## Contribute

You can help this project by reporting problems, suggestions, localizing it or contributing to the code.

### Report a problem or suggestion

Go to our [issue tracker](https://github.com/zhukov/webogram/issues) and check if your problem/suggestion is already reported. If not, create a new issue with a descriptive title and detail your suggestion or steps to reproduce the problem.

### Localization

If you don't see your native language available for Webogram and you can help with translation, please consult the [Telegram Translations Manual](https://core.telegram.org/translating\_telegram).

To test your translation live, use [Localization guide](/app/js/locales/README.md).

### Contribute to the code

If you know how to code, we welcome you to send fixes and new features, but in order to be efficient we ask you to follow the following procedure:

\* Fork this repo using the button at the top.

\* Clone your forked repo locally.

``$ git clone git@github.com:yourname/webogram.git``

\* Don't modify or work on the master branch, we'll use it to always be in sync with webogram upstream.

```

$ git remote add upstream git@github.com:zhukov/webogram.git

$ git fetch upstream

```

\* Always create a new issue when you plan to work on a bug or new feature and wait for other devs input before start coding.

\* Once the new feature is approved or the problem confirmed, go to your local copy and create a new branch to work on it. Use a descriptive name for it, include the issue number for reference.

``$ git checkout -b improve-contacts-99``

\* Do your coding and push it to your fork. Include as few commits as possible (one should be enough) and a good description. Always include a reference to the issue with "Fix #number".

```

$ git add .

$ git commit -m "Improved contact list. Fix #99"

$ git push origin improve-contacts-99

```

\* Do a new pull request from your "improve-contacts-99" branch to webogram "master".

#### How to implement changes suggested on a pull request

Sometimes when you submit a PR, you will be asked to correct some code. You can make the changes on your work branch and commit normally and the PR will be automatically updated.

``$ git commit -am "Ops, fixing typo"``

Once everything is OK, you will be asked to merge all commit messages into one to keep history clean.

``$ git rebase -i master``

Edit the file and mark as fixup (f) all commits you want to merge with the first one:

```

pick 1c85e07 Improved contact list. Fix #99

f c595f79 Ops, fixing typo

```

Once rebased you can force a push to your fork branch and the PR will be automatically updated.

``$ git push origin improve-contacts-99 --force``

#### How to keep your local branches updated

To keep your local master branch updated with upstream master, regularly do:

```

$ git fetch upstream

$ git checkout master

$ git pull --rebase upstream master

```

To update the branch you are coding in:

```

$ git checkout improve-contacts-99

$ git rebase master

```