exceeding 100 MB in size. For more information, see "Working

Article versions

- GitHub.com
- GitHub Enterprise 2.3
- GitHub Enterprise 2.2
- GitHub Enterprise 2.1
- GitHub Enterprise 2.0

d to serve as a backup tool. However, there are many d for performing backups that are worth checking out, y andCrashPlan.

Database dumps

Large SQL files do not play well with version control systems such as Git. If you are looking to provide your developers with the most recent production dataset, we recommend using Dropbox for sharing files like these among your developers.

If you are looking to backup your production servers, see the **Backups** section above.

External dependencies

Another thing that causes Git repositories to become large and bloated are external dependencies. It's best to leave these files out of the repository and use a package manager instead. Most popular languages come with package managers that can do this for you. Bundler, Node's Package Manager and Maven. They each support using a Git repository directly as well, so you don't need pre-packaged sources.

Packaged release versions

We don't recommend distributing compiled code and pre-packaged releases within your repository. For more information on sharing large files, see "Distributing large binaries."

Changing history of an existing repository

If you already have a repository that's quite large, don't fret! You can remove large files from the repository's history to reduce its size. For more information, see "Removing files from a repository's history."

- **Contact a human**