Part 1: Permissions and automation testing

Imagine that you've recently met with a potential customer who is interested in improving their API development process with Postman, she follows up after the meeting with an email that includes the following questions. How would you answer her?

1. I am looking into the possibility of using Postman in my team. From my initial investigations, it looks like the content created within my Postman team is visible and editable to anyone on the team. This is great, but concerns me from a workflow/compliance perspective. Can you tell me how Postman handles this?

2. How can Postman help me with testing automation?

Discovery

Value - Needs team based collaborative tool, permissions are important, automated testing, email support
Economic buyer - Unidentified
Decision process - Unidentified
Decision criteria - Unidentified
Identify pain - Compliance issues, workflow problems, standardization, security, test automation
Competition - Swaggerhub, Spotlight, Insomnia, ReadyAPI, RESTassured, JMeter
Champion - Sue
Urgency - Unidentified

Response notes

QUESTION 1
SSO only available through Professional and Enterprise plans
- Duo

SCIM only available in enterprise
- Okta
- Azure AD

Integrations with collaboration tools:
- Github
- Gitlab
- Slack

Teams have defined roles for different access:
- Super Admin (enterprise only)
- Admin
- Billing
- Developer
- Community manager (pro and enter)

Partner roles also available for outside the team with enterprise ultimate
- Partner manager (internal)
- Partner
- Guest

- Flow editor

Workspaces have 3 roles
- Admin
- Editor (Pro and Enter)
- Viewer (Pro and Enter)

Built in version control
Privacy and security

Question 2
Test creation with Javascript
Execution order including pre request scripts
Postman flows
Create dynamic mock responses
CI integration
Collection runner can run collections with external data
Scheduling collection runs
Newman CLI
Automated observability with recent Akita acquisition

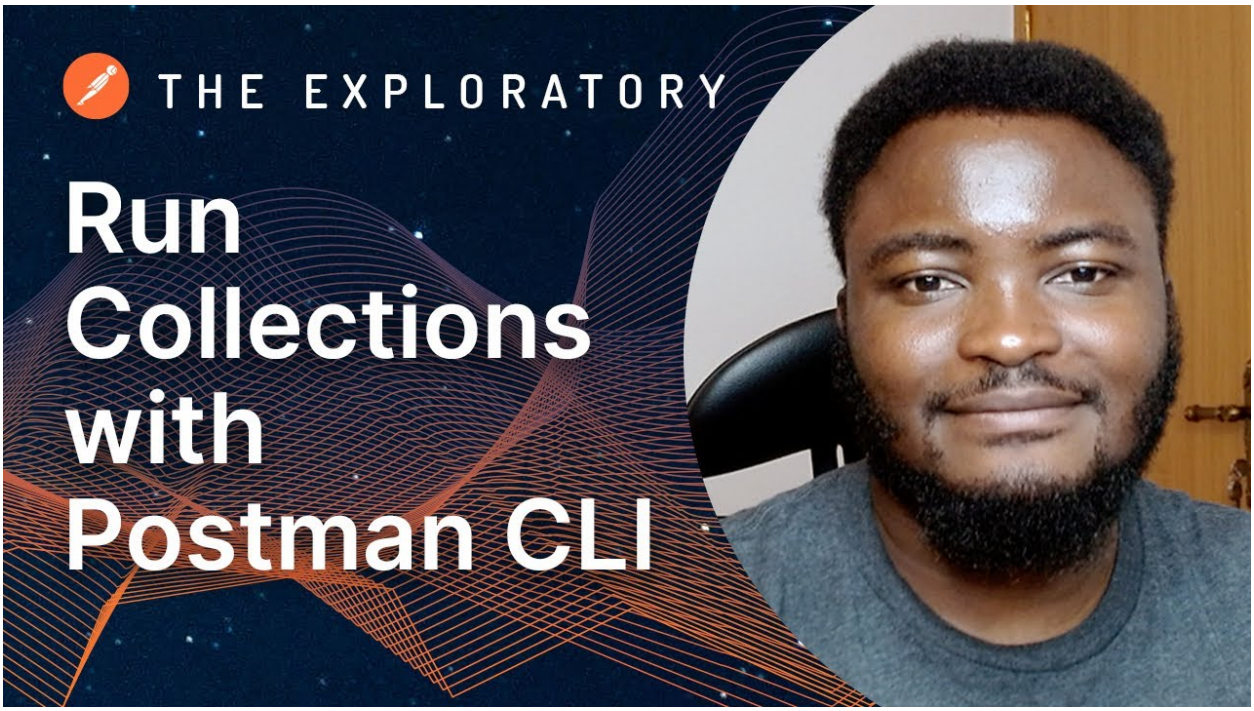
Sources

https://learning.postman.com/docs/collaborating-in-postman/working-with-your-team/enabling-team-discovery/
https://learning.postman.com/docs/collaborating-in-postman/roles-and-permissions/
https://learning.postman.com/docs/collaborating-in-postman/using-workspaces/managing-workspaces/#managing-workspace-roles
https://learning.postman.com/docs/administration/scim-provisioning/scim-provisioning-overview/
https://www.postman.com/trust/security/
https://learning.postman.com/docs/designing-and-developing-your-api/versioning-an-api/versioning-an-api-overview/#connecting-to-a-remote-git-repository
https://learning.postman.com/docs/designing-and-developing-your-api/versioning-an-api/versioning-an-api-overview/
https://learning.postman.com/docs/collaborating-in-postman/using-version-control/version-control-overview/

https://learning.postman.com/docs/writing-scripts/intro-to-scripts/
https://learning.postman.com/docs/writing-scripts/pre-request-scripts/
https://learning.postman.com/docs/writing-scripts/test-scripts/
https://learning.postman.com/docs/postman-flows/gs/flows-overview/
https://learning.postman.com/docs/designing-and-developing-your-api/mocking-data/creating-dynamic-responses/
https://learning.postman.com/docs/integrations/ci-integrations/
https://learning.postman.com/docs/collections/running-collections/working-with-data-files/
https://learning.postman.com/docs/collections/running-collections/scheduling-collection-runs/

https://learning.postman.com/docs/collections/using-newman-cli/command-line-integration-with-newman/
https://blog.postman.com/postman-acquires-akita-for-automated-api-observability/

Video Resources



Postman intergalactic events

https://www.postman.com/events/intergalactic/collaboration-and-governance-for-api-teams/
https://www.postman.com/events/intergalactic/
https://www.postman.com/events/intergalactic/advanced-api-testing-best-practices-and-automation-techniques/

Email draft

Hello Sue,

It was a pleasure meeting with you earlier. Some of the challenges you are facing are valid concerns that many teams grapple with, and it's only natural to seek solutions that not only address these challenges but also seamlessly integrate with your team's unique workflow. I'm glad to address your concerns regarding content visibility and editability within a Postman team, as well as share how Postman can assist your team with test automation.

You are correct that Postman allows team members to collaborate and share content within the team workspace. However, Postman provides robust access control and collaboration features to ensure that your team's content remains secure and compliant with your workflow requirements. Here's how Postman handles this (each topic contains a link to detailed documentation around each topic):

1. Team Roles and Permissions:
   - Our distinct team roles let you precisely control who can do what. This structure can greatly alleviate workflow problems by ensuring that everyone knows their specific responsibilities and access levels. The roles include Admin, Billing, and Developer. Additional roles of Super Admin and Community Manager are also available with Enterprise or Pro plans. This hierarchical approach helps you control who can make changes to your team's content.

2. SSO & SCIM Integrations:
   - Single Sign-On is available through our Professional and Enterprise plans. This functionality ensures that only authenticated individuals can access your Postman environment, offering you peace of mind regarding unauthorized access and potential compliance breaches. Additionally, SCIM is available exclusively in the Enterprise plan, further strengthening your control over identity and user provisioning.

3. Workspace and Collection Permissions:
   - For a more granulated control, roles within workspaces are designed to ensure that while collaboration is easy, not everyone can make changes unless explicitly authorized. This level of granularity enables you to manage content visibility based on project requirements and team roles. You can also create private workspace that are only accessible by invited team members.

4. Built-In Version Control and Audit History:
   - We understand that maintaining consistency and trackability in your API development process is paramount. With built-in version control, you have an overview of all changes, allowing for better compliance and standardization.

5. Integration with Collaboration Tools:
   - To further streamline your workflow, Postman integrates with popular collaboration tools like Github, Gitlab, Slack, and more. This will allow you to manage access, monitor changes, and get real-time notifications about your API development activities.

As for your question around testing and automation. These are integral aspects of Postman, and I'm glad you brought this up. Given your emphasis on compliance and the need for test automation, leveraging Postman's testing capabilities can not only streamline your processes but also offer added confidence in the robustness and reliability of your APIs.

1. Execution Order & Pre-Request Scripts:
   - When you make a request in Postman, it executes pre-request scripts first (useful for setting up environment variables, generating data, etc.) followed by the request itself, and then the tests. This ensures a systematic flow and preparation before any test is actually executed.

2. Postman Flows:
   - This allows users to set conditional workflows based on the response of a previous request. It's perfect for mimicking real-world scenarios where the output of one API call might determine the input or behavior of another.

3. Dynamic Mock Responses:
   - With Postman, you can create dynamic mock servers. These servers return responses based on the input request, allowing for a richer simulation of real-world API interactions and aiding in early-stage development and testing.

4. CI Integration:
   - Postman can be integrated into your Continuous Integration (CI) pipeline. This ensures that your APIs are automatically tested each time you commit changes, helping catch regressions and issues early.

5. Collection Runner with External Data:
   - The Collection Runner isn't limited to just the data within Postman. It can pull in external data in formats like JSON and CSV, allowing for data-driven tests. This is invaluable when you want to test the same API request with numerous data sets.

6. Scheduling Collection Runs:
   - Automated tests don't need manual initiation. With Postman Monitors, you can schedule collection runs, ensuring your APIs are validated at regular intervals.

7. Newman CLI:
   - Newman is Postman's command-line companion. It lets you run collections directly from the command line, which is immensely useful for integrating Postman tests into existing CI/CD pipelines or other automation tools.

8. Automated Observability with Akita:
   - With the recent acquisition of Akita, Postman enhances its automated observability capabilities. Akita provides insights into API behavior, detects anomalies, and offers a clearer view into the intricacies of API interactions. This can be vital for understanding and validating complex API scenarios.

In summary, while content created within a Postman team is indeed visible and editable to team members, you have full control over who can access, modify, and view the content. By

leveraging roles, permissions, version control, and private workspaces, you can tailor your team's collaboration experience to meet your workflow and compliance requirements.

Also, given your emphasis on automation, Postman's robust suite of features directly aligns with your requirements. It provides a cohesive and powerful environment to not only test but also observe and understand your APIs better.

If you have any further questions or if you would like a hands-on demonstration or deeper insights into any specific feature, I'm here to help. Please don't hesitate to reach out. We're here to support you in making the most of Postman for your API development needs.

Best regards,

Corey Vernon
Strategic Solutions Engineer
Postman