

Tech Workshop #3

Braden Guliano

Director of Technical Development

Kappa Theta Pi

Agenda

1. Finish going over the rest of the KTP tech stack
 1. Zustand
 2. React Query
2. Upload new project to GitHub
3. KTP GitHub process
4. Questions

Zustand

- Lightweight
- Minimal boilerplate
 - Versus, i.e. useState
- Easier than Redux
 - ~~Actions~~
 - ~~Reducers~~
 - ~~Types~~
 - ~~Provider~~



React Query

- Library for **server state**
- Simplifies API calls
- Handles loading, error, success easily



Benefits



- Currently, you need to
 - Use `useEffect` and `fetch`
 - Handle loading and error
 - Write caching logic for repetitive API calls
 - Messy + boilerplate intense
- React Query, however,
 - Fetches data automatically when needed
 - Caches results
 - Manages loading/error states

Deploying to GitHub

1. Edit `next.config.js`
2. Create GitHub Actions workflow
3. Change Actions > General to Read/Write
4. Set Pages > Build and deployment > Source to GitHub Actions



KTP GitHub

1. Merge-commit vs squash merge
2. Pull request process



Merge-commit vs squash merge

Merge-commit

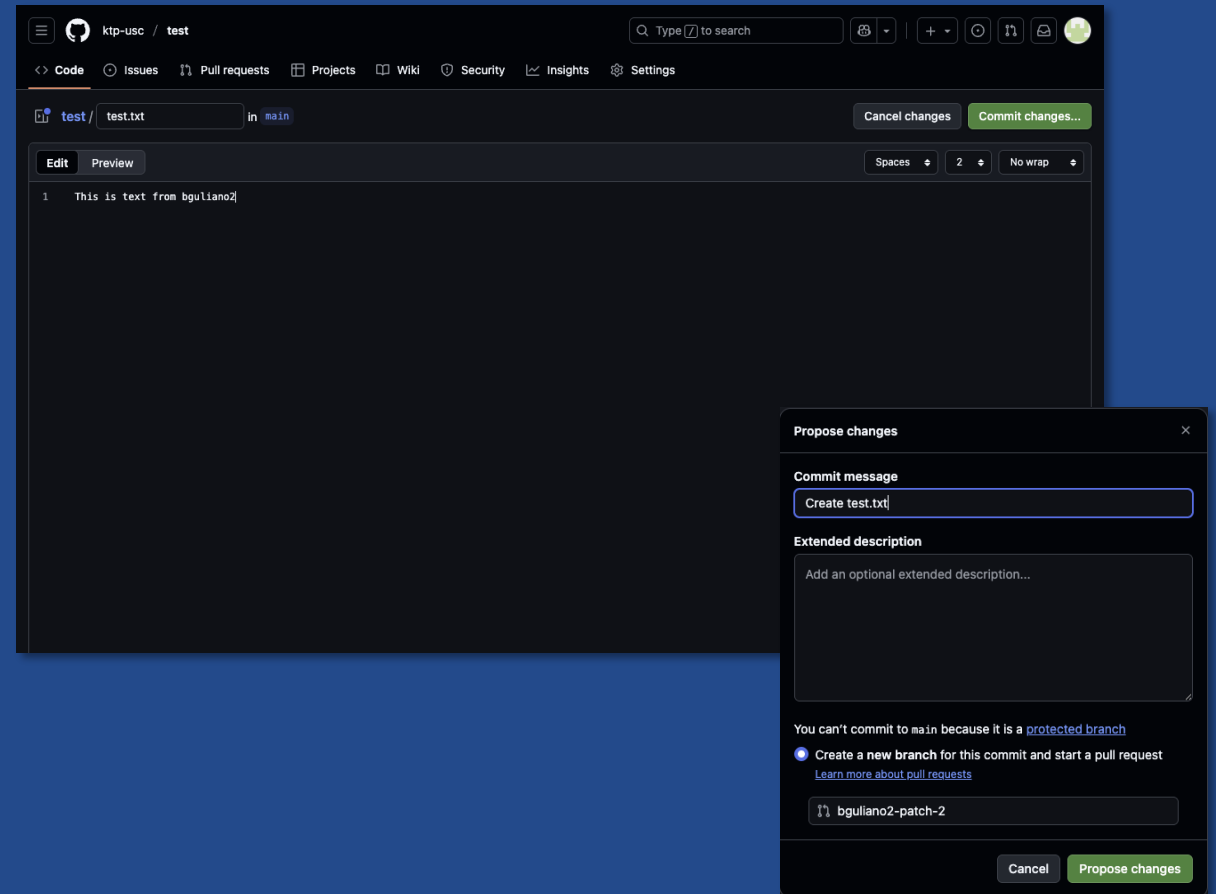
- Preserves all commits, adds a merge commit
- **Pro:** full history
- **Con:** gets messy

Squash merge

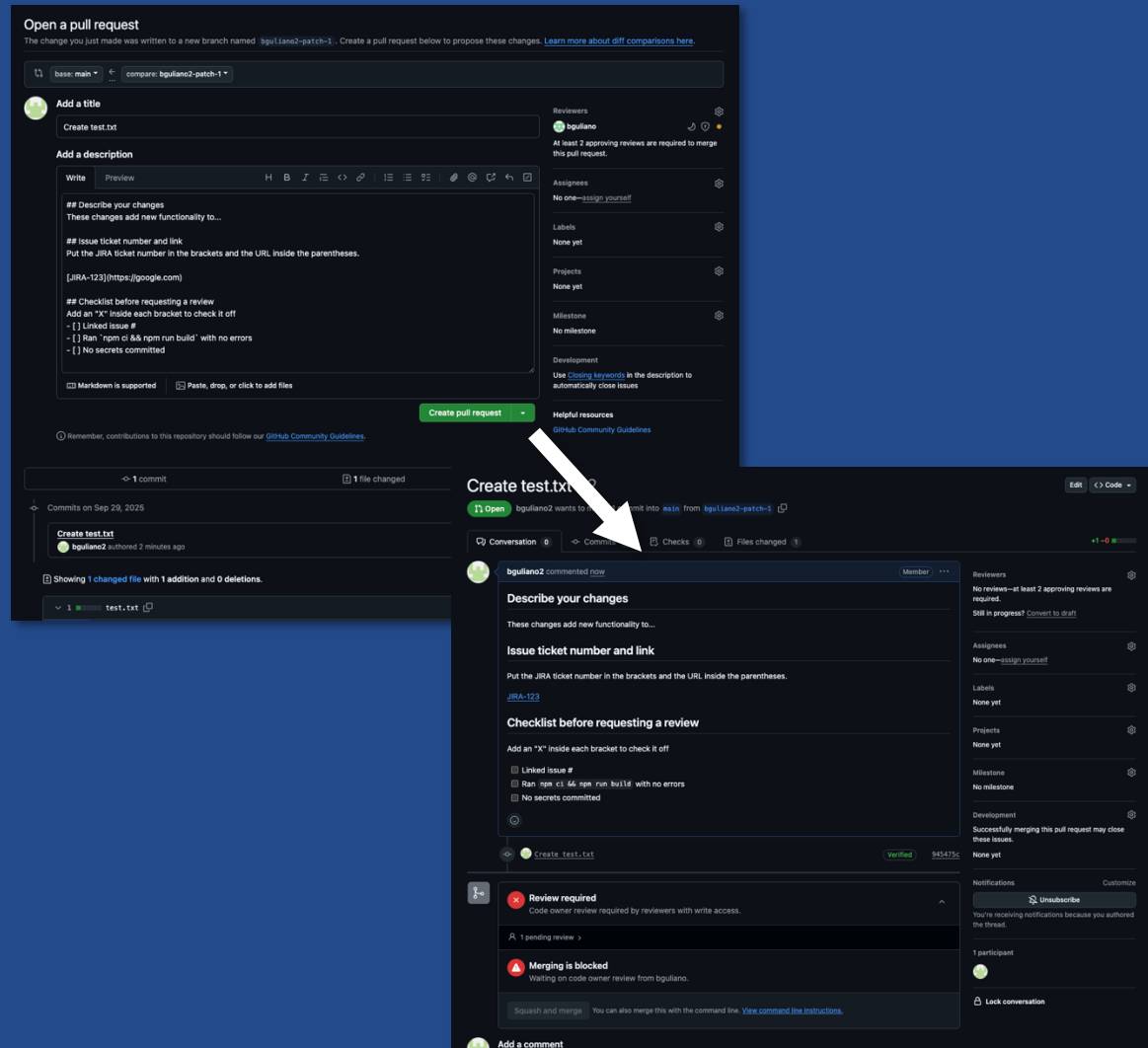
- Combines all PR commits into one commit
- **Pro:** clean history, easy revert
- **Con:** loss of granular steps

Pull request process

- Create a new branch with some sorts of commits
- Here, GitHub does these steps all at once
 - Created a file called test.txt
 - Committed the changes
 - Create a branch for the commit to go to
 - Start a pull request to merge this new branch into main



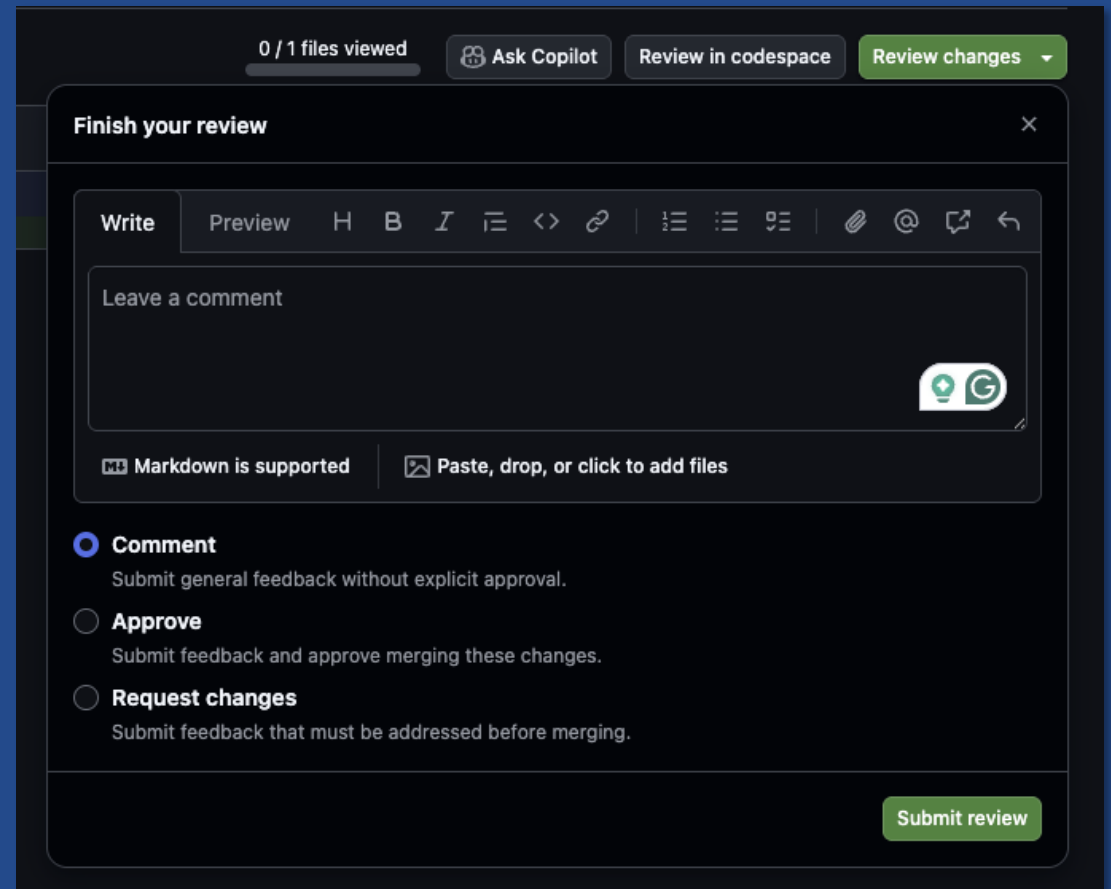
Pull request process



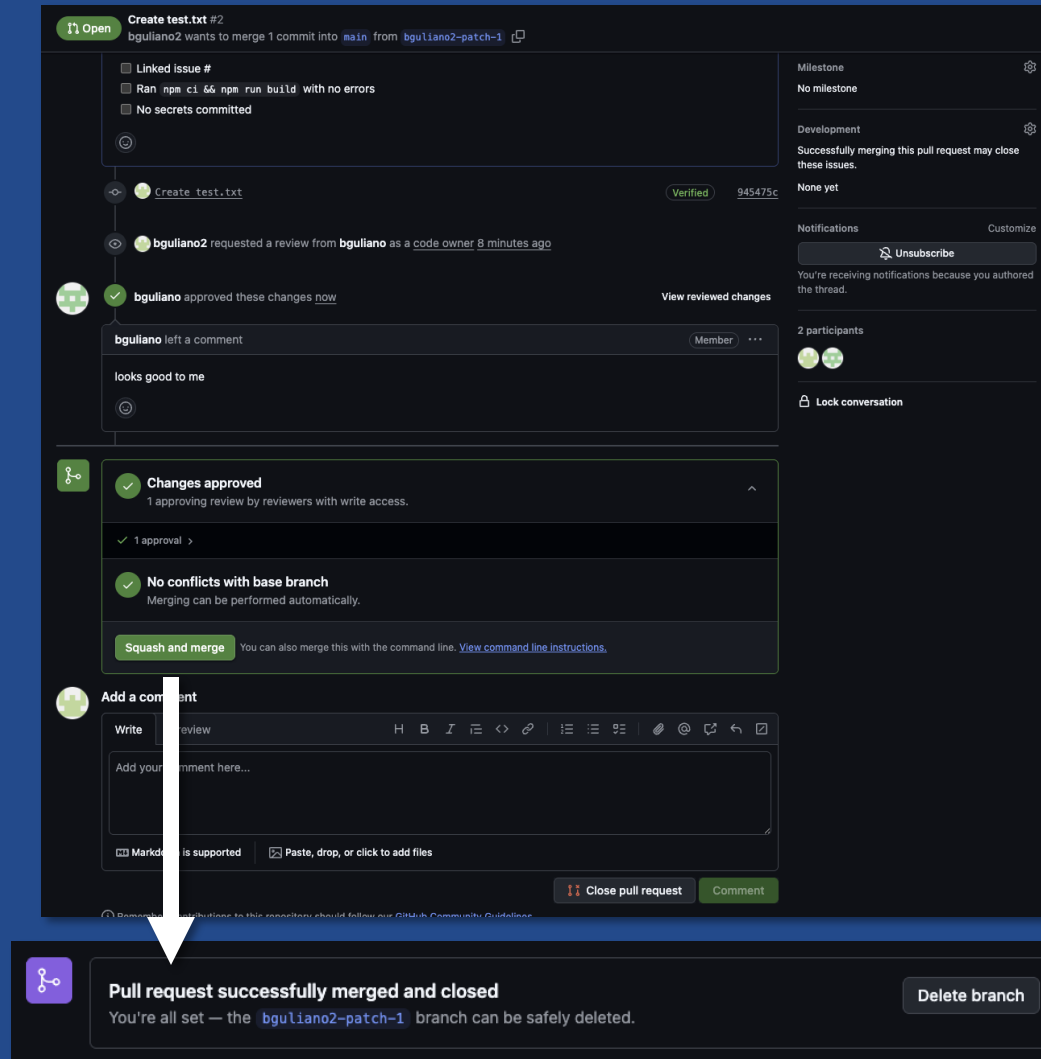
- A pull request is created in GitHub
- A notification is sent to reviewers
 - Project leads
 - Director of Technical Development
- Block is created until changes are approved

Pull request process

- Reviewers ensure code is complete, error-free, and meets project goals
- 2 approvals required to merge
- Can also request new changes to be added to the pull request before merge



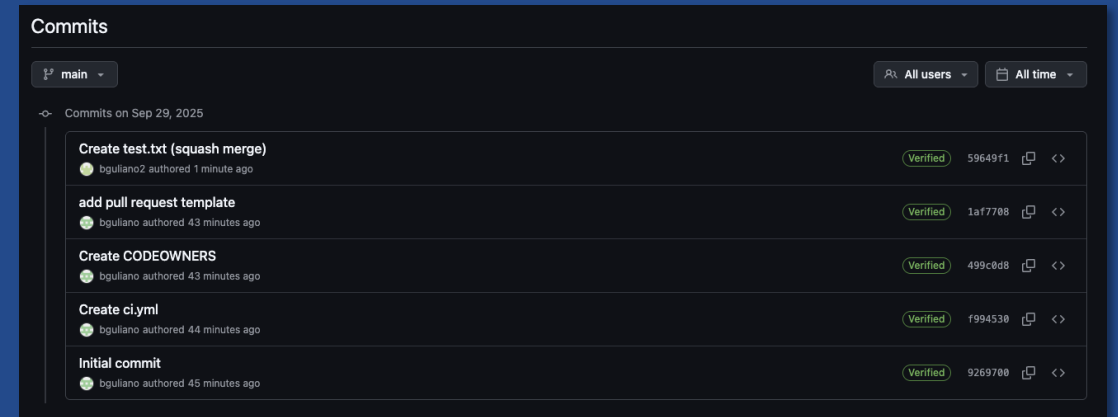
Pull request process



- This chain is where discussion and comments about the changes should live
 - This is because the commits will be deleted, but the pull requests will not
- Once approvals are met, squash merge can occur
- “Squash and merge” button will ask for a commit title

Pull request process

- Commit is converted to squashed commit and placed in GitHub history
- Pull request is moved to “Merged” status



Create test.txt #2

Merged bguliano2 merged 1 commit into `main` from `bguliano2-patch-1` 6 minutes ago

Questions?

