

Act I: The Foundation - Users, Auth, and Admin Oversight

Narrator: "Welcome. We'll begin by establishing the foundational user accounts for our platform: a Client, a Freelancer, and we'll then log in as an Administrator to see how they can manage the system."

Step	Operator Action (in Postman)	Narrator's Script
1	Auth > Register New Client: Run.	"First, a new client registers. The API validates the input and creates the account."
2	Auth > Register New Freelancer: Run.	"Next, a freelancer joins the platform through the same registration process."
3	Auth > Login as Admin: Run.	"Now, we'll log in as an Administrator. Watch the collection variables: our script automatically saves the Admin's JWT token for all future requests."
4	Users > Get All Users (Admin): Run.	"As an Admin, we have a complete overview and can retrieve a paginated list of all users."
5	Users > Get User by ID (Client): Run.	"The Admin can drill down to view the specific details of any user, like our newly created client. We'll save this client's ID for later."
6	Auth > Login as Client: Run.	"Let's switch roles. We're now logged in as the client. The JWT token has been automatically updated."
7	Users > Update User (Self): Run.	"A user has full control over their own profile details, such as updating their name."

8	<code>Users > Change Password (Self): Run.</code>	"And they can securely change their own password without involving an administrator."
---	------------------------------------------------------	---------------------------------------------------------------------------------------

Act II: Building Professional Identities

Narrator: "With accounts created, our users need to build their professional profiles. This involves creating profiles and managing the skills that define their work."

Step	Operator Action (in Postman)	Narrator's Script
9	<code>Auth > Login as Admin: Run.</code>	"First, the Admin curates the skills available on the platform."
10	<code>Skills > Create Skill (Admin): Run.</code>	"Let's add a new skill called 'Postman Demo Skill'. The API returns the newly created skill, and we'll save its ID."
11	<code>Skills > Get All Skills: Run.</code>	"We can then retrieve the entire, filterable list of skills in the system."
12	<code>Skills > Get Skill by ID: Run.</code>	"And, of course, fetch any specific skill by its unique ID."
13	<code>Auth > Login as Client: Run.</code>	"Switching back to our client, they can now create their company profile."
14	<code>Client Profiles > Create Client Profile: Run.</code>	"The profile is created and linked to their user account. We'll save the new profile ID."
15	<code>Client Profiles > Get My Client Profile: Run.</code>	"A dedicated endpoint allows users to easily fetch their own profile."
16	<code>Client Profiles > Update Client Profile: Run.</code>	"And they can update their details at any time."

17	Auth > Login as Freelancer: Run.	"Now for the freelancer. After logging in, they'll create their profile."
18	Freelancer Profiles > Create Freelancer Profile: Run.	"The freelancer populates their profile with a headline, bio, and associates it with the skills we created earlier."
19	Freelancer Profiles > Get My Freelancer Profile: Run.	"Just like the client, they have a dedicated endpoint to view their own profile."
20	Freelancer Profiles > Update Freelancer Profile: Run.	"And they can update it to keep it fresh."
21	Freelancer Profiles > Upload Profile Picture: Run.	"Our API handles multipart file uploads, allowing the freelancer to upload a profile picture."
22	Freelancer Profiles > Upload Resume: Run.	"And a resume, both of which are critical for their identity on the platform."
23	Auth > Login as Client: Run.	"Let's log back in as the client to see what they can view."
24	Client Profiles > Get Client Profile by ID: Run.	"They can view other public profiles, including other clients."
25	Freelancer Profiles > Get Freelancer Profile by ID: Run.	"And more importantly, they can view our freelancer's newly created profile."
26	Freelancer Profiles > Get All Freelancer Profiles: Run.	"They can also browse and filter through the entire marketplace of available freelancers."

Act III: The Core Business Logic - Projects, Proposals, and Files

Narrator: "Now we get to the heart of the platform: a client creates a project, and a freelancer applies for it."

Step	Operator Action (in Postman)	Narrator's Script
27	Auth > Login as Client: Run.	"Our client is ready to post a job."
28	Projects > Create Project: Run.	"They create a new project, defining the scope, budget, and required skills. We save the new Project ID."
29	Projects > Upload Project Attachment: Run.	"To provide more detail, they upload a project brief as an attachment. We save the Attachment ID."
30	Auth > Login as Freelancer: Run.	"The freelancer logs in to find work."
31	Projects > Get All Projects: Run.	"They browse the project listings and find the new project from our client."
32	Projects > Get Project Attachments: Run.	"They can view the list of attachments for the project..."
33	Projects > Download Project Attachment: Run.	"...and download the project brief to review the requirements."
34	Proposals > Create Proposal: Run.	"Impressed, they submit a proposal with a cover letter and budget. We save the Proposal ID."
35	Auth > Login as Client: Run.	"The client logs back in to see if they've received any proposals."

36	Proposals > Get Proposals For Project: Run.	"They can view all proposals submitted for their specific project."
37	Proposals > Get Proposal by ID: Run.	"And they can drill down to see the details of our freelancer's proposal."
38	Proposals > Update Proposal Status: Run to "Accept" it.	"The client accepts the proposal. This triggers a critical state change in our system."
39	Projects > Get Project by ID: Run.	"If we now re-fetch the project, we'll see its status has automatically been updated to 'Assigned' and the freelancer's ID is now linked."
40	Projects > Mark Project Completed: Run.	"After the work is notionally completed, the client marks the project as finished."

Act IV: Resolution & Reputation - Ratings

Narrator: "With the project complete, the final step in the workflow is the two-way rating system, which builds trust and reputation on the platform."

Step	Operator Action (in Postman)	Narrator's Script
41	Auth > Login as Client: Run.	"The client will now rate the freelancer's performance."
42	Ratings > Create Rating (Client rates Freelancer): Run.	"They submit a 5-star rating with a positive comment. We save the Rating ID."
43	Auth > Login as Freelancer: Run.	"Now the freelancer logs in to provide their feedback on the client."

44	Proposals > Get Proposals By Freelancer: Run.	"A freelancer can always review a list of all proposals they have submitted."
45	Ratings > Get Rating by ID: Run.	"Anyone involved can view the details of a specific rating."
46	Ratings > Get Ratings Received By User: Run for the Freelancer.	"We can easily query for all ratings a specific user has received."
47	Ratings > Get Ratings Given By User: Run for the Client.	"Or all the ratings they have given."
48	Ratings > Get Average Rating For User: Run for the Freelancer.	"And crucially, the API can calculate the average rating for any user, which is their public reputation score."

Act V: Security, Housekeeping, and Cleanup

Narrator: "Finally, we'll demonstrate the robustness of our security and the comprehensive cleanup capabilities available to users and administrators."

Step	Operator Action (in Postman)	Narrator's Script
49	Auth > Login as Freelancer: Run.	"Let's prove our role-based security works. We are logged in as a freelancer."
50	Projects > Create Project: Run.	"When the freelancer tries to create a project—a client-only action—the API correctly responds with a 403 <code>Forbidden</code> error."
51	Files > Get File Metadata: Run.	"Our API also provides generic file management endpoints. Here we can fetch the metadata for the attachment we uploaded earlier."

52	Files > Download File: Run.	"And download it again, independent of the project context."
53	Auth > Login as Admin: Run.	"Now for the final cleanup, performed by the Admin."
54	Files > Delete File (Admin): Run.	"The Admin can delete any file in the system."
55	Ratings > Delete Rating (Admin): Run.	"They can moderate and remove ratings."
56	Proposals > Delete Proposal: Run.	"They can remove proposals."
57	Projects > Delete Project (Admin): Run.	"And entire projects."
58	Client Profiles > Delete Client Profile (Admin): Run.	"They have the power to delete user profiles."
59	Freelancer Profiles > Delete Freelancer Profile (Admin): Run.	"Including freelancer profiles."
60	Skills > Delete Skill (Admin): Run.	"They can curate the skills list by removing entries."
61	Users > Delete User (Admin): Run.	"And finally, they can delete user accounts, completing the administrative lifecycle."

Narrator (Conclusion): "This concludes our comprehensive tour. We have successfully demonstrated all 61 endpoints, showcasing a complete, secure, and logical workflow from user creation to project completion and system administration. Thank you."