DEVELOPER REFERENCE / OPENAPI 3.0.3

RedRover API

Access setup and complete integration reference.

Production endpoint
https://askredrover.com/api/v1
November 17, 2026 edition

9 integration operations	4 browser key operations	21 MCP tools
Read and write work through the supported REST interface.	Create, list, rotate and revoke keys through an authorized browser session.	Full deployed argument schemas, plus the separate MCP transport.

This reference includes the current production base URL, required grants, request parameters, schemas, examples, errors and setup steps. Complete machine-readable OpenAPI JSON and YAML are embedded in this PDF and supplied alongside it.

Published by Hatchery LLC / November 17, 2026

Scope: the supported external integration API, its browser key administration and MCP. Internal web-app AJAX routes, provider callbacks and public support intake are separate contracts.

Contents

1. Set up access 3

2. Authentication and permissions 4

3. First requests 5

4. Operation index 6

5. Integration operation reference 7

6. Work schemas and type rules 14

7. Browser-only key administration 18

8. Discovery and capacity models 21

9. MCP access and complete tool catalog 32

10. Errors, retries and troubleshooting 44

11. Downloadable contracts 45

How to use this reference
Start with access setup and the quickstart. Use the operation index for an endpoint and the schema sections for fields. PDF bookmarks and the contents links navigate the document.

1. Set up access

Use an account owner or eligible Super User to create a REST key. The administrator must currently have system.manage. A key acts as the selected user and cannot add permissions that user lacks.

1 / Sign in

Open askredrover.com/login and sign in to the intended account.

2 / Open key administration

Choose System > API keys & integrations, or open askredrover.com/api-keys while signed in.

3 / Create a key

Click Create API key. Enter an integration name, select the acting account user, and choose an expiry of 1-365 days. The default is 90 days.

4 / Choose the grants

Start with workspaces:read and grant only the resources the integration needs. Choose write only where it must save work or append comments. Ensure the acting user has current workspace membership and the required domain permissions.

5 / Save the secret once

Copy the secret from Save this key now into a server secret manager. RedRover shows it only at creation or rotation. Do not put it in a repository, URL, browser bundle or localStorage.

6 / Verify access

Call GET /capabilities, then GET /workspaces. Choose a returned workspace ID. Read GET /options before selecting status, priority or assignee values.

> If the key page is unavailable
> Confirm the current account, administrator eligibility and system.manage permission. An API key cannot open or manage other keys. Do not change another person's roles as a workaround.

2. Authentication and permissions

The REST API accepts exactly one of these credential headers:

```
X-API-Key: <secret>

# Alternative header
Authorization: ApiKey <secret>
```

Do not use Bearer for this REST interface. Bearer is the separate MCP transport credential. Invalid or conflicting explicit credentials never fall back to a signed-in browser cookie. URL parameters named key, apiKey or api_key are rejected.

Resource	Allowed grants	What it enables
workspaces	none / read	Required read access for workspace-scoped work.
tasks	none / read / write	List, read, create or update tasks.
requirements	none / read / write	List, read, create or update requirements.
goals	none / read / write	List, read, create or update eligible goals.
releases	none / read / write	List, read, create or update eligible releases.
milestones	none / read / write	List, read, create or update milestones.
comments	none / read / write	Read or append comments on readable work.
capacity	none / read	Read resource/forecast data with all required planning grants and time oversight.

A write grant also allows reads. Access is the intersection of key grants, current active account/user, workspace membership, domain permissions and readable ancestors. For a task inside a requirement and release, grant read access to the requirement and release too. Removing access or revoking the key takes effect on subsequent requests.

> Capacity grant checklist
> Grant workspaces, capacity, tasks, requirements, releases and milestones read access. The selected user must also have time oversight for the project, or the applicable resource-management authority.

Same-origin browser clients may use the existing session. GET /capabilities supplies a csrf token in session mode. Send it as X-CSRF-Token on every session mutation. Cross-origin browser access is not enabled by default; place integration keys in your own authenticated backend.

3. First requests

Load the secret securely into RR_API_KEY in your terminal or backend environment. The examples deliberately contain no live secret. Use HTTPS and do not enable redirect following for a credential-bearing request.

```
curl --silent --show-error --fail-with-body \
  -H "X-API-Key: ${RR_API_KEY}" \
  "https://askredrover.com/api/v1/capabilities"

curl --silent --show-error --fail-with-body \
  -H "X-API-Key: ${RR_API_KEY}" \
  "https://askredrover.com/api/v1/workspaces"
```

Set RR_WORKSPACE_ID to an ID returned by /workspaces. Internal API identifiers remain numeric; customer-facing record links use RedRover's existing opaque routing scheme.

```
curl --silent --show-error --fail-with-body \
  -H "X-API-Key: ${RR_API_KEY}" \
  "https://askredrover.com/api/v1/options?workspaceId=${RR_WORKSPACE_ID}"

curl --silent --show-error --fail-with-body \
  -H "X-API-Key: ${RR_API_KEY}" \
  "https://askredrover.com/api/v1/items?workspaceId=${RR_WORKSPACE_ID}&type=task&limit=50"
```

Import the Swagger / OpenAPI contract

Import RedRover-OpenAPI.yaml or RedRover-OpenAPI.json into a compatible OpenAPI client, editor or code generator. The same files are attached inside this PDF. Use a trusted desktop client or server-side implementation for authenticated execution; do not paste a production key into a public hosted editor.

> Production base URL
> Use https://askredrover.com/api/v1 for REST requests. The /mcp agent interface uses its separate endpoint described in section 9.

4. Operation index

REST paths below are relative to https://askredrover.com/api/v1. MCP has its own root endpoint.

Method	Path	Access / purpose
GET	/capabilities	Any valid REST key or current session.
GET	/workspaces	workspaces:read; visible active workspaces.
GET	/options	workspaces:read plus a readable planning resource.
GET	/items	workspaces:read plus selected type read.
GET	/item	workspaces:read plus item and ancestor read.
POST	/items/save	workspaces:read plus selected type write.
GET	/comments	comments:read plus item and ancestor read.
POST	/comments	comments:write plus item read and comments.add.
GET	/capacity	Six read grants and current time oversight.
GET	/keys	Browser administrator session only.
POST	/keys/create	Browser administrator + CSRF only.
POST	/keys/rotate	Browser administrator + CSRF only.
POST	/keys/revoke	Browser administrator + CSRF only.
POST	https://askredrover.com/mcp	Separate personal MCP Bearer connection.

POST success uses HTTP 200. This interface has no general DELETE route. Removal, approval, user management, finances, uploads and account configuration are not exposed through an integration key. Shared work states must obey domain rules and release scope locks.

5. Integration operation reference

GET /capabilities

Inspect authentication and key resource grants

Required access: A valid REST key or current browser session.

No parameters. Key mode returns a resource-to-grant map. Session mode returns supported resource names and a fresh csrf token.

200 response example

```
{
  "version": 1,
  "authentication": "api-key",
  "resources": {
    "workspaces": "read",
    "tasks": "write",
    "requirements": "read"
  },
  "permissionModel": "Key grants intersect current account, workspace, item and ancestor access."
}
```

Error envelope: {"error":"message"}. See section 10 for status handling. Examples are illustrative and contain no live account identifiers.

GET /workspaces

Find the workspaces the current actor can access

Required access: workspaces:read.

No parameters. Returns visible active workspaces. The response includes the existing workspace id, name, uuid, sample flag and appearance metadata.

200 response example

```
{
  "workspaces": [
    {
      "id": 123,
      "name": "Example workspace",
      "sample": false,
      "uuid": "00000000-0000-4000-8000-000000000123",
      "appearance": {
        "color": "#e12026",
        "icon": "layers-intersect",
        "useLogo": false,
        "logoId": 0,
        "revision": 0,
        "canEdit": false
      }
    }
  ]
}
```

Error envelope: {"error":"message"}. See section 10 for status handling. Examples are illustrative and contain no live account identifiers.

GET /options

Read configured workflow, priority and assignment choices

Required access: workspaces:read and read access to at least one planning type.

Call this before writing task/requirement statusKey, priorityKey or assigneeId. The types array is filtered by key grants. Status definitions apply to tasks and requirements; the other type states are listed in section 6.

Parameter	Required	Meaning
workspaceId	Yes	Positive ID from GET /workspaces.

200 response example


```
{
  "statuses": [
    {
      "type": "task",
      "key": "planned",
      "canonicalState": "planned",
      "label": "Planned",
      "color": "#527FAD",
      "sortOrder": 20,
      "active": true,
      "version": 1
    }
  ],
  "priorities": [
    {
      "key": "normal",
      "label": "Normal",
      "color": "#527FAD",
      "sortOrder": 30,
      "active": true,
      "version": 1
    }
  ],
  "assignees": [
    {
      "id": 456,
      "name": "Example assignee",
      "avatar": 0
    }
  ],
  "types": [
    "goal",
    "milestone",
    "release",
    "requirement",
    "task"
  ]
}
```


Error envelope: {"error":"message"}. See section 10 for status handling. Examples are illustrative and contain no live account identifiers.

GET /items

List a bounded page of one planning type

Required access: workspaces:read plus the selected type read, with readable ancestors.

Items are ordered by ascending ID. The after cursor is exclusive. Continue with nextAfter while hasMore is true. A nested task may be omitted when its requirement or release is unreadable.

Parameter	Required	Meaning
workspaceId	Yes	Positive workspace ID.
type	No	task default; requirement, goal, release or milestone.
after	No	Exclusive item ID cursor; default 0.
limit	No	1-100 results; default 50.

200 response example

```
{
  "hasMore": true,
  "nextAfter": 789,
  "items": [
    {
      "id": 789,
      "type": "task",
      "title": "Review an integration",
      "description": "Example task",
      "status": "planned",
      "version": 1,
      "ownerId": 456,
      "fields": {
        "statusKey": "planned",
        "priorityKey": "normal",
        "estimatedHours": 2,
        "progressPercent": 0
      }
    }
  ]
}
```

Error envelope: {"error":"message"}. See section 10 for status handling. Examples are illustrative and contain no live account identifiers.

GET /item

Read one current planning record

Required access: workspaces:read plus item and ancestor read.

Returns the Work schema. Fetch it immediately before an update, preserve its id and version, and send only the supported record fields. Unavailable or foreign-account work is denied.

Parameter	Required	Meaning
workspaceId	Yes	Positive workspace ID.
id	Yes	Positive work-item ID.

Error envelope: {"error":"message"}. See section 10 for status handling. Examples are illustrative and contain no live account identifiers.

POST /items/save

Create or update work through RedRover domain validation

Required access: workspaces:read plus tasks/requirements/goals/releases/milestones:write for the selected type.

Content-Type: application/json. Maximum JSON body is 150000 bytes. Omit id/version to create; include the current id/version to update. Unknown input fields are rejected. Omitted internal/private associations are preserved. Release locks, live actor rights and approval rules remain in force.

Request example

```
{
  "workspaceId": 123,
  "record": {
    "type": "task",
    "title": "Review API integration",
    "description": "Confirm the external integration.",
    "status": "planned",
    "ownerId": 456,
    "fields": {
      "priorityKey": "normal",
      "estimatedHours": 2,
      "progressPercent": 0
    }
  }
}
```

200 response example

```
{
  "id": 789,
  "type": "task",
  "title": "Review API integration",
  "description": "Confirm the external integration.",
  "status": "planned",
  "version": 1,
  "ownerId": 456,
  "fields": {
    "statusKey": "planned",
    "priorityKey": "normal",
    "estimatedHours": 2,
    "progressPercent": 0
  }
}
```

Error envelope: {"error":"message"}. See section 10 for status handling. Examples are illustrative and contain no live account identifiers.

Update safely
Fetch /item, edit the supported record and submit its current version. A 409 requires a fresh read and reconciliation. Creates and comments have no general idempotency key; inspect the result before retrying an uncertain write.

GET /comments

Read recent work comments

Required access: workspaces:read, comments:read and item/ancestor read.

Returns the latest 100 comments in descending ID order. This route has no pagination cursor. GET createdAt is a database timestamp string.

Parameter	Required	Meaning
workspaceId	Yes	Positive workspace ID.
id	Yes	Positive work-item ID.

200 response example

```
{
  "comments": [
    {
      "id": 901,
      "body": "Integration reviewed.",
      "createdAt": "2026-11-17 12:00:00"
    }
  ]
}
```

Error envelope: {"error":"message"}. See section 10 for status handling. Examples are illustrative and contain no live account identifiers.

POST /comments

Append a comment

Required access: workspaces:read, comments:write, item/ancestor read, and the actor's comments.add permission.

Comment body must be nonblank, 1-12000 characters and contain no NUL. The route accepts only workspaceId, id and body. Notification behavior follows current domain policy and recipient preferences. POST adds author and actorId to the Comment response.

Request example

```
{
  "workspaceId": 123,
  "id": 789,
  "body": "Integration reviewed."
}
```

200 response example

```
{
  "id": 901,
  "body": "Integration reviewed.",
  "author": "Example user",
  "createdAt": "2026-11-17T12:00:00Z",
  "actorId": 456
}
```

Error envelope: {"error":"message"}. See section 10 for status handling. Examples are illustrative and contain no live account identifiers.

GET /capacity

Read resource load and delivery scenarios

Required access: Read grants for workspaces, capacity, tasks, requirements, releases and milestones; current time oversight/resources.manage authority.

Returns eight UTC weeks starting on Monday. If from is omitted, the current UTC week is used. The date must be within two years of today. workspaceId omitted or 0 means all visible workspaces. Forecast dates may be null when data is incomplete. Capacity-key output is read-only; canManage, canSetTarget and work.canEdit are false.

Parameter	Required	Meaning
workspaceId	No	0/all visible by default; positive ID narrows the projects.
from	No	ISO YYYY-MM-DD; normalized to Monday.

200 response example


```
{
  "workspaceId": 123,
  "canManage": false,
  "from": "2026-11-16",
  "weeks": 8,
  "today": "2026-11-17",
  "timezone": "UTC",
  "scopeNote": "Only workspaces with current time oversight are included.",
  "workspaces": [
    {
      "id": 123,
      "title": "Example workspace"
    }
  ],
  "weekDates": [
    "2026-11-16",
    "2026-11-23"
  ],
  "people": [],
  "focusConnections": [],
  "projects": [],
  "reportCharts": {},
  "focusPeriods": []
}
```


Error envelope: {"error":"message"}. See section 10 for status handling. Examples are illustrative and contain no live account identifiers.

The shortened example shows the envelope. Real weekDates contains eight week starts. The capacity model section lists the complete business-field names and the deliberately extensible chart/release projections.

6. Work schemas and type rules

Work property	Type	Contract
id	integer / int64	Response; omit on create, include on update.
type	string	task, requirement, goal, release or milestone.
title	string	Required nonblank text, at most 240 characters.
description	string	Text up to 20000 characters; optional input, returned as text.
status	string	Required canonical state valid for this type.
version	integer	Response; current value required for update.
ownerId	integer / int64	Actor-owned/domain-authorized current user ID; omitted/0 selects the actor on create.
fields	object	Required supported type-specific fields; unknown properties rejected.

Type	Canonical states accepted by the current service	Additional requirements
task	draft, planned, in_progress, done	Use current task statusKey/priorityKey from /options.
requirement	draft, planned, in_progress, approved, done	Entering approved is blocked by this API; use the approval workflow.
release	draft, planned, in_progress, committed, released	versionLabel required. Release readiness/approval rules remain enforced.
goal	draft, planned, in_progress, done	level, positive target and unit required.
milestone	draft, planned, in_progress, done	Optional releaseId and dueDate; release locks apply.

For tasks and requirements, send the configured fields.statusKey with its canonical status when choosing a custom status. A complete RICE estimate requires reach, impact, confidence, effort and reachPeriod. Effort is person-months; estimatedHours is person-hours.

Supported fields object

Field	Type	Meaning / limits
statusKey	string / null	Configured workspace status key for tasks/requirements; use with its canonical status.
priorityKey	string / null	Configured workspace priority key. Query /options.
assigneeName	string / null	Returned display name; use assigneeId to select an assignee.
blockedReason	string / null	Explanation of blocked work, up to 2000 characters. Maximum 2000 characters.
dueDate	string / date / null	ISO date, YYYY-MM-DD; omitted or null where optional.
startDate	string / date / null	ISO date, YYYY-MM-DD; omitted or null where optional.
targetDate	string / date / null	ISO date, YYYY-MM-DD; omitted or null where optional.
actualDate	string / date / null	ISO date, YYYY-MM-DD; omitted or null where optional.
versionLabel	string / null	Required for releases; 1 to 40 characters.
acceptance	string / null	Requirement acceptance criteria, up to 12000 characters.
reachPeriod	string / null	RICE quarter such as 2026-Q4. Needed with a complete estimate.
unit	string / null	Required goal unit, 1 to 80 characters.
level	string / null	Required goal visibility. workspace/corporate/executive/personal; legacy corporate/executive values are retained by the validator. Personal goals remain owner-only.
estimatedHours	number / null	Estimated person-hours; separate from RICE effort and recorded time.
progressPercent	number / null	Completion percentage from 0 to 100.
reach	number / null	RICE reach, 0 to 1000000000, at most 4 decimal places.
impact	number / null	RICE impact: 0.25, 0.5, 1, 2 or 3.
confidence	number / null	RICE confidence percentage from 0 to 100; at most 2 decimal places.
effort	number / null	RICE effort in person-months, positive and at most 1000000; at most 4 decimal places.
target	number / null	Required positive goal target; at most 1000000000000 and 4 decimal places.
current	number / null	Nonnegative goal progress, at most 1000000000000 and 4 decimal places.
assigneeId	integer / int64 / null	Eligible current workspace assignee ID; query /options.

Field	Type	Meaning / limits
parentId	integer / int64 / null	Parent goal ID. Circular goal ancestry is rejected.
parentItemId	integer / int64 / null	Parent requirement or task ID. A task inherits its parent release.
requirementId	integer / int64 / null	Legacy direct requirement association for tasks; prefer parentItemId.
releaseId	integer / int64 / null	Associated release ID; membership changes obey release scope locks.
goalId	integer / int64 / null	Linked goal ID; key and actor must be able to read the linked goal.

Working type-specific examples

Each object below is the record inside POST /items/save. Supply workspaceId in the outer object. The examples intentionally use default draft states and no real user IDs.

Requirement

```
{
  "type": "requirement",
  "title": "A reliable integration",
  "description": "External systems can safely synchronize work.",
  "status": "draft",
  "fields": {
    "acceptance": "Authorized workspace changes are available to connected systems.",
    "reach": 100,
    "impact": 2,
    "confidence": 80,
    "effort": 1,
    "reachPeriod": "2026-Q4"
  }
}
```

Release

```
{
  "type": "release",
  "title": "Integration launch",
  "description": "Deliver the connected workspace integration.",
  "status": "draft",
  "fields": {
    "versionLabel": "1.0",
    "startDate": "2026-11-17",
    "targetDate": "2026-12-01"
  }
}
```

Goal

```
{
  "type": "goal",
  "title": "Complete integration readiness",
  "description": "Track readiness for the integration.",
  "status": "draft",
  "fields": {
    "level": "workspace",
    "target": 100,
    "current": 0,
    "unit": "percent",
    "dueDate": "2026-12-01"
  }
}
```

Milestone

```
{
  "type": "milestone",
  "title": "Integration acceptance complete",
  "description": "Finish the integration acceptance review.",
  "status": "draft",
  "fields": {
    "dueDate": "2026-12-01"
  }
}
```

Relationships and protected fields
Use parentItemId for a task under a requirement/task; release membership is inherited and must be consistent. All linked ancestors require read access. Scope locks still apply. Approval state, internal approval snapshots, dependencies, rich-content fields and other fields outside the bounded DTO cannot be inserted through this route.

7. Browser-only key administration

These four endpoints share /api/v1 but require a current browser session. The administrator must have system.manage and eligible Account Owner/Super User authority in the current account. An integration key receives 403 even if a browser cookie is also present.

Session and CSRF
GET /keys or GET /capabilities returns csrf. Include X-CSRF-Token on each browser POST. Do not place a session cookie or token into shared documentation or an integration application.

GET /keys

List account keys, usage metadata and eligible users

Required access: Administrator session; API keys denied.

No parameters. Returns KeyAdministration: accountId, defaultDays, resources, readOnlyResources, keys, usage, users and csrf. Secrets and database hashes are absent. Keys are scoped to the current account.

Error envelope: {"error":"message"}. See section 10 for status handling. Examples are illustrative and contain no live account identifiers.

POST /keys/create

Issue a new expiring API key

Required access: Administrator session plus X-CSRF-Token.

label is 1-100 characters. userId must be an eligible active user in the current account. days is 1-365, default 90. A workspaces:read grant is required. Only catalog resources are accepted; workspaces/capacity cannot have write access.

Request example

```
{
  "label": "Reporting integration",
  "userId": 456,
  "days": 90,
  "permissions": {
    "workspaces": "read",
    "tasks": "read",
    "requirements": "read",
    "releases": "read"
  }
}
```

200 response example

```
{
  "issued": {
    "key": {
      "id": "00000000-0000-4000-8000-000000000001",
      "label": "Reporting integration",
      "prefix": "display-only",
      "permissions": {
        "workspaces": "read",
        "tasks": "read"
      }
    },
    "secret": "<shown only once; store securely>"
  }
}
```

Error envelope: {"error":"message"}. See section 10 for status handling. Examples are illustrative and contain no live account identifiers.

POST /keys/rotate

Replace the secret for an active key

Required access: Administrator session plus X-CSRF-Token.

Required id is the key UUID, not the work-item ID. days is optional, 1-365 (default 90). Returns issued.key and issued.secret. The old secret becomes invalid. Save the new secret before changing the integration configuration.

Request example

```
{
  "id": "00000000-0000-4000-8000-000000000001",
  "days": 90
}
```

Error envelope: {"error":"message"}. See section 10 for status handling. Examples are illustrative and contain no live account identifiers.

POST /keys/revoke

Immediately end key access

Required access: Administrator session plus X-CSRF-Token.

Required id is the key UUID. Returns {"ok":true}. Calls with the revoked secret return 401. A revoked key cannot be rotated; create a new key through normal administration.

Request example

```
{
  "id": "00000000-0000-4000-8000-000000000001"
}
```

200 response example

```
{
  "ok": true
}
```

Error envelope: {"error":"message"}. See section 10 for status handling. Examples are illustrative and contain no live account identifiers.

Key metadata fields

Field group	Fields / behavior
Identity	id, accountId, userId, label, owner, role, roles.
Access	permissions, access (read/write), generation, active.
Lifecycle	createdAt, expiresAt, lastUsedAt, revokedAt; unused/revoked values may be null.
Display	prefix and compatibility key are display prefixes only.
Issue response	issued.key holds metadata; issued.secret appears only on create/rotate.
Usage	keyId, requests, denials, failures and optional metadata summaries. No request body or API secret.

8. Discovery and capacity models

The tables below expand the remaining response models. Response objects may contain additional current presentation metadata. Integrations should tolerate added response properties. Input objects remain deliberately bounded.

Capabilities

Property	Type / reference	Meaning
version	integer / int64	Required response field. Values: 1.
authentication	string	Required response field. Values: api-key, session.
resources	object | array<string>	Required response field. API key: map of granted resource to read/write. Browser session: array of supported resource names.
csrf	string	Browser sessions only; submit this as X-CSRF-Token for mutations.
permissionModel	string	Required response field.

Workspace

Property	Type / reference	Meaning
id	integer / int64	Required response field.
name	string	Required response field.
uuid	string	Required response field.
sample	boolean	See the attached schema; server-computed or domain metadata.
appearance	WorkspaceAppearance	See the attached schema; server-computed or domain metadata.

WorkspaceAppearance

Property	Type / reference	Meaning
color	string	See the attached schema; server-computed or domain metadata.
icon	string	See the attached schema; server-computed or domain metadata.
useLogo	boolean	See the attached schema; server-computed or domain metadata.
logoId	integer / int64	See the attached schema; server-computed or domain metadata.
revision	integer / int64	See the attached schema; server-computed or domain metadata.
canEdit	boolean	Acting user appearance authority; this does not grant the key an appearance write route.

Options

Configured status and priority keys apply to tasks and requirements. Canonical states for goals, releases and milestones remain type-specific.

Property	Type / reference	Meaning
statuses	array<StatusOption>	Required response field.
priorities	array<PriorityOption>	Required response field.
assignees	array<Assignee>	Required response field.
types	array<string>	Required response field.

StatusOption

Property	Type / reference	Meaning
type	string	See the attached schema; server-computed or domain metadata.
key	string	See the attached schema; server-computed or domain metadata.
canonicalState	string	See the attached schema; server-computed or domain metadata.
label	string	See the attached schema; server-computed or domain metadata.
color	string	See the attached schema; server-computed or domain metadata.
sortOrder	integer / int64	See the attached schema; server-computed or domain metadata.
active	boolean	See the attached schema; server-computed or domain metadata.
version	integer / int64	See the attached schema; server-computed or domain metadata.

PriorityOption

Property	Type / reference	Meaning
key	string	See the attached schema; server-computed or domain metadata.
label	string	See the attached schema; server-computed or domain metadata.
color	string	See the attached schema; server-computed or domain metadata.
sortOrder	integer / int64	See the attached schema; server-computed or domain metadata.
active	boolean	See the attached schema; server-computed or domain metadata.
version	integer / int64	See the attached schema; server-computed or domain metadata.

Assignee

Property	Type / reference	Meaning
id	integer / int64	See the attached schema; server-computed or domain metadata.
name	string	See the attached schema; server-computed or domain metadata.
avatar	integer / int64	See the attached schema; server-computed or domain metadata.

Capacity

Permission-scoped scenario data. Dates may be null when estimates, assignments, or capacity are missing. UTC Monday-Friday. Read-only API resource.

Property	Type / reference	Meaning
workspaceId	integer	See the attached schema; server-computed or domain metadata.
today	string / date	See the attached schema; server-computed or domain metadata.
from	string / date	See the attached schema; server-computed or domain metadata.
weeks	integer	See the attached schema; server-computed or domain metadata.
timezone	string	See the attached schema; server-computed or domain metadata.
canManage	boolean	False in API-key responses.
scopeNote	string	See the attached schema; server-computed or domain metadata.
projects	array<ProjectForecast>	See the attached schema; server-computed or domain metadata.
people	array<PersonCapacity>	See the attached schema; server-computed or domain metadata.
workspaces	array<object>	See the attached schema; server-computed or domain metadata.
weekDates	array<string / date>	See the attached schema; server-computed or domain metadata.
focusConnections	array<FocusConnection>	See the attached schema; server-computed or domain metadata.
reportCharts	object	Axis/Etch presentation metadata; not an additional business API.
focusPeriods	array<FocusPeriod>	See the attached schema; server-computed or domain metadata.

ProjectForecast

Property	Type / reference	Meaning
id	integer	See the attached schema; server-computed or domain metadata.
title	string	See the attached schema; server-computed or domain metadata.
targetDate	string	ISO date or empty string when no workspace date is configured.
forecastDate	string / date / null	See the attached schema; server-computed or domain metadata.
bufferedDate	string / date / null	See the attached schema; server-computed or domain metadata.
scopeHours	number	See the attached schema; server-computed or domain metadata.
remainingHours	number	See the attached schema; server-computed or domain metadata.
bufferedHours	number	See the attached schema; server-computed or domain metadata.
weightedProgress	number / null	See the attached schema; server-computed or domain metadata.
outlook	string	Values: no_scope, needs_data, needs_target, overcommitted, at_risk, within_capacity.
method	string	See the attached schema; server-computed or domain metadata.
dataIncomplete	boolean	See the attached schema; server-computed or domain metadata.
trend	array<ForecastTrend>	See the attached schema; server-computed or domain metadata.
work	array<ForecastWork>	See the attached schema; server-computed or domain metadata.
startDate	string	ISO date or empty string when no workspace date is configured.
forecastStart	string / date / null	See the attached schema; server-computed or domain metadata.
earnedHours	number	See the attached schema; server-computed or domain metadata.
capacityByTarget	number	See the attached schema; server-computed or domain metadata.
recordedPeriodHours	number	See the attached schema; server-computed or domain metadata.
plannedPeriodHours	number	See the attached schema; server-computed or domain metadata.
reservedWeeklyHours	number	See the attached schema; server-computed or domain metadata.
hoursGap	number	See the attached schema; server-computed or domain metadata.
briefVersion	integer / int64	See the attached schema; server-computed or domain metadata.

Property	Type / reference	Meaning
missingEstimates	integer / int64	See the attached schema; server-computed or domain metadata.
unassigned	integer / int64	See the attached schema; server-computed or domain metadata.
workCount	integer / int64	See the attached schema; server-computed or domain metadata.
completedCount	integer / int64	See the attached schema; server-computed or domain metadata.
assumptionsUsed	boolean	See the attached schema; server-computed or domain metadata.
canSetTarget	boolean	False in API-key responses.
targetBeyondHorizon	boolean	See the attached schema; server-computed or domain metadata.
releases	array<ReleaseForecast>	See the attached schema; server-computed or domain metadata.
reportChart	object	Presentation metadata for the owning Axis/Etch chart renderer.

ReleaseForecast

Read-only release delivery, readiness and forecast projection. Dates can be empty when unset; computed forecast dates can be null. It does not expose an approval mutation.

Property	Type / reference	Meaning
id	integer / int64	See the attached schema; server-computed or domain metadata.
workspaceId	integer / int64	See the attached schema; server-computed or domain metadata.
workCount	integer / int64	See the attached schema; server-computed or domain metadata.
completedCount	integer / int64	See the attached schema; server-computed or domain metadata.
missingEstimates	integer / int64	See the attached schema; server-computed or domain metadata.
daysToTarget	integer / int64	See the attached schema; server-computed or domain metadata.
forecastVarianceDays	integer / int64	See the attached schema; server-computed or domain metadata.
workspaceTitle	string	See the attached schema; server-computed or domain metadata.
title	string	See the attached schema; server-computed or domain metadata.
version	string	See the attached schema; server-computed or domain metadata.
method	string	See the attached schema; server-computed or domain metadata.
today	string	ISO date or empty string when unset.

Property	Type / reference	Meaning
startDate	string	ISO date or empty string when unset.
targetDate	string	ISO date or empty string when unset.
baselineTarget	string	ISO date or empty string when unset.
actualDate	string	ISO date or empty string when unset.
forecastDate	string / date / null	See the attached schema; server-computed or domain metadata.
bufferedDate	string / date / null	See the attached schema; server-computed or domain metadata.
scopeHours	number	See the attached schema; server-computed or domain metadata.
earnedHours	number	See the attached schema; server-computed or domain metadata.
remainingHours	number	See the attached schema; server-computed or domain metadata.
bufferedHours	number	See the attached schema; server-computed or domain metadata.
progressPercent	number / null	See the attached schema; server-computed or domain metadata.
plannedProgress	number / null	See the attached schema; server-computed or domain metadata.
assumptionsUsed	boolean	See the attached schema; server-computed or domain metadata.
releaseApproved	boolean	See the attached schema; server-computed or domain metadata.
readyToRelease	boolean	See the attached schema; server-computed or domain metadata.
outlook	string	Values: ahead, on_track, at_risk, behind, needs_data, needs_target, no_scope, released.
readiness	array<object>	See the attached schema; server-computed or domain metadata.
attention	array<object>	See the attached schema; server-computed or domain metadata.
milestones	array<object>	See the attached schema; server-computed or domain metadata.
history	array<object>	See the attached schema; server-computed or domain metadata.

ForecastWork

Property	Type / reference	Meaning
id	integer / int64	See the attached schema; server-computed or domain metadata.
type	string	See the attached schema; server-computed or domain metadata.
title	string	See the attached schema; server-computed or domain metadata.
status	string	See the attached schema; server-computed or domain metadata.
assigneeId	integer / int64	See the attached schema; server-computed or domain metadata.
assigneeName	string	See the attached schema; server-computed or domain metadata.
estimatedHours	number	See the attached schema; server-computed or domain metadata.
progressPercent	number	See the attached schema; server-computed or domain metadata.
recordedHours	number	See the attached schema; server-computed or domain metadata.
remainingHours	number	See the attached schema; server-computed or domain metadata.
bufferedHours	number	See the attached schema; server-computed or domain metadata.
missingEstimate	boolean	See the attached schema; server-computed or domain metadata.
complexity	string	See the attached schema; server-computed or domain metadata.
priority	string	See the attached schema; server-computed or domain metadata.
revision	integer / int64	See the attached schema; server-computed or domain metadata.
remainingOverride	number / null	See the attached schema; server-computed or domain metadata.
canEdit	boolean	False for capacity work in API-key responses.

ForecastTrend

Property	Type / reference	Meaning
date	string / date	See the attached schema; server-computed or domain metadata.
scopeHours	number	See the attached schema; server-computed or domain metadata.
earnedHours	number	See the attached schema; server-computed or domain metadata.
remainingHours	number	See the attached schema; server-computed or domain metadata.
bufferedHours	number	See the attached schema; server-computed or domain metadata.
missingEstimates	integer / int64	See the attached schema; server-computed or domain metadata.

PersonCapacity

Property	Type / reference	Meaning
id	integer	See the attached schema; server-computed or domain metadata.
name	string	See the attached schema; server-computed or domain metadata.
weeklyHours	number	See the attached schema; server-computed or domain metadata.
confirmed	boolean	See the attached schema; server-computed or domain metadata.
weeks	array<CapacityWeek>	See the attached schema; server-computed or domain metadata.
revision	integer / int64	See the attached schema; server-computed or domain metadata.
allocations	array<ResourceAllocation>	See the attached schema; server-computed or domain metadata.

ResourceAllocation

Property	Type / reference	Meaning
workspaceId	integer / int64	See the attached schema; server-computed or domain metadata.
title	string	See the attached schema; server-computed or domain metadata.
hours	number	See the attached schema; server-computed or domain metadata.
revision	integer / int64	See the attached schema; server-computed or domain metadata.
confirmed	boolean	See the attached schema; server-computed or domain metadata.

CapacityWeek

Property	Type / reference	Meaning
week	string / date	See the attached schema; server-computed or domain metadata.
availableHours	number	See the attached schema; server-computed or domain metadata.
allocatedHours	number	See the attached schema; server-computed or domain metadata.
plannedHours	number	See the attached schema; server-computed or domain metadata.
recordedHours	number	See the attached schema; server-computed or domain metadata.
overextended	boolean	See the attached schema; server-computed or domain metadata.
overHours	number	See the attached schema; server-computed or domain metadata.
loadPercent	number / null	See the attached schema; server-computed or domain metadata.
allVisibleAllocatedHours	number	See the attached schema; server-computed or domain metadata.
allVisiblePlannedHours	number	See the attached schema; server-computed or domain metadata.
availabilityRevision	integer / int64	See the attached schema; server-computed or domain metadata.
availabilityOverride	boolean	See the attached schema; server-computed or domain metadata.

FocusConnection

Weekly person/workspace totals. For FocusPeriod connections, week is omitted because the enclosing period supplies it.

Property	Type / reference	Meaning
userId	integer / int64	See the attached schema; server-computed or domain metadata.
workspaceId	integer / int64	See the attached schema; server-computed or domain metadata.
week	string / date	See the attached schema; server-computed or domain metadata.
recordedHours	number	See the attached schema; server-computed or domain metadata.
plannedHours	number	See the attached schema; server-computed or domain metadata.

FocusPeriod

Property	Type / reference	Meaning
week	string	Week start as ISO date, or empty string for the full displayed period.
connections	array<FocusConnection>	See the attached schema; server-computed or domain metadata.

Capacity.reportCharts and project.reportChart are extensible Axis/Etch presentation metadata. They have no write operation in this key API. ReleaseForecast readiness entries contain itemId/title/message; attention entries contain id/title/reason; milestones contain id/title/date/done/approved. Release history contains date, scopeHours, earnedHours, remainingHours and missingEstimates. The attached schemas define these nested objects.

9. MCP access and complete tool catalog

MCP is a separate agent interface at https://askredrover.com/mcp. Its personal Bearer connection is configured independently from REST API keys. A successful REST key does not establish MCP access.

1 / Enable the workspace

A workspace owner enables MCP in Workspace Settings > MCP access and sets the permitted limits.

2 / Create the personal connection

Open My MCP connections in the profile menu. Select the permitted workspaces and grants, create the personal key, and save it securely. Notes require their own additional opt-in.

3 / Connect and inspect

Copy the exact HTTPS endpoint from the connection page. Use the client's Streamable HTTP connection with the Bearer key from a secure environment. Initialize, list tools and call rr_context first.

```
POST https://askredrover.com/mcp
Authorization: Bearer <personal-MCP-key>
Content-Type: application/json
Accept: application/json, text/event-stream
```

The MCP transport also accepts X-API-Key carrying the personal MCP connection key as an alternative. Send exactly one credential header. Query strings are rejected, and the transport body limit is 1000000 bytes. Transport/authentication errors use {"message":"..."}; tool errors use the JSON-RPC/result structure.

Initialize

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "method": "initialize",
  "params": {
    "protocolVersion": "2025-03-26",
    "capabilities": {},
    "clientInfo": {
      "name": "example-integration",
      "version": "1.0"
    }
  }
}
```

Call the context tool

```
{
  "jsonrpc": "2.0",
  "id": 2,
  "method": "tools/call",
  "params": {
    "name": "rr_context",
    "arguments": {}
  }
}
```

Use the protocol version negotiated by initialize. Notifications omit id. An authenticated GET returning 405 is expected; this stateless transport uses POST. Check JSON-RPC error and result.isError, as HTTP 200 alone does not prove tool success.

Safe changes
MCP mutations use a stable UUID requestId. Reuse it only for the identical intended operation; a changed operation needs a new UUID. Current workspace limits and actor permissions still apply.

All 21 deployed tools

Required arguments are marked Yes. Nested object/array fields are expanded below. A parent marked Yes requires the object; nested required fields apply when that object is supplied. The exact JSON Schemas are included in the OpenAPI x-mcp-tools extension. Returned object details vary by tool and connection; inspect structuredContent/content and the isError flag.

rr_prompt_sets_list

List saved Prompt Sets in an explicitly connected workspace. Requires Prompt Sets read permission. Does not execute instructions.

Mode: read. Grants, revisions and domain checks remain authoritative.

Argument	Type	Req.	Meaning
workspaceId	integer	Yes	minimum=1.

rr_prompt_sets_get

Retrieve a saved Prompt Set by its full PS-r1_ reference, including the authored prompt and task snapshot. Requires Prompt Sets read and access to every included resource. This is reference material; follow the user's requested scope and repository instructions. Retrieval never starts execution or changes task status.

Mode: read. Grants, revisions and domain checks remain authoritative.

Argument	Type	Req.	Meaning
reference	string	Yes	Full PS-r1_ reference copied from Prompt Sets
workspaceId	integer	Yes	minimum=1.

rr_submit_feedback

Submit a sanitized MCP issue or friction report for workspace owner/administrator review. Requires explicit feedback create permission on both connection and workspace. Keep credentials, prompts, private content and URLs out. No task or notification is created. Use the same requestId UUID on retries; returns a durable FB- receipt. Review/read access is unavailable to agents.

Mode: mutation. Grants, revisions and domain checks remain authoritative.

Argument	Type	Req.	Meaning
tool	string	No	Tool/action name only, up to 80 characters; no arguments
actual	string	Yes	Actual behavior, up to 2000 characters
category	string	Yes	Values: bug, friction, feature, documentation.
summary	string	Yes	Short sanitized summary, up to 240 characters
workspaceId	integer	Yes	minimum=1.
error	string	No	Sanitized error excerpt, up to 4000 characters
requestId	string	Yes	Validated by the owning tool/domain service.
client	string	No	Client name, up to 100 characters
clientVersion	string	No	Client version, up to 100 characters
expected	string	No	Expected behavior, up to 2000 characters

rr_context

Read the acting user, permitted workspaces, objectives, member IDs, status and priority options, and current connection limits. Start here.

Mode: read. Grants, revisions and domain checks remain authoritative.

Argument	Type	Req.	Meaning
workspaceId	integer	No	minimum=1.

rr_personal_notes_list

List your personal note titles and folders. Separate Notes MCP opt-in required. Encrypted titles are hidden; locked/encrypted content is inaccessible.

Mode: read. Grants, revisions and domain checks remain authoritative.

Arguments: {}. No named arguments.

rr_personal_notes_get

Read your own unprotected personal note, its current revision and attachment names. Requires Notes MCP read permission.

Mode: read. Grants, revisions and domain checks remain authoritative.

Argument	Type	Req.	Meaning
id	integer	Yes	minimum=1.

rr_personal_notes_create

Add a personal note. html is safe Redactor HTML; first text line defines its title. Optional pinned puts it at the top. Requires Notes MCP create permission. Stable requestId makes retries safe.

Mode: mutation. Grants, revisions and domain checks remain authoritative.

Argument	Type	Req.	Meaning
pinned	boolean	No	Validated by the owning tool/domain service.
folderId	integer	No	minimum=0.
requestId	string	Yes	Validated by the owning tool/domain service.
html	string	Yes	Note HTML up to 200000 characters; first line is title

rr_personal_notes_update

Edit an unprotected personal note using its current revision. Supply full html and folderId (0 for no folder). Optional pinned changes pinning; omit to retain it. Protected notes require the Notes UI. Stable requestId makes retries safe.

Mode: mutation. Grants, revisions and domain checks remain authoritative.

Argument	Type	Req.	Meaning
revision	integer	Yes	minimum=1.
folderId	integer	Yes	minimum=0.
html	string	Yes	Full replacement note HTML
pinned	boolean	No	Validated by the owning tool/domain service.
requestId	string	Yes	Validated by the owning tool/domain service.
id	integer	Yes	minimum=1.

rr_personal_notes_delete

Delete your own unprotected personal note and attachments. Requires separate Notes MCP delete permission, current revision and stable requestId. Protected notes require the Notes UI.

Mode: mutation. Grants, revisions and domain checks remain authoritative.

Argument	Type	Req.	Meaning
requestId	string	Yes	Validated by the owning tool/domain service.
id	integer	Yes	minimum=1.
revision	integer	Yes	minimum=1.

rr_brand_get

Read workspace Identity & Brand with its approval status, revision, colors, logos, typography, voice and copy. Requires explicit brand read grant.

Mode: read. Grants, revisions and domain checks remain authoritative.

Argument	Type	Req.	Meaning
workspaceId	integer	Yes	minimum=1.
setId	integer	No	minimum=1.

rr_brand_create

Create a draft workspace brand. Requires explicit brand create grant and identity.manage. Stable requestId makes retries safe.

Mode: mutation. Grants, revisions and domain checks remain authoritative.

Argument	Type	Req.	Meaning
workspaceId	integer	Yes	minimum=1.
requestId	string	Yes	Validated by the owning tool/domain service.
label	string	Yes	Brand set name

rr_brand_save

Save one complete brand section using its current revision. Requires explicit brand update grant and identity.manage. Locked brands cannot change. Approval remains a human workflow.

Mode: mutation. Grants, revisions and domain checks remain authoritative.

Argument	Type	Req.	Meaning
workspaceId	integer	Yes	minimum=1.
setId	integer	Yes	minimum=1.
section	string	Yes	Values: brand, colors, typography, guide, copy, styles, copyTypes.
value	any	Yes	Arbitrary JSON value.
requestId	string	Yes	Validated by the owning tool/domain service.
revision	integer	Yes	minimum=1.

rr_find_work

Find work across explicitly connected workspaces. Default: open work assigned to the acting user. Returns interpreted calendar dates and pagination. Use afterId for the next page.

Mode: read. Grants, revisions and domain checks remain authoritative.

Argument	Type	Req.	Meaning
afterId	integer	No	minimum=1.
from	string	No	ISO calendar date YYYY-MM-DD
priority	string	No	Workspace priority key
status	string	No	open (default), all, or a canonical/custom status key
assignee	string	No	me, all, or a member ID; default me
limit	integer	No	minimum=1; maximum=100.
requirementId	integer	No	minimum=1.
workspaceId	integer	No	minimum=1.
type	string	No	Values: all, task, requirement, release, milestone.
to	string	No	ISO calendar date YYYY-MM-DD
dueWindow	string	No	Values: all, today, next7Days, nextWeek, overdue.
releaseId	integer	No	minimum=1.
search	string	No	Title or description search
timezone	string	No	IANA zone; defaults to this connection's zone

rr_get_item

Read a task or requirement, notes, current version, and tasks within a requirement. Files and financial details are outside this release.

Mode: read. Grants, revisions and domain checks remain authoritative.

Argument	Type	Req.	Meaning
id	integer	Yes	minimum=1.
workspaceId	integer	Yes	minimum=1.

rr_create_items

Create an atomic batch of 1-100 tasks, requirements, releases or milestones. Use a stable requestId UUID on retries. Requirements describe outcomes; tasks describe work. Link existing requirementId/releaseId in fields. RICE: reach, impact, confidence (0-100), effort, reachPeriod (YYYY-Q1 to YYYY-Q4); acceptance defines done. Approval is unavailable.

Mode: mutation. Grants, revisions and domain checks remain authoritative.

Argument	Type	Req.	Meaning
workspaceId	integer	Yes	minimum=1.
requestId	string	Yes	Validated by the owning tool/domain service.
items	array<object>	Yes	minItems=1; maxItems=100.
items[].description	string	No	Description
items[].type	string	Yes	Values: task, requirement, release, milestone.
items[].status	string	No	Use rr_context status options; default draft
items[].fields	object	No	Workspace work fields, including assigneeId, priorityKey, statusKey, startDate, dueDate, requirementId, releaseId and RICE. rr_context provides keys; unknown or protected fields are rejected.
items[].title	string	Yes	Descriptive task or outcome name, up to 240 characters. Never use an ID-only placeholder such as Work 207506 or Task #207506

rr_update_item

Patch work with its current version. Linked requirementId/releaseId and dates live in changes.fields. Notes belong in rr_add_note; avoid replacing descriptions just to log progress. A conflict means reread before changing anything.

Mode: mutation. Grants, revisions and domain checks remain authoritative.

Argument	Type	Req.	Meaning
version	integer	Yes	minimum=1.
changes	object	Yes	Validated by the owning tool/domain service.
changes.title	string	No	Descriptive name; never an ID-only placeholder such as Work 207506
changes.fields	object	No	Workspace work fields, including assigneeId, priorityKey, statusKey, startDate, dueDate, requirementId, releaseId and RICE. rr_context provides keys; unknown or protected fields are rejected.
changes.status	string	No	Canonical status
changes.description	string	No	Replace description only when intended
workspaceId	integer	Yes	minimum=1.
id	integer	Yes	minimum=1.
requestId	string	Yes	Validated by the owning tool/domain service.

rr_add_note

Append a progress note without overwriting prior work. Be factual and concise: work done, evidence, blockers, next action. Plain developer language is applied if enabled; code/logs/quotes stay unchanged.

Mode: mutation. Grants, revisions and domain checks remain authoritative.

Argument	Type	Req.	Meaning
workspaceId	integer	Yes	minimum=1.
requestId	string	Yes	Validated by the owning tool/domain service.
id	integer	Yes	minimum=1.
note	string	Yes	1-12000 characters

rr_remove_item

Archive work through RedRover's normal removal rules. Requires explicit delete grant, disabled by default. Never delete to mark work completed.

Mode: mutation. Grants, revisions and domain checks remain authoritative.

Argument	Type	Req.	Meaning
workspaceId	integer	Yes	minimum=1.
version	integer	Yes	minimum=1.
requestId	string	Yes	Validated by the owning tool/domain service.
id	integer	Yes	minimum=1.

rr_my_time

Read only the acting user's authorized time entries and running timer in this workspace. Optional connected plan blocks. Recorded work is distinct from estimates and scheduled hours.

Mode: read. Grants, revisions and domain checks remain authoritative.

Argument	Type	Req.	Meaning
period	string	No	Values: day, week, month.
from	string	No	ISO calendar date YYYY-MM-DD
to	string	No	ISO calendar date YYYY-MM-DD
workspaceId	integer	Yes	minimum=1.
before	integer	No	minimum=1.
itemId	integer	No	minimum=1.

rr_time

Start/stop your clock, manually log minutes, or annotate an entry. Stop/annotate need entryId and revision from rr_my_time. Log corrections need entryId/revision/reason. Never infer recorded time from a plan. One active clock per person.

Mode: mutation. Grants, revisions and domain checks remain authoritative.

Argument	Type	Req.	Meaning
activity	string	No	What you worked on, required without itemId
requestId	string	Yes	Validated by the owning tool/domain service.
revision	integer	No	minimum=1.
reason	string	No	Reason for a correction
workDate	string	No	ISO calendar date YYYY-MM-DD
workspaceId	integer	Yes	minimum=1.
notes	string	No	Timer annotation; preserves previous work notes
durationMinutes	integer	No	minimum=1; maximum=1440.
entryId	integer	No	minimum=1.
itemId	integer	No	minimum=1.
category	string	No	Activity category, required without itemId
operation	string	Yes	Values: start, stop, log, annotate.

rr_plan_block

Save a future work block for the acting user, cancel it, or start it. Create requires title, startLocal and endLocal. Plans do not record hours. Default autoStart/autoStop false: ask the person to choose; notifyStart/notifyEnd are optional. Server reminders run without the agent connected.

Mode: mutation. Grants, revisions and domain checks remain authoritative.

Argument	Type	Req.	Meaning
operation	string	Yes	Values: create, cancel, start.
startLocal	string	No	YYYY-MM-DDTHH:mm in the chosen timezone
workspaceId	integer	Yes	minimum=1.
blockId	integer	No	minimum=1.
requestId	string	Yes	Validated by the owning tool/domain service.
revision	integer	No	minimum=1.
title	string	No	Use the linked task or release name for this work block; never an ID-only placeholder such as Work 207506
autoStop	boolean	No	Validated by the owning tool/domain service.
notifyEnd	boolean	No	Validated by the owning tool/domain service.
autoStart	boolean	No	Validated by the owning tool/domain service.
endLocal	string	No	YYYY-MM-DDTHH:mm in the chosen timezone
notes	string	No	Plan notes
notifyStart	boolean	No	Validated by the owning tool/domain service.
itemId	integer	No	minimum=1.
timezone	string	No	IANA timezone; defaults to connection

10. Errors, retries and troubleshooting

Status	Meaning	What to do
200	Operation succeeded at the HTTP layer.	REST returns JSON. MCP also requires checking JSON-RPC errors and result.isError.
400	Input rejected.	Check JSON, content type, field names, IDs, supported type/status, limits and type-specific required fields.
401	Missing, malformed, invalid, expired or revoked credentials.	Check exactly one REST credential header, the correct endpoint and a currently valid key. Browser sessions may require sign-in.
403	Authenticated access denied.	Check resource grants, acting user/account, membership, ancestors, current domain rights or session CSRF. HTML 403 may be an edge response; distinguish it from JSON application denial.
409	Version conflict.	Fetch current work, reconcile the change, then submit the new current version.
429	Request limit reached.	Honor Retry-After. The endpoint returns 60 seconds on this limiter; reduce request volume.
503	Service temporarily unavailable.	Retain the failed operation details. Reconcile uncertain writes before attempting another mutation.

An unknown authenticated integration-key path is denied by the route allowlist. Unsupported HTTP verbs can return container-level 405/other responses outside the JSON handler. Do not infer a missing route from a generic edge error.

Common setup problems

Symptom	Likely check
Local-context path fails on production	Use https://askredrover.com/api/v1, without /redrover.
Task missing from a list	Confirm its requirement/release ancestry is readable by both the key and actor.
Goal create returns 400	Include level (for example workspace), positive target and nonblank unit.
Release create returns 400	Include versionLabel and valid date order.
Capacity returns 403	Include milestones:read as well as the other five read grants, and confirm time oversight.
Browser POST returns 403	Use a current same-origin session plus fresh X-CSRF-Token.
Request outcome is uncertain	Read/list the affected record and recent comments; do not blindly repeat create/comment.
REST key fails on /mcp	Create the separate personal MCP connection and use its Bearer credential.

11. Downloadable contracts

Use these full production URLs to download the API guide and machine-readable contracts:

File	Full URL
API reference PDF	https://askredrover.com/assets/api/redrover-api-reference.pdf
OpenAPI JSON	https://askredrover.com/assets/api/openapi.json
OpenAPI YAML	https://askredrover.com/assets/api/openapi.yaml
Postman collection	https://askredrover.com/assets/api/redrover.postman_collection.json

The OpenAPI JSON and YAML describe 14 HTTP operations, 37 named request/response schemas and 21 MCP tools. Import either format into a compatible API client, editor or code generator. The MCP tool argument schemas are included under x-mcp-tools.

Both OpenAPI files are also embedded in this PDF. A PDF reader with attachment support can extract them. The separate downloads are available for clients that cannot read PDF attachments.

Useful application links

https://askredrover.com/developers
https://askredrover.com/api-keys
https://askredrover.com/mcp-access

Key administration and MCP connection management require sign-in and the appropriate current permissions. Use the selected person and workspace permissions for every integration; keys never expand those rights.

Hatchery LLC / RedRover / November 17, 2026