redmineR
and the story of automating useR!2017 abstract review process

Maxim Nazarov
maxim.nazarov@openanalytics.eu
Redmine

www.redmine.org

Redmine is a flexible project management web application. Written using the Ruby on Rails framework, it is cross-platform and cross-database.

Redmine is open source and released under the terms of the GNU General Public License v2 (GPL).

- Used in Open Analytics for project management, knowledge base (wiki)
- Was used to simplify useR!2017 abstract review process
Redmine: details

Main entities:
- Projects
  - Issues
  - Notes (comments)

Also:
- time entries
- versions, roadmaps, news
- wiki pages, files, ...

Highly customizable:
- user roles and permissions
- workflows
- custom fields, custom issue statuses
Redmine: API

RESTful API available → all common tasks can be performed via API calls

http://www.redmine.org/projects/redmine/wiki/Rest_api

- API clients in several languages available: Ruby, Python, PHP, Java, ...
- uses API keys for authentication
- choice of JSON or XML response formats
redmineR package

- An R client for the Redmine API
  [https://github.com/openanalytics/redmineR](https://github.com/openanalytics/redmineR)
- Based on user-friendly `httr` package and following its "best practices"
  `vignette("api-packages", package = "httr")`
- API functions
  - `redmine_create_issue`, `redmine_update_issue`, `redmine_delete_issue`, ...
- Convenience functions
  - `redmine_search_id` *(search different endpoints to get id's for later use)*
  - `redmine_download_attachments` *(for an issue)*
  - `redmine_issues`, `redmine_projects` *(show issues/projects as a data frame)*
  - ...

redmineR: example code

```r
projectID <- redmine_search_id("testProject", endpoint = "projects")

## redmineR listing for 'projects':
## id   name       identifier
## 17 887 testProject2 test-project-two
## description status
## 17 I *am* a description 1
## created_on updated_on
## parent
## 17 886, changed Name

urgentID <- redmine_search_id("Urgent", endpoint = "enumerations/issue_priorities")

## redmineR listing for 'enumerations/issue_priorities':
## id   name is_default
## 4 6 Urgent NA

issueID <- redmine_create_issue(
  project_id = projectID,
  subject = "Finish useR! presentation",
  description = "Just do it!",
  priority_id = urgentID)

## [1] 12325

redmine_show_issue(issueID)

## redmineR API call:
https://projects.openanalytics.eu/issues/12325.json
## List of 1
## $ issue:List of 12
##  ..$ id : int 12325
##  ..$ project :List of 2
##  ..  ..$ id : int 887
##  ..  ..$ name: chr "testProject2"
##  ..$ tracker :List of 2
##  ..  ..$ id : int 1
##  ..  ..$ name: chr "Bug"
##  ..$ status :List of 2
##  ..  ..$ id : int 1
##  ..  ..$ name: chr "New"
##  ..$ priority :List of 2
##  ..  ..$ id : int 6
##  ..  ..$ name: chr "Urgent"
##  ..$ author :List of 2
##  ..  ..$ id : int 27
##  ..  ..$ name: chr "Maxim Nazarov"
##  ..$ subject : chr "Finish useR! presentation"
##  ..$ description: chr "Just do it!"
##  ..$ start_date : chr "2017-06-27"
##  ..$ done_ratio : int 0
##  ..$ created_on : chr "2017-06-27T14:12:09Z"
##  ..$ updated_on : chr "2017-06-27T14:12:09Z"
```
use(R!2017) case: setup

- A Redmine website was set up for the review process
  - **projects** = tutorials / abstracts / diversity scholarship applications
  - **issues** = submissions
  - for abstracts: **subprojects** per topic with restricted access for reviewers

- Custom user roles: *Reviewer*, *Manager* (Chair), *DSA Reviewer*

- Custom issue statuses and workflows:
  
  *Pending Review* → *First Review* → *Second Review* → *Accepted/Rejected*

- Email notifications (automatically done by Redmine)
use(R!2017) case: R and redmineR

review process:
- extract submission data from MySQL: RMySQL package
- turn submissions into redmine 'issues': redmine_add_issue
- assign to sub-projects: redmine_move_issue
- list all accepted submissions: redmine_issues

scheduling:
- fill-in titles/keywords into custom fields: redmine_update_issue
- download attachments from issues to add to schedule: redmine_download_attachments
**use(R!2017) case: Insights**

Abstracts formats submitted

<table>
<thead>
<tr>
<th></th>
<th>7z</th>
<th>doc</th>
<th>docx</th>
<th>gz</th>
<th>html</th>
<th>md</th>
<th>pdf*</th>
<th>rar</th>
<th>Rmd</th>
<th>txt</th>
<th>zip</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>8</td>
<td>5</td>
<td>15</td>
<td>164</td>
<td>46</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>201</td>
</tr>
</tbody>
</table>

* of those pdf’s with text saved as image: 1

UseR!2017 abstract submissions by day

- Total submissions: 75%
- Text-only submissions: 11%
Lessons learned and take home messages

- If you are organizing a conference:
  ○ remember most of the submissions come last week/day
  ○ provide easy-to-parse templates and limit number of accepted submission formats
  ○ think about fields for auto-parsing (keywords, titles, affiliations)

- If you submit an abstract:
  ○ See if you can make organizers' lives easier :)

- If you want to write an API client in R:
  ○ **httr** provides a quick start
Thank you!

Great appreciation to all the reviewers and especially program committee chairs Ziv Shkedy and Heather Turner

Check out:
- [github.com/openanalytics/redmineR](https://github.com/openanalytics/redmineR)
- [openanalytics.eu](http://openanalytics.eu)
- [redmine.org](http://redmine.org)

Feedback:
- [maxim.nazarov@openanalytics.eu](mailto:maxim.nazarov@openanalytics.eu)