employer Profile**
 - Goal:** Verify the created profile.
 - API:** `GET /api/employer-profiles/{id}`
 - Controller:** `EmployerProfileController`
 - Postman Request:** `Employer Profiles > Get Profile`
 - Details:**
 - URL: `{{baseUrl}}/employer-profiles/1`
 - Expected Outcome:** Profile details returned.
- Update Employer Profile**
 - Goal:** Modify employer profile details.
 - API:** `PUT /api/employer-profiles/{id}`
 - Controller:** `EmployerProfileController`
 - Postman Request:** `Employer Profiles > Update Profile`
 - Details:**
 - URL: `{{baseUrl}}/employer-profiles/1`
 - Body: `{ "userId": 2, "companyName": "Tech Corp Inc.", "contactInfo": "newcontact@techcorp.com" }`
 - Expected Outcome:** Profile updated.
- Create a Company**
 - Goal:** Set up a company for job listings.
 - API:** `POST /api/companies`
 - Controller:** `CompanyController`
 - Postman Request:** `Companies > Create Company`
 - Details:**
 - Body: `{ "name": "Tech Corp", "description": "Innovative tech solutions", "website": "https://techcorp.com", "location": "New York" }`
 - Note the `companyId` (e.g., `1`).
 - Expected Outcome:** Company created.
- Get Company Details**
 - Goal:** Verify company details.
 - API:** `GET /api/companies/{id}`
 - Controller:** `CompanyController`
 - Postman Request:** `Companies > Get Company`
 - Details:**
 - URL: `{{baseUrl}}/companies/1`
 - Expected Outcome:** Company details returned.
- Update Company**
 - Goal:** Modify company information.
 - API:** `PUT /api/companies/{id}`
 - Controller:** `CompanyController`
 - Postman Request:** `Companies > Update Company`
 - Details:**
 - URL: `{{baseUrl}}/companies/1`
 - Body: `{ "name": "Tech Corp Inc.", "description": "Leading tech solutions", "website": "https://techcorp.com", "location": "New York" }`
 - Expected Outcome:** Company updated.
- Get All Categories**
 - Goal:** Retrieve available job categories.
 - API:** `GET /api/categories`
 - Controller:** `CategoryController`
 - Postman Request:** `Categories > Get All Categories`
 - Details:**
 - URL: `{{baseUrl}}/categories`
 - Note existing `categoryId` (e.g., `1`) or proceed to create one.
 - Expected Outcome:** List of categories returned.
- Create a Category (Optional)**
 - Goal:** Define a new job category if needed.
 - API:** `POST /api/categories`
 - Controller:** `CategoryController`
 - Postman Request:** `Categories > Create Category`
 - Details:**
 - Body: `{ "name": "Software Development" }`
 - Note the `categoryId` (e.g., `1`).
 - Expected Outcome:** Category created.
- Get Category Details**
 - Goal:** Verify category details.
 - API:** `GET /api/categories/{id}`
 - Controller:** `CategoryController`
 - Postman Request:** `Categories > Get Category`
 - Details:**
 - URL: `{{baseUrl}}/categories/1`
 - Expected Outcome:** Category details returned.
- Get All Locations**
 - Goal:** Retrieve available job locations.
 - API:** `GET /api/locations`
 - Controller:** `LocationController`
 - Postman Request:** `Locations > Get All Locations`
 - Details:**
 - URL: `{{baseUrl}}/locations`
 - Note existing `locationId` (e.g., `1`) or proceed to create one.
 - Expected Outcome:** List of locations returned.
- Create a Location (Optional)**
 - Goal:** Define a new job location if needed.
 - API:** `POST /api/locations`
 - Controller:** `LocationController`
 - Postman Request:** `Locations > Create Location`
 - Details:**
 - Body: `{ "city": "New York", "state": "NY", "country": "USA" }`
 - Note the `locationId` (e.g., `1`).
 - Expected Outcome:** Location created.
- Get Location Details**
 - Goal:** Verify location details.
 - API:** `GET /api/locations/{id}`
 - Controller:** `LocationController`
 - Postman Request:** `Locations > Get Location`
 - Details:**
 - URL: `{{baseUrl}}/locations/1`
 - Expected Outcome:** Location details returned.
- Get All Skills**
 - Goal:** Retrieve available skills for job requirements.
 - API:** `GET /api/skills`
 - Controller:** `SkillController`
 - Postman Request:** `Skills > Get All Skills`
 - Details:**
 - URL: `{{baseUrl}}/skills`
 - Note existing skill IDs or names.
 - Expected Outcome:** List of skills returned.
- Create a Skill (Optional, Manual)**
 - Goal:** Add a skill if not using **Bulk Skill Upload**.
 - API:** `POST /api/skills`
 - Controller:** `SkillController`
 - Postman Request:** `Skills > Create Skill`
 - Details:**
 - Body: `{ "name": "Java" }`
 - Note the `skillId` (e.g., `1`).
 - Expected Outcome:** Skill created.
- Get Skill Details**
 - Goal:** Verify skill details.
 - API:** `GET /api/skills/{id}`
 - Controller:** `SkillController`
 - Postman Request:** `Skills > Get Skill`
 - Details:**
 - URL: `{{baseUrl}}/skills/1`
 - Expected Outcome:** Skill details returned.
- Create a Job**
 - Goal:** Post a job listing with skill names.
 - API:** `POST /api/jobs`
 - Controller:** `JobController`
 - Postman Request:** `Jobs > Create Job`
 - Details:**
 - Body:
 - Note the `jobId` (e.g., `1`).
 - Expected Outcome:** Job created, new skills added via **Bulk Skill Upload**.
- Add Skill to Job (Optional, Manual)**
 - Goal:** Associate an existing skill with a job if not using **Bulk Skill Upload**.
 - API:** `POST /api/job-skills`
 - Controller:** `JobSkillController`
 - Postman Request:** `Job Skills > Add Skill to Job`
 - Details:**
 - Body: `{ "jobId": 1, "skillId": 1 }`
 - Expected Outcome:** Skill linked to job.
- Get Job Details**
 - Goal:** Verify job details.
 - API:** `GET /api/jobs/{id}`
 - Controller:** `JobController`
 - Postman Request:** `Jobs > Get Job`
 - Details:**
 - URL: `{{baseUrl}}/jobs/?userId=2`
 - Expected Outcome:** Job details returned, view logged.
- Get All Jobs**
 - Goal:** View all posted jobs.
 - API:** `GET /api/jobs`
 - Controller:** `JobController`
 - Postman Request:** `Jobs > Get All Jobs`
 - Details:**
 - Query: `?page=0&size=10`
 - Expected Outcome:** Paginated list of jobs.
- Search Jobs**
 - Goal:** Find jobs by criteria (optional for employers).
 - API:** `GET /api/jobs/search`
 - Controller:** `JobController`
 - Postman Request:** `Jobs > Search Jobs`
 - Details:**
 - Query: `?keyword=Software&categoryId=1&locationId=1&page=0&size=10`
 - Expected Outcome:** Paginated list of matching jobs.
- Get Job Analytics**
 - Goal:** Analyze job performance (views, applications).
 - API:** `GET /api/jobs/{id}/analytics`
 - Controller:** `JobController`
 - Postman Request:** `Jobs > Get Job Analytics`
 - Details:**
 - URL: `{{baseUrl}}/jobs/1/analytics`
 - Response: `{ "jobId": 1, "views": 50, "applications": 10 }`
 - Expected Outcome:** Analytics data returned.
- Get Applications for a Job**
 - Goal:** Review applications for a job.
 - API:** `GET /api/applications?jobId={jobId}`
 - Controller:** `JobApplicationController`
 - Postman Request:** `Job Applications > Get Applications by Job`
 - Details:**
 - URL: `{{baseUrl}}/applications?jobId=1`
 - Note `applicationId` (e.g., `1`).
 - Expected Outcome:** List of applications.
- Get Application Details**
 - Goal:** View details of a specific application.
 - API:** `GET /api/applications/{id}`
 - Controller:** `JobApplicationController`
 - Postman Request:** `Job Applications > Get Application`
 - Details:**
 - URL: `{{baseUrl}}/applications/1`
 - Expected Outcome:** Application details, including comments.
- Update Application Status with Comment**
 - Goal:** Update application status and add a comment.
 - API:** `PUT /api/applications/{id}/status`
 - Controller:** `JobApplicationController`
 - Postman Request:** `Job Applications > Update Application Status with Comment`
 - Details:**
 - URL: `{{baseUrl}}/applications/1/status?status=ACCEPTED&comment=Strong%20candidate`
 - Expected Outcome:** Status updated, comment saved, notification sent.
- Get All Employers**
 - Goal:** View all employer accounts (optional, administrative).
 - API:** `GET /api/users/all/employers`
 - Controller:** `UserController`
 - Postman Request:** `Users > Get All Employers`
 - Details:**
 - URL: `{{baseUrl}}/users/all/employers`
 - Expected Outcome:** List of employers returned.
- Get All Users**
 - Goal:** View all users (optional, administrative).
 - API:** `GET /api/users/all`
 - Controller:** `UserController`
 - Postman Request:** `Users > Get All Users`
 - Details:**
 - URL: `{{baseUrl}}/users/all`
 - Expected Outcome:** List of all users returned.
- Update User**
 - Goal:** Modify employer account details.
 - API:** `PUT /api/users/{id}`
 - Controller:** `UserController`
 - Postman Request:** `Users > Update User`
 - Details:**
 - URL: `{{baseUrl}}/users/2`
 - Body: `{ "email": "newemployer@example.com", "password": "newpassword", "role": "EMPLOYER", "firstName": "John", "lastName": "Doe" }`
 - Expected Outcome:** User details updated.
- Delete Employer Profile (Optional)**
 - Goal:** Remove employer profile if needed.
 - API:** `DELETE /api/employer-profiles/{id}`
 - Controller:** `EmployerProfileController`
 - Postman Request:** `Employer Profiles > Delete Profile`
 - Details:**
 - URL: `{{baseUrl}}/employer-profiles/1`
 - Expected Outcome:** Profile deleted.
- Delete Company (Optional)**
 - Goal:** Remove company if needed.
 - API:** `DELETE /api/companies/{id}`
 - Controller:** `CompanyController`
 - Postman Request:** `Companies > Delete Company`
 - Details:**
 - URL: `{{baseUrl}}/companies/1`
 - Expected Outcome:** Company deleted.
- Delete User (Optional)**
 - Goal:** Delete employer account if needed.
 - API:** `DELETE /api/users/{id}`
 - Controller:** `UserController`
 - Postman Request:** `Users > Delete User`
 - Details:**
 - URL: `{{baseUrl}}/users/2`
 - Expected Outcome:** User account deleted.

Notes

- Sequence:** Steps are ordered to satisfy dependencies (e.g., user registration before profile creation, company creation before job posting).
- Optional Steps:** Steps 11, 14, 17, 20, 31–33 are optional, as **Bulk Skill Upload** reduces the need for manual skill creation, and deletion is situational.
- Bulk Skill Upload:** Step 19 uses `skillNames` in `POST /api/jobs`, automatically creating new skills.
- Analytics:** Step 24 leverages **Employer Dashboard Analytics**.
- Comments:** Step 27 includes **Application Review Comments**.
- Administrative Actions:** Steps 28–29 are optional for employers with administrative access.

Job Seekers Workflow

The Job Seekers Workflow details the sequence of API calls for job seekers to interact with the Job Portal API, covering account setup, profile management, job search, applications, and notifications. All endpoints are accessed at `http://localhost:8091/api` and can be tested using the Postman collection "Job Portal API." The steps are ordered to ensure dependencies are met (e.g., registering before applying). Each step includes the API, Postman request, and expected outcome.

Workflow Steps

1. Register as a Job Seeker

Goal: Create a job seeker account to apply for jobs.

API: `POST /api/users/register`

Controller: `UserController`

Postman Request: `Users > Register Job Seeker`

Details:

Body: `{"email": "seeker@example.com", "password": "password", "role": "JOB_SEEKER", "firstName": "Jane", "lastName": "Smith"}`

Note the `userId` (e.g., `1`) from the response.

Expected Outcome: Job seeker account created with role: `JOB_SEEKER`.

2. Log In as Job Seeker

Goal: Authenticate to obtain user details.

API: `POST /api/users/login`

Controller: `UserController`

Postman Request: `Users > Login Job Seeker`

Details:

Body: `{"email": "seeker@example.com", "password": "password"}`

Confirms `userId` (e.g., `1`) and role.

Expected Outcome: User details returned, confirming authentication.

3. Get User Details

Goal: Verify job seeker account details.

API: `GET /api/users/{id}`

Controller: `UserController`

Postman Request: `Users > Get User`

Details:

URL: `{{baseUrl}}/users/1`

Response includes `email`, `role`, etc.

Expected Outcome: Job seeker details retrieved.

4. Create Job Seeker Profile

Goal: Set up a detailed job seeker profile.

API: `POST /api/job-seeker-profiles`

Controller: `JobSeekerProfileController`

Postman Request: `Job Seeker Profiles > Create Profile`

Details:

Body: `{"userId": 1, "resume": "resume.pdf", "skills": ["Java"], "experience": "2 years"}`

Note the `profileId` (e.g., `1`).

Expected Outcome: Job seeker profile created.

5. Get Job Seeker Profile

Goal: Verify the created profile.

API: `GET /api/job-seeker-profiles/{id}`

Controller: `JobSeekerProfileController`

Postman Request: `Job Seeker Profiles > Get Profile`

Details:

URL: `{{baseUrl}}/job-seeker-profiles/1`

Expected Outcome: Profile details returned.

6. Update Job Seeker Profile

Goal: Modify profile details.

API: `PUT /api/job-seeker-profiles/{id}`

Controller: `JobSeekerProfileController`

Postman Request: `Job Seeker Profiles > Update Profile`

Details:

URL: `{{baseUrl}}/job-seeker-profiles/1`

Body: `{"userId": 1, "resume": "updated_resume.pdf", "skills": ["Java", "Python"], "experience": "3 years"}`

Expected Outcome: Profile updated.

7. Add Skill to User

Goal: Associate skills with the job seeker for **Recommended Jobs**.

API: `POST /api/user-skills`

Controller: `UserSkillController`

Postman Request: `User Skills > Add Skill to User`

Details:

Body: `{"userId": 1, "skillId": 1}`

Repeat for additional skills (e.g., `skillId: 2` for Python).

Expected Outcome: Skill linked to user.

8. Get All Skills

Goal: View available skills for profile or job search.

API: `GET /api/skills`

Controller: `SkillController`

Postman Request: `Skills > Get All Skills`

Details:

URL: `{{baseUrl}}/skills`

Expected Outcome: List of skills returned.

9. Get Skill Details

Goal: Verify specific skill details.

API: `GET /api/skills/{id}`

Controller: `SkillController`

Postman Request: `Skills > Get Skill`

Details:

URL: `{{baseUrl}}/skills/1`

Expected Outcome: Skill details returned.

10. Get All Categories

Goal: View job categories for search.

API: `GET /api/categories`

Controller: `CategoryController`

Postman Request: `Categories > Get All Categories`

Details:

URL: `{{baseUrl}}/categories`

Expected Outcome: List of categories returned.

11. Get Category Details

Goal: Verify category details.

API: `GET /api/categories/{id}`

Controller: `CategoryController`

Postman Request: `Categories > Get Category`

Details:

URL: `{{baseUrl}}/categories/1`

Expected Outcome: Category details returned.

12. Get All Locations

Goal: View job locations for search.

API: `GET /api/locations`

Controller: `LocationController`

Postman Request: `Locations > Get All Locations`

Details:

URL: `{{baseUrl}}/locations`

Expected Outcome: List of locations returned.

13. Get Location Details

Goal: Verify location details.

API: `GET /api/locations/{id}`

Controller: `LocationController`

Postman Request: `Locations > Get Location`

Details:

URL: `{{baseUrl}}/locations/1`

Expected Outcome: Location details returned.

14. Get All Jobs

Goal: Browse all available jobs.

API: `GET /api/jobs`

Controller: `JobController`

Postman Request: `Jobs > Get All Jobs`

Details:

Query: `?page=0&size=10`

Expected Outcome: Paginated list of jobs.

15. Search Jobs

Goal: Find jobs by criteria.

API: `GET /api/jobs/search`

Controller: `JobController`

Postman Request: `Jobs > Search Jobs`

Details:

Query: `?keyword=Software&categoryId=1&locationId=1&page=0&size=10`

Note `jobId` (e.g., `1`) for application.

Expected Outcome: Paginated list of matching jobs.

16. Get Recommended Jobs

Goal: Discover jobs matching user skills.

API: `GET /api/jobs/recommended`

Controller: `JobController`

Postman Request: `Jobs > Get Recommended Jobs`

Details:

Query: `?userId=1&page=0&size=10`

Expected Outcome: Paginated list of recommended jobs.

17. Get Job Details

Goal: Review a specific job.

API: `GET /api/jobs/{id}`

Controller: `JobController`

Postman Request: `Jobs > Get Job`

Details:

URL: `{{baseUrl}}/jobs/1?userId=1`

Logs a view for **Employer Dashboard Analytics**.

Expected Outcome: Job details returned, view recorded.

18. Apply for a Job

Goal: Submit a job application.

API: `POST /api/applications`

Controller: `JobApplicationController`

Postman Request: `Job Applications > Create Application`

Details:

Body: `{"jobId": 1, "userId": 1, "coverLetter": "I am excited to apply..."}`

Note the `applicationId` (e.g., `1`).

Expected Outcome: Application submitted with status `PENDING`.

19. Get Application Details

Goal: View a specific application.

API: `GET /api/applications/{id}`

Controller: `JobApplicationController`

Postman Request: `Job Applications > Get Application`

Details:

URL: `{{baseUrl}}/applications/1`

Expected Outcome: Application details, including status and comments.

20. Get Applications by User

Goal: Track all submitted applications.

API: `GET /api/applications?userId={userId}`

Controller: `JobApplicationController`

Postman Request: `Job Applications > Get Applications by User`

Details:

URL: `{{baseUrl}}/applications?userId=1`

Expected Outcome: List of user's applications.

21. Get User Notifications

Goal: Check application status updates and comments.

API: `GET /api/notifications`

Controller: `NotificationController`

Postman Request: `Notifications > Get User Notifications`

Details:

Query: `?userId=1`

Response includes notifications, e.g., "Your application for Software Engineer has been **ACCEPTED**. Comment: Strong candidate."

Expected Outcome: Notifications retrieved.

22. Get All Job Seekers

Goal: View all job seeker accounts (optional, administrative).

API: `GET /api/users/all/jobseekers`

Controller: `UserController`

Postman Request: `Users > Get All Job Seekers`

Details:

URL: `{{baseUrl}}/users/all/jobseekers`

Expected Outcome: List of job seekers returned.

23. Get All Users

Goal: View all users (optional, administrative).

API: `GET /api/users/all`

Controller: `UserController`

Postman Request: `Users > Get All Users`

Details:

URL: `{{baseUrl}}/users/all`

Expected Outcome: List of all users returned.

24. Update User

Goal: Modify job seeker account details.

API: `PUT /api/users/{id}`

Controller: `UserController`

Postman Request: `Users > Update User`

Details:

URL: `{{baseUrl}}/users/1`

Body: `{"email": "newseeker@example.com", "password": "newpassword", "role": "JOB_SEEKER", "firstName": "Jane", "lastName": "Smith"}`

Expected Outcome: User details updated.

25. Delete Job Seeker Profile (Optional)

Goal: Remove job seeker profile if needed.

API: `DELETE /api/job-seeker-profiles/{id}`

Controller: `JobSeekerProfileController`

Postman Request: `Job Seeker Profiles > Delete Profile`

Details:

URL: `{{baseUrl}}/job-seeker-profiles/1`

Expected Outcome: Profile deleted.

26. Delete User (Optional)

Goal: Delete job seeker account if needed.

API: `DELETE /api/users/{id}`

Controller: `UserController`

Postman Request: `Users > Delete User`

Details:

URL: `{{baseUrl}}/users/1`

Expected Outcome: User account deleted.

Notes

- Sequence:** Steps are ordered to satisfy dependencies (e.g., profile creation before adding skills, job search before applying).
 - Optional Steps:** Steps 22–26 are optional, typically for administrative or cleanup purposes.
 - Skills:** Step 7 enhances **Recommended Jobs** by linking skills to the user.
 - Notifications:** Step 21 reflects **Application Status Notifications** and **Application Review Comments**.
 - Analytics:** Step 17 logs views for **Employer Dashboard Analytics**.

Made with **GAMMA**