

Issues

Listing issues

```
GET /issues.xml
```

Returns a paginated list of issues. By default, it returns open issues only.

Parameters:

Available as of **1.1.0** :

- **offset**: skip this number of issues in response (optional)
- **limit**: number of issues per page (optional)
- **sort**: column to sort with. Append `:desc` to invert the order.

use this for **"stable 1.0 branch"** (instead of "offset" and "limit"):

- **page**: page number (optional)

Optional filters:

- **project_id**: get issues from the project with the given id, where id is either project id or project identifier
- **tracker_id**: get issues from the tracker with the given id
- **status_id**: get issues with the given status id only. Possible values: `open`, `closed`, `*` to get open and closed issues, status id
- **assigned_to_id**: get issues which are assigned to the given user id
- **cf_x**: get issues with the given value for custom field with an ID of x. (Custom field must have 'used as a filter' checked.)
- ...

Examples:

```
GET /issues.xml
GET /issues.xml?project_id=2
GET /issues.xml?project_id=2&tracker_id=1
GET /issues.xml?assigned_to_id=6
GET /issues.xml?status_id=closed
GET /issues.xml?status_id=*
GET /issues.xml?cf_1=abcdef

Paging example:
GET /issues.xml?project_id=testproject&query_id=2&offset=0&limit=100
GET /issues.xml?project_id=testproject&query_id=2&offset=50&limit=100

To fetch issues for a date range:
GET /issues.xml?created_on=><2012-03-01|2012-03-07
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<issues type="array" count="1640">
  <issue>
    <id>4326</id>
    <project name="Redmine" id="1"/>
    <tracker name="Feature" id="2"/>
    <status name="New" id="1"/>
    <priority name="Normal" id="4"/>
    <author name="John Smith" id="10106"/>
    <category name="Email notifications" id="9"/>
    <subject>
      Aggregate Multiple Issue Changes for Email Notifications
    </subject>
    <description>
      This is not to be confused with another useful proposed feature that
      would do digest emails for notifications.
    </description>
    <start_date>2009-12-03</start_date>
    <due_date></due_date>
    <done_ratio>0</done_ratio>
    <estimated_hours></estimated_hours>
    <custom_fields>
      <custom_field name="Resolution" id="2">Duplicate</custom_field>
      <custom_field name="Texte" id="5">Test</custom_field>
      <custom_field name="Boolean" id="6">1</custom_field>
      <custom_field name="Date" id="7">2010-01-12</custom_field>
    </custom_fields>
    <created_on>Thu Dec 03 15:02:12 +0100 2009</created_on>
    <updated_on>Sun Jan 03 12:08:41 +0100 2010</updated_on>
  </issue>
```

Issues

Listing issues	
Showing an issue	
Using XML	
Using JSON	
Creating an issue	
Using XML	
Using JSON	
Updating an issue	
Using XML	
Using JSON	
Deleting an issue	

```
<issue>
  <id>4325</id>
  ...
</issue>
</issues>
```

Showing an issue

Using XML

```
GET /issues/[id].xml
```

Using JSON

```
GET /issues/[id].json
```

Parameters:

Available as of **1.1.0** :

- **include**: fetch associated data (optional). Possible values: children, attachments, relations, changesets and journals. To fetch multiple associations use comma (e.g ?include=relations,journals). See [Issue journals](#) for more information.

Creating an issue

Using XML

```
POST /issues.xml
<?xml version="1.0"?>
<issue>
  <subject>Example</subject>
  <project_id>1</project_id>
  <priority_id>4</priority_id>
</issue>
```

Other available tags:

- description
- category_id
- assigned_to_id - ID of the user to assign the issue to (currently no mechanism to assign by name)
- status_id
- parent_issue_id - ID of the parent issue
- ...

Attachments can be added when you create an issue, see [Attaching files](#).

Using JSON

```
POST /issues.json
{
  "issue": {
    "project_id": "example",
    "subject": "Test issue",
    "custom_field_values": {
      "1": "1.1.3" #the affected version field
    }
  }
}
```

Updating an issue

Attachments can be added when you update an issue, see [Attaching files](#).

Using XML

```
PUT /issues/[id].xml
```

Using JSON

```
PUT /issues/[id].json
{
  "issue": {
    "subject": "Example issue (was: Test issue)",
    "notes": "Changing the subject"
  }
}
```

Deleting an issue

```
DELETE /issues/[id].xml
```