

SLA in Redmine

Make agreed response and resolution times visible in the issue and evaluate them



Feature of the Redmine Reporting plugin, AlphaNodes GmbH, EN version

alphanodes.com/redmine-reporting

Contents

1. The essentials up front
2. Where the deadlines come from
3. What you see in the issue
4. How the clock runs
5. When a response and when a solution counts
6. Keeping an eye on SLA day to day
7. Three examples from practice
8. What the SLA measures
9. Frequently asked questions
10. Next steps

1. The essentials up front

An SLA, short for Service Level Agreement, is an agreement about how quickly a request is responded to and how quickly it is dealt with. It is made between your team and its customers, or internally between two departments.

With the Reporting plugin this commitment does not only live in the contract, but in the issue itself. The issue shows by when a response and by when a solution is due, and whether that was met. You do not have to maintain anything on top of it, because these values appear automatically as soon as an issue falls under an SLA agreement.

At its core there are two times:

- **Response time:** by when someone has to take care of the issue.
- **Resolution time:** by when the issue has to be done.

Both are counted from the moment the issue was created. It is also possible to agree on only one of the two times.

What SLA is meant for

SLA pays off wherever you have to meet agreed times and want to be able to prove it: in support and service desk, in IT operations, with internal service providers that have a service agreement. The ideal case is a commitment of the form “we get back to you within X hours and deliver a solution within Y hours”, measured from the moment the request comes in.

SLA pays off as soon as such commitments exist: it makes them visible in the issue and verifiable afterwards. For the plain question “what is urgent and by when”, priority and due date are enough.

What you can rely on

- **No duplicate maintenance:** The SLA values are ordinary issue fields. You can filter, sort, group and export by them just like by any other field.
- **The same display everywhere:** in the issue, in the issue list, on the task board cards, in the SLA report and in the dashboard block.
- **Unambiguous even without colour:** The three states of a deadline carry three different symbols, not the same symbol three times in a different colour.
- **Verifiable:** Which issues missed their deadline can be evaluated for any period and written out as a file.

SLA is a feature of the Redmine Reporting plugin and is set up per project. The plugin is available on its own or bundled with further AlphaNodes plugins. All descriptions refer to the Reporting plugin from version 4.6.0 onwards.

2. Where the deadlines come from

Three settings work together. They are often mixed up, which is why they come right at the beginning:

- **The tracker decides whether.** Only issues in the selected trackers fall under SLA at all, for example Support and Incident. An issue in the tracker Feature stays out of it.
- **The priority decides how long.** For every priority the hours to respond and the hours to solve are stored. Urgent gets shorter times than Normal.
- **The project decides which values apply.** SLA is set up per project. Two projects can have completely different times for the same priority.

So the tracker does not control the duration. There is no “four hours for incidents”. There is “four hours for priority High in this project”, and incidents only fall under SLA at all because their tracker was selected. If you want different times for a type of issue, the way to do it is through the priority or through a separate project.

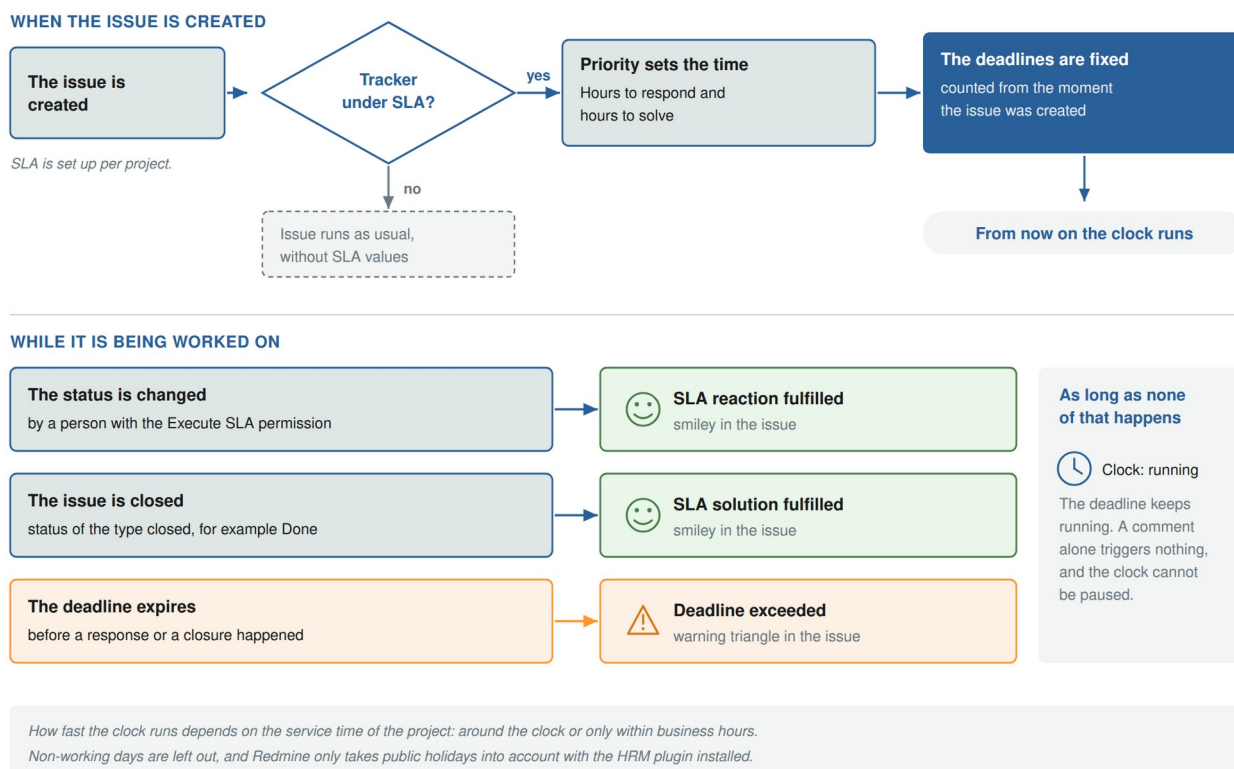


Figure 1: How an SLA issue runs. When it is created, it is decided whether and with which deadlines an issue falls under SLA. After that, only what happens in the issue counts.

Not every priority needs values. If a row stays empty, there is no deadline for that priority. In the same way only a response time can be agreed, with no resolution time at all.

3. What you see in the issue

If an issue falls under SLA, four additional values appear in it:

Field	What it tells you		
SLA reaction	Whether the issue has already been responded to and within what time frame		
SLA solution	Whether the issue has already been solved and when the resolution period expires		
Hours to respond	The time allowed until a response, and the concrete end time		
Hours to solve	The time allowed until the solution, and the concrete end time		
Round trips:	0	Issue life time:	14 days
SLA reaction:	Responded in about 5 hours 😊	Hours to respond:	6:00 hours - 11/15/2021 15:00
SLA solution:	Solved in 14 days ⚠️	Hours to solve:	48:00 hours - 11/22/2021 12:00

Figure 2: The SLA section in the issue. On the left the state of reaction and solution with its symbol, on the right the time allowed and the concrete end time. Round trips belongs to a different feature of the plugin.

The three symbols

You can tell the state of a deadline from the shape of the symbol:

- **Smiley:** The commitment was met.
- **Warning triangle:** The deadline was exceeded.
- **Clock:** The deadline is still running.

SLA fulfilled	SLA expiration reaction	SLA expiration solution
No	Responded in about 5 hours 😊	Solved in 14 days ⚠️
Yes	Responded in less than a minute 😊	
No	Responded in 5 months ⚠️	Solved in 5 months ⚠️
Yes	Responded in 8 minutes 😊	Solved in 11 minutes 😊
Yes	Responded in less than 10 seconds 😊	

Figure 3: The SLA columns in the issue list. The smiley stands for a commitment that was met, the warning triangle for a deadline that was exceeded.

Colour is never the only feature here. The display is the same everywhere: in the issue, in the columns SLA expiration reaction and SLA expiration solution of the issue list, on the task board cards as well as in the SLA report and in the corresponding dashboard block.

You control who sees the SLA values through the roles: the View SLA permission shows the section in the issue. If a wiki page with the SLA conditions is stored in the project, it is linked directly in the issue, so that everyone can look up the agreed key figures.

4. How the clock runs

A deadline of four hours can mean very different things. What matters is the service time defined in the project.

- **Around the clock:** The time runs continuously, at night and at the weekend as well. Four hours are four hours.
- **Business hours only:** The time only runs within the agreed working day. If an issue comes in on Friday at 16:30 and the working day ends at 17:00, only 30 minutes have been used up by Monday.

In business time mode, non-working days do not count. These days apply uniformly to all projects in Redmine and cannot be set differently per project.

Public holidays

If the HRM plugin is installed as well, the public holidays maintained there are automatically taken into account for the business hours. So the clock stands still on a public holiday. Without the HRM plugin there is no holiday management in Redmine, and public holidays then count as normal working days.

Two points worth knowing

- **The server time zone is what counts**, not the time zone you have set in your user account.
- **The end time is fixed from creation** and stays stable. The deadline is therefore known from the start, also while an issue is waiting for a reply. What this measures is explained in chapter 8.

5. When a response and when a solution counts

The response

A response is recorded with the status change, for example from *New* to *In Progress*, made by a person with the Execute SLA permission. That makes the response unambiguous: the issue has demonstrably gone into progress.

This is deliberate. On the one hand the issue workflow stays binding, because the commitment only counts as met once the issue has actually gone into progress. On the other hand people from outside, such as colleagues of the requester, cannot mark a deadline as met by accident.

The solution

An issue is solved as soon as it has a status of the type closed. This also applies to your own statuses, as long as they are marked as closed.

If only a response time is agreed, the SLA counts as fulfilled as soon as a response has been made. If a resolution time is agreed on top of that, it only counts as fulfilled once the issue has been closed as well.

As usual, anyone who is allowed to work in the project may edit an SLA issue. For the assessment of the commitment, however, only what someone with the Execute SLA permission does counts.

6. Keeping an eye on SLA day to day

An agreed deadline is of little use if nobody sees it before it expires. There are several ways to do that, and they combine well.

The issue list

In projects with SLA additional columns and filters are available, among them *SLA fulfilled*, *SLA expiration reaction*, *SLA expiration solution*, *SLA solved on* and *first response date*. Particularly useful are the filters *SLA respond time exceeded* and *SLA solve time exceeded*. A list set up this way can be saved as a query and shared with the team, so that everyone uses the same view.

The dashboard block in the project

The *SLA Information* block belongs on the project overview. At the top it shows the numbers: how many SLA issues there are, how many of them are open and for how many the response or resolution time was exceeded. Below that are the agreed key figures, that is the service time, the SLA trackers and the table of priorities with their hours. Every number is linked to the matching pre-filtered issue list, which can be edited further and exported.

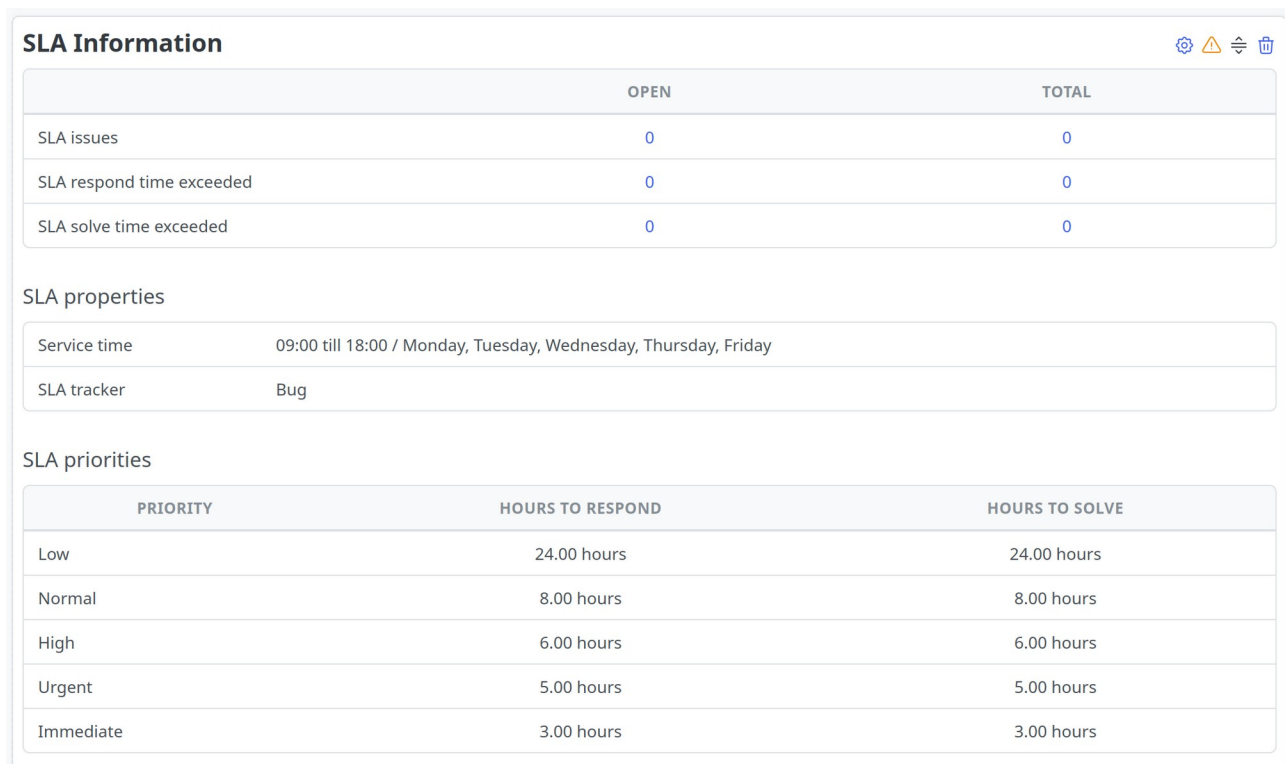


Figure 4: The SLA Information dashboard block on the project overview. At the top the numbers for open and exceeded issues, below them the agreed key figures and the hours per priority.

This way the block answers both questions in one place: what is currently running and what was actually agreed. Anyone new to the team does not have to go looking for the agreement first.

The dashboard block on the home page

On the Redmine home page the same block is available in a shorter form, across all projects you work in. It shows three lists: *SLA issues*, *exceeded response time* and *exceeded resolution time*, each of them open and as a total. That is the right view if you look after several customer projects at the same time.

Your own SLA dashboard

Instead of spreading individual blocks around, you can put together your own entry page just for SLA and share it with the team. Sensible parts of it are the *SLA Information* block, saved issue lists for the critical cases, *counter boxes* with thresholds and the chart for the *first response time*. Counter boxes are particularly good for escalation: with the threshold you define from which point a number is displayed prominently, so that an increase catches the eye immediately.

Evaluation and escalation

For a monthly record you filter the issue list to the period and export it as a CSV, PDF or Excel file. If, when a deadline is approaching or has been missed, notifications are to go

out automatically or issues are to be passed on, the Automation plugin takes care of that. It uses the same SLA data for this.

7. Three examples from practice

Support with two levels

In the customer project the trackers *Support* and *Incident* are placed under SLA. For priority *High* one hour to respond and four hours to solve apply, for *Normal* four hours to respond and one working day to solve. The *service time* is business hours. The project overview carries the *SLA Information* block, and the service desk staff work with a saved list sorted by SLA expiration reaction.

Internal IT operations

The IT department has agreed times with the departments and only maps the response time, because the duration of a solution varies too much. So only the *Hours to respond* column is filled in. The service time is business hours, and because the HRM plugin is in use, public holidays are automatically left out.

Monthly record

At the end of the month the team lead filters the issue list to the previous month and the trackers concerned, shows the columns *SLA fulfilled*, *SLA expiration reaction* and *SLA expiration solution*, and exports the result. This becomes the report for the customer, without anyone having to compile figures manually.

8. What the SLA measures

To put the tool to good use, it helps to look at which figure it measures.

The time is measured from the request

The end time of a deadline is fixed the moment the issue is created and stays stable. This way the SLA measures exactly what was promised to the customer: the time from their request onwards. That figure is transparent for both sides, because it cannot be shifted afterwards.

If you work with a model in which the deadline rests while a question is pending with the customer, talk to us. We will look at your agreement and tell you how it can be represented in Redmine.

Agreed times stay stable

- The SLA applies from the moment you set it up. Issues that already exist stay as they are, so that no commitment is created retroactively.

- If you adjust the hours later, they apply to new issues. A running issue keeps the deadline that applied when it was created and picks up the new values with its next change of priority.
- If a project no longer needs SLA, you switch the module off: the values disappear from the display, while the issues themselves are kept in full.

One agreement per project

Within a project exactly one set of times applies. If you look after several customers with different contracts, give each customer their own project. That is the clearer solution anyway, because permissions and evaluations are then cleanly separated as well.

9. Frequently asked questions

- **When does an issue show SLA values?** As soon as four things come together: its tracker is selected for SLA, hours are stored for its priority, it was created after SLA was set up, and your role may see SLA.
- **What counts as a response?** The status change. It shows that the issue has actually gone into progress, and not just that someone wrote something.
- **What happens to running issues when we change the times?** They keep their original deadline and pick up the new values with their next change of priority.
- **How is the time counted while we are waiting for the customer?** The deadline keeps running, because what is measured is the time from the request onwards. If your agreement works differently, we will clarify up front how it can be represented. See chapter 8.
- **Does an answer outside business hours count?** Yes. When a response was made is always recorded. Business hours only determine how the deadline is calculated.
- **Whose actions count towards the assessment?** Those of a person with the Execute SLA permission. That keeps the evaluation reliable, even when many people are involved in an issue.
- **How do we get a record for the customer?** Filter the issue list to the period, show the SLA columns and export it. See chapter 6.
- **Do we see it early when a deadline gets tight?** Yes. Counter boxes with a threshold and sorting by expiration time make it visible, and the Automation plugin notifies or escalates automatically.

10. Next steps

- **Take a look:** In the public demo at redmine-demo.alphanodes.net the SLA fields, filters and the dashboard block are set up.
- **Check the fit:** Chapter 8 describes which figure is measured. From that you can quickly tell whether your agreement matches it.

- **Go deeper:** The Reporting plugin user manual describes every setting in detail.

Provided by:

AlphaNodes GmbH, Bavariafilmpfad 3, D-82031 Grünwald, Germany.

Legal notice: <https://alphanodes.com/imprint>

Last updated: August 2026.