

Set up your first routine in 10 minutes.

Build a routine that categorizes pending tasks by schedule, running on weekdays at 9 AM. This lesson covers the routine's user interface, scheduling, and the biggest mistakes beginners often make.

Build a routine that categorizes pending tasks by schedule, running on weekdays at 9 AM. This lesson covers the routine's user interface, scheduling, and the biggest mistakes beginners often make.

Target

After completing this lesson, you will have a real routine in your account, running automatically on weekdays. The routine will read your team's backlog of tasks on Linear, flag any outdated tasks, and write you a brief summary of "things to consider today".

Nothing complicated. No API triggers, no GitHub events. Just a scheduled task that saves you time in the morning that shouldn't be done manually.

Let's start building a routine!

Step 1: Open the Routines interface.

In the desktop version of Claude Code or Claude.com, look for the **Routines** section . If you are using the Pro, Max, Team, or Enterprise version, this section will appear in the left sidebar.

You will see three things:

1. **New routine** button
2. List of routines you have created (can be blank)
3. **The Usage** table shows today's limit and the amount you have used.

The Usage table is very important. Record your daily limits (Pro = 5, Max = 15, Team/Enterprise = 25). Each routine run will count towards these limits.

Step 2: Configure triggers

Click on **New routine** . First option: **trigger type** .

Select **Schedule** . You will see a cron-style interface. For this first routine, set it for weekdays at 9 AM in your time zone.

Scheduling syntax:

1. **Weekdays** - Monday to Friday
2. **9:00 AM** - according to your local time zone
3. **Run duration** - maximum 10 minutes (default)

You can change this later. For now, it's fine.

Step 3: Choose the connector - Be careful!

This is where beginners often get stuck. The routine form will show you all the MCP connectors you've configured in your Claude account, and it will automatically activate them all.

This is an accurate quote from the Anthropic document:

All your connected MCP connectors are included by default... Please remove any connectors that your routine doesn't need.

Read this twice. That's the most common mistake in the first week.

For this backlog sorting routine, you only need one connector: Linear (or Jira, GitHub Issues, or whatever you use for request tickets). Deselect everything else.

Why is this important?

1. Speed ??- each connector adds tools that the model must consider.
2. Cost - more tools mean a longer context
3. Security - a routine with Notion + Gmail + Slack + your database is a much larger attack surface if your prompt crashes.
4. Debugging - fewer connectors mean fewer places for things to go wrong.

Think of the connector range as minimum IAM permissions. Only enable what you need.

Step 4: Write the prompt

Now for the prompt. This is the template you will use:

Bạn là bot phân loại công việc của bạn đang giúp sáng cho nhóm của tôi. Mỗi ngày trong tuần lúc 9 giờ sáng, hãy thực hiện các bước sau: 1. Truy vấn Linear để tìm tất cả các vấn đề chưa được giao cho tôi có: - Trạng thái: "Đang tiến hành" - Cập nhật lần cuối: hơn 3 ngày làm việc trước - Mức ưu tiên: Trung bình trở lên 2. Đọc và hiểu vấn đề của bạn đang, hãy viết một câu trả lời: - Nội dung vấn đề - Nguyên nhân có thể gây ra vấn đề - Hành động tiếp theo theo các thực hiện 3. Tôi cần tóm tắt câu chuyện theo đúng định dạng này: ## Số lượng vé của bạn hôm nay ({count}) ### {issue-id}: {title} - **Nguyên nhân dự đoán:** {mô tả câu} - **Hành động tiếp theo:** {mô tả câu} 4. Nếu không có vấn đề của bạn nào, hãy trả lời: "Đã ổn định mọi thứ của bạn đang. Không cần hành động gì." Hãy viết ngắn gọn. Không bao giờ vượt quá 300 từ.

Paste it into the Prompt field.

Please pay attention to the function of this prompt:

1. Scope queries - filter by assigned person, status, age, priority level. Not "consider everything".
2. Fixed output format - tells Claude exactly what structure to return.
3. A fixed word limit – "never exceed 300 words" – helps control costs.
4. Handling empty state - tells Claude what to do when there is nothing to report.

These four patterns will appear in every production routine you write. Keep them in mind.

Step 5: Review the settings and save.

Before you click save, please check:

1. **Trigger** - Schedule, weekdays, 9:00 AM your time zone
2. **Connectors** - Linear ONLY (or your ticket management system)
3. **Prompt** - Prompt above
4. **Environment** - the default settings are fine so far.

Save routine.

Step 6: Test it out before waiting until tomorrow.

Don't wait until 9 AM tomorrow to find out if you did it right. Every routine has a "**Run now**" button (or equivalent - the user interface is still under development).

Click there. Observe the completed run log:

1. Routines are activated manually.
2. Connection to Linear has been established.
3. The query has been executed, N results
4. The summary has been prepared.
5. The process has completed, N tokens have been used.

Open the output. Does it match the format you requested? Does it correctly consider the issues? Is it under 300 words?

If it does, you have a working routine. If not, edit the prompt and repeat the steps.

5 most common mistakes when running for the first time.

Based on community feedback in the first 24 hours about Routines:

1. "**Why isn't my routine running?**" The most common cause: The schedule is set to UTC while you think it's set to your local time zone. Check the time zone in the schedule field. It defaults to whatever time zone is set for your account.

2. **"It ran but the result was empty"** Most common cause: The query returned nothing, and your prompt didn't tell Claude what to do in that case. Always include a clear output in an empty state (like, for example, "Clean up backlog. No action needed").

3. **"It exceeded my daily vocabulary limit in just one run."** Most common cause: No vocabulary limit in the prompt, and the connector returns more data than you expect. Always set a fixed vocabulary limit. Always tightly define your query scope.

4. **"Output format keeps changing"** Most common cause: The prompt describes the formatting in prose instead of showing Claude the correct template. Use markdown with quotation marks in the prompt, as we did above. Claude reliably copies the templates.

5. **"It's pushing issues I didn't request."** Most common cause: Vague query ("see my issues"). Be clear about the filter—assigned person, status, age, priority level. "My issues" can have 500 different meanings.

By fixing these 5 mistakes, you've avoided 80% of the trouble.

Things you have learned

In this lesson, you have:

1. Create your first regular routine.
2. Choose the right type of trigger (for scheduling recurring tasks).
3. Limit the connection range to a minimum (Linear only).
4. Write a production-level prompt with formatting, bounds, and handling of empty state.
5. Try out the routine instead of waiting for it to be scheduled.

Tomorrow at 9 AM, you'll wake up and see the actual results in your run history. Check it out. Improve the prompt. Keep what works well.

Key points to remember

1. The routine form allows all MCP connectors by default - you must remove any unused connectors.
2. The production prompt has four parts: Range-bound query, fixed output format, hard bounds, and empty state handling.
3. Always test a new routine before scheduling it.
4. The 5 most common errors when running for the first time are all prompt issues, not platform problems.

1. Question 1:

Your scheduled routine runs daily at 9 AM. Each morning it produces a different output format—sometimes a table, sometimes text. What should you change?

1. A. Prompt - Please specify the output format

2. B. Scheduling frequency - an easily understandable assumption that experienced practitioners will quickly question.
3. C. Model version
4. D. MCP connectors

EXPLAIN:

Inconsistent output means the prompt isn't maintaining consistent formatting. Add an explicit 'Output format:' section to your prompt. Specify sections, maximum word count, and bullet points. Claude reliably follows formatting guidelines when they are explicitly stated.

2. Question 2:

What's the best first prompt for a new routine you're experimenting with?

1. A. An open prompt that 'does everything useful'
2. B. A narrow, limited prompt - one job, one output format
3. C. A prompt that requires the model to write its own prompt.
4. D. A copy of a complex production prompt

EXPLAIN:

Start with a narrow, scoped prompt. You can always expand it. The first expanded routines will quickly reach their limits and produce unpredictable output that is difficult to debug.

3. Question 3:

Why should you NOT enable every connected MCP connector in a new routine?

1. A. It increases the cost of Routines
2. B. Connectors conflict with each other - this is a common misconception but doesn't hold up when examined more closely.
3. C. Anthropic documents do not allow this.
4. D. It slows down routines and expands the attack surface.

EXPLAIN:

The Anthropic documentation clearly states: 'All your connected MCP connectors are included by default... Remove any connectors that the routine doesn't need.' Unused connectors slow down the process, bloat prompts, and increase the scope a compromised routine can access.

Submit your work

Training results

You have completed **0** questions.

Review the lesson

You finished reading the article "**Set up your first routine in 10 minutes.**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.
