Lecture 6: Project collaboration in GitHub
Previous lectures

- Installing Git
- Configuring Git
- Commiting changes to Git repository
- Working with repository history
- Branching & merging
- Working with merge conflicts
- Using Git to trace changes in code

Enough for typical Git usage
Today

By the end of today’s lecture, you’ll know:

- What is issue tracking
- What is it good for
- What kind of solutions exist
- What project collaboration features GitHub provides
  - Issues
  - Project boards
  - Pull requests
  - Wiki
Beyond code

Your codebase is usually connected to other information and systems

- Documentation - “how stuff works?/what stuff is?”
- Bug reporting - “what needs fixing?”
- Feature requests - “what needs to be added?”
- Development plans - “when will we have it?”

Giving other team members (non-developers) an overview of the state of the project
Issue tracking system

- Keeps track of reported issues in a project.
- Issue - “unit of work to accomplish an improvement in a system”
- Can be bug, feature, improvement (refactoring) or any other task
- Also “task tracking”, “bug tracking”
- Integral part of a project management system
- Consistent use is considered a “hallmark of a good software team”
Different levels of flexibility & integration

1. All-in-one solution
2. Third-party application integrated into an existing solution
3. Separate application
1. All-in-one hosting and project management

- Typically simplest
- Least project management features
- Direct integration with the code repository
- Mostly meant specifically for developers, more tech-oriented
2. Integration on top of an existing solution

- Uses existing project structure
- Adds some information
- Alternative grouping/filtering views
- Good integration to the code repository
- More non-developer friendly
- Still lets developers use the underlying tools
3. Separate application

- Separate concepts, often more flexible
  - Issues can have a complex structure with many fields
  - Different ways to group issues and plan development
- Less integration into the code repository
- Used by big teams
- Better for projects with multiple repositories
- Biggest selection of project management features
GitHub project collaboration tools

- Repository-specific
- Lightweight
- Tightly integrated with the repository
- Enabled by default
- Can be disabled

1. Issue tracking
2. Project boards
3. Pull request management
4. Wiki
Enabling and disabling features

- **Features**
  - **Wiki**
    - GitHub Wiki is a simple way to let others contribute content. Any GitHub user can create and edit pages to use for documentation, examples, support, or anything you wish.
  - **Restrict editing to collaborators only**
    - Public wikis will still be readable by everyone.
  - **Issues**
    - GitHub Issues adds lightweight issue tracking tightly integrated with your repository. Add issues to milestones, label issues, and close & reference issues from commit messages.
  - **Projects**
    - Project boards on GitHub help you organize and prioritize your work. You can create project boards for specific feature work, comprehensive roadmaps, or even release checklists.
Where to find them

Base repository for the practical session of the fifth week of the course on Collaboration Tools in Software Engineering
1. Issue tracker

● Required fields
  ○ ID number (automatically generated)
  ○ Title

● Optional fields
  ○ Description
  ○ Assignees (assigned persons)
  ○ Labels
  ○ Milestones
  ○ Project boards
Welcome to Issues!

Issues are used to track todos, bugs, feature requests, and more. As issues are created, they’ll appear here in a searchable and filterable list. To get started, you should create an issue.

-ProTip! Click a checkbox on the left to edit multiple issues at once.
Stopa / cse-prac5-troubleshooting

Title

Write Preview

Leave a comment

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

Submit new issue

Styling with Markdown is supported
1.1. Title

- Title should give information at a glance
- Not too long
- Project/company policy can enforce rules
- Often issues are named like user stories

As a <role>, I want to <action>, in order to <goal>

As a <user>, I want to <log in>, in order to <make purchases>

- Can be a more free description of the task

Implement login page
Tax calculator crashes with non-numeric data
1.2. Description

- Detailed description of the issue
- Github supports Markdown for rich text
- Can attach files
- Issues can be decomposed into task lists
- Can reference other issues and pull requests
@Stopa please fix #1:

- [ ] two
- [ ] one
- [x] three
Discussion

- All issues can be commented on
- Comments also support Markdown, files, etc
- Users can add “reactions” to comments

queuebit commented 14 days ago

- Is there a way to stop it?
- Is this a problem with other big repos like https://github.com/rails/rails?
- Am I just contributing to the spam by asking this question?
1.3. Assignee

- Issue can be assigned to users or groups
- Up to 10
- Shows who is responsible for the issue
- Assignees get notifications
1.4. Labels

- Issue can have several labels
- To show the status of the issue
- A list of default labels is created
- Can be edited
- Label has: color, description, name
Issue with labels

Another issue #2

@Stopa please fix #1:

- two
- one
- three
<table>
<thead>
<tr>
<th>Label</th>
<th>Description</th>
<th>Edit</th>
<th>Delete</th>
</tr>
</thead>
<tbody>
<tr>
<td>bug</td>
<td>Something isn't working</td>
<td></td>
<td></td>
</tr>
<tr>
<td>duplicate</td>
<td>This issue or pull request already exists</td>
<td></td>
<td></td>
</tr>
<tr>
<td>enhancement</td>
<td>New feature or request</td>
<td></td>
<td></td>
</tr>
<tr>
<td>good first issue</td>
<td>Good for newcomers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>help wanted</td>
<td>Extra attention is needed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>invalid</td>
<td>This doesn’t seem right</td>
<td></td>
<td></td>
</tr>
<tr>
<td>question</td>
<td>Further information is requested</td>
<td></td>
<td></td>
</tr>
<tr>
<td>wontfix</td>
<td>This will not be worked on</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Adding a new label
1.5. Milestone

- Set a deadline for issues
- Set a timeline for development and release
- Have to define them manually, none are created by default
You haven't created any Milestones.

Use Milestones to create collections of Issues and Pull Requests for a particular release or project.

Create a Milestone
Creating a new milestone

New milestone
Create a new milestone to help organize your issues and pull requests. Learn more about milestones and issues.

Title
| title |

Due date (optional)
| pp, kk, aaaa |

Description

Create milestone
Milestones list

2.0 release
Due by June 1, 2018  Last updated less than a minute ago
Bigger, better release
0% complete  0 open  0 closed
Edit  Close  Delete

1.0 release
Due by May 1, 2018  Last updated less than a minute ago
First release of the project
0% complete  0 open  0 closed
Edit  Close  Delete
Milestone list

- Shows how many issues are in each milestone
- How many are closed
Assigning issue to a milestone

Another issue #2

@Stopa please fix #1:

- two
- one
- three

Assignees

Stopa

Labels

- bug
- good first issue

Projects

None yet

Milestone

1.0 release
Populated milestones list

2 Open 0 Closed

1.0 release
- Due by May 1, 2018
- Last updated 1 minute ago
- First release of the project
- 50% complete 1 open 1 closed
  - Edit Close Delete

2.0 release
- Due by June 1, 2018
- Last updated 6 minutes ago
- Bigger, better release
- 0% complete 0 open 0 closed
  - Edit Close Delete
Populated issues list

<table>
<thead>
<tr>
<th>Status</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open</td>
<td>As a &lt;user&gt;, I want to &lt;log in&gt; to &lt;make purchases&gt;</td>
</tr>
<tr>
<td></td>
<td>Test issue</td>
</tr>
</tbody>
</table>

Labels: enhancement, good first issue, help wanted
Filters - the state of the project

- Simple filters from the UI
- Advanced filtering syntax
Closing issues from commits

- Issues can be closed directly from Git
- Reference issue number in commit message

$ git commit -m "closes #123"

- Issue will be closed when commit merged into master
2. Project boards

- Give an overview of the project in your own terms
- Create cards
  - Issues
  - Pull requests
  - Notes
    - Support markdown and todo lists, like issues
- Organise cards into columns (“Open”, “In progress”, “Closed”)
- Create your own columns or use a template
- Move between columns as status changes
- Automation: can close issues when they’re moved to a column etc
- Can include cards from other repositories
Multiple project boards

- Can create multiple project boards in one repository
- Different sprints/releases/milestones
- Different teams
- Different subprojects
Release 1.0
Updated a minute ago

3 To do
- Search for and add issues or PRs to your project
- Manage automation on columns

Added by Stopa

Cards
Cards can be added to your board to track the progress of issues and pull requests. You can also add note cards, like this one!

Added by Stopa

Automation
Automatically move your cards to the right place based on the status and activity of your issues and pull requests.

Added by Stopa

0 In progress

0 Done

Search results

1 As a <user>, I want to <log in> to <make purchases>
#2 opened by Stopa
- enhancement
- good first issue
- help wanted

1 Test issue
#1 opened by Stopa
Cards

To do
- Search for and add issues or PRs to your project
- Manage automation on columns
  Added by Stopa

Cards
- Cards can be added to your board to track the progress of issues and pull requests. You can also add note cards, like this one!
  Added by Stopa

Automation
- Automatically move your cards to the right place based on the status and activity of your issues and pull requests.
  Added by Stopa
3. Pull request management

- Typically changes should not be made to master branch
- Only authorised collaborators can make new branches in open source repositories
- Collaboration workflow:
  - If you’re a collaborator, make new branch
  - If not, fork the repository and make new branch
  - Make the changes
  - When ready, open a pull request (or “merge request”)
  - “Requesting the maintainer to pull in your changes”
Pull request

- Associated with one branch
- Can be associated with issues
  - Issues closed when PR merged
- Title, description, assignees, labels, milestone
- Comments, discussion
Code review

- Reviewers can be added
- Reviewers can approve the PR or request changes
- PR cannot be merged until it is approved
- Code review is important part of agile workflow
Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.

- **base** master ▼
- **compare branch-2 ▼

**Able to merge.** These branches can be automatically merged.

**Add requirements**

Write ▼ Preview

**Leave a comment**

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

- **Reviewers** ▼ No reviews
- **Assignees** ▼ No one—assign yourself
- **Labels** ▼ None yet
- **Projects** ▼ None yet
- **Milestone** ▼ No milestone

- **1 commit** ▼ 1 file changed ▼ 0 commit comments ▼ 1 contributor

Commits on Apr 26, 2018

- Stopa ▼ add 2

Showing 1 changed file with 0 additions and 0 deletions.
• Reviewers get a notification that their input is requested
• Can approve/deny and add comments
  ○ Whole PR
  ○ One file
  ○ Specific lines
lib: remove parameter from catch where not used #20332

danbev wants to merge 2 commits into nodejs:master from danbev:use_empty_catch

This commit removes the exception object from catch clauses where is it not used.

Checklist

- make -j4 test (UNIX), or vcbuild test (Windows) passes
- commit message follows commit guidelines

danbev commented 3 hours ago

danbev added some commits 3 hours ago

nodejs-github-bot added the lib / src label 3 hours ago
lib: remove parameter from catch where not used #20332

This commit removes the exception object from catch clauses where it is not used.

Checklist
- make -j4 test (UNIX), or vcbuild test (Windows) passes
- commit message follows commit guidelines

Reviewers
- TimothyGu
- BridgeAR

Assignees
No one assigned

Labels
- author ready
- dont-land-on-v4.x
- dont-land-on-v5.x
- dont-land-on-v6.x
- dont-land-on-v7.x
- dont-land-on-v8.x
- dont-land-on-v9.x
- lib / src
- refactor to ES6+
lib: remove parameter from catch where not used #20332

- lib/buffer.js

```javascript
try {
    const { InternalBinding } = require('internal/bootstrap/loaders');
    isAnyArrayBuffer = InternalBinding('types').isAnyArrayBuffer;

    catch (e) {
        throw e;
    }
}
```

- lib/events.js

```javascript
function emit(type, ...args) {
    value: enhanceStackTrace.bind(null, er, capture),
    configurable: true
    };

    catch (e) ()
    catch (e)
```

// Note: The comments on the 'throw' lines are intentional, they show
// up in Node's output if this results in an unhandled exception.
lib: remove parameter from catch where not used #20332

Changes from all commits

Submit your review

Review summary

Leave a comment

- Comment
  - Submit general feedback without explicit approval.

- Approve
  - Submit feedback approving these changes.

- Request changes
  - Submit feedback suggesting changes.

Submit review

```javascript
lib/buffer.js

@@ -43,7 +43,7 @@

try {
    const { internalBinding } = require('internal/bootstrap/loaders');
    isArrayBuffer = internalBinding('types').isArrayArrayBuffer;
-} catch (e) {
+} catch (e) {
    isArrayArrayBuffer = require('util').types.isArrayArrayBuffer;
    const {
```

```javascript
lib/events.js

@@ -160,7 +160,7 @@

    value: enhanceStackTrace.bind(null, er, capture),
        configurable: true
    }
-} catch (e) {}
+} catch (e) {} // Note: The comments on the 'throw' lines are intentional, they show
```

// up in Node's output if this results in an unhandled exception.
let isAnyArrayBuffer;

try {
  const { internalBinding } = require('internal/bootstrap/loaders');
  isAnyArrayBuffer = internalBinding('types').isAnyArrayBuffer;
} catch (e) {


isAnyArrayBuffer = require('util').types.isAnyArrayBuffer;
}
```javascript
let isArrayBuffer = require('util').types.isArrayBuffer;
```

Stopa: Pending

Change this back

Reply...
4. Wiki

- Create wiki pages
- Documentation
- Guidelines
- Policies
- Supports Markdown and several other formats
- Managed by a separate git repository
  - Can be cloned and edited locally
  - Each edit has an edit message
  - Can view history of each page
Home

Stepan Bolotnikov edited this page just now • 2 revisions

Sample wiki page

Clone this wiki locally

https://github.com/Stopa/

Clone in Desktop
Recap

We covered

- Issue tracking software in a nutshell
- Different issue tracking software
- Project management tools in Github
  - Issue tracking
  - Project boards
  - Pull requests
  - Wiki
References

- [https://github.com/features/project-management](https://github.com/features/project-management)