

# FINkers United *for dummies*





First time seeing a “for Dummies” brochure, the expression seemed a bit offensive. Even now, we thought twice whether to name this “for Dummies” or “FINkers United ABCs”. But reading this kind of brochure one can understand the value of it: having things explained, plain and simple, common language, without all the metaphors or exaggerated sales propaganda; having a concise, straight answer to the questions that can pop-up in anybody’s mind.

So that’s what we want do for FINkers United.

## Chapter 1: General Aspects

### What is FINkers United?

FINkers United® is an [open source community](#) for all financial thinkers’ enthusiasts – from IT developers to financial experts – gathered around the FinTP® Project.

PS: this also answers another question: where does the name come from? Well, from financial thinkers united.

### What is FinTP?

[FinTP](#) is an open source financial transactions processing application meant to be used as an open platform in a strictly regulated market.

FinTP aims to push financial transactions interoperability forward – from the current message syntax standard layer towards transactions interpretation – thus attaining a higher semantic standardization layer.

### What is the purpose of this community?

It’s to encourage creativity in the financial industry, to create an environment where ideas are seeded and brought to ripeness to the benefit of all. It is aimed at freeing people’s creativity and brains from reinventing the wheel, as it happens

in a closed traditional environment, and allowing them to take advantage of all the benefits an open approach brings.

In fewer words, it aims to grow the FinTP application that Allevo has provided the code for through contributions at any level, from requirements to be implemented to various operating systems compatibility, support applications and so on.

### **Who can become a member?**

Anyone who has an interest in this area, likes the idea, is passionate about developing in open source or is keen in taking part in building an innovative product may become a member. All you need to do is register on [www.fintp.org](http://www.fintp.org). Actually, there is no community if it does not have purpose and this is given by its members.

All of you are welcome to our community, to use our code, to praise or criticize us, and give us feedback. Users who continue to engage with the project and its community will often become more and more involved. Such users may find themselves becoming contributors.

### **What is a contributor?**

Contributors are community members who wish to take part in concrete ways in the community running. Their contributions can take many forms, but all are to be submitted as patches that should be validated by committers or software maintainers.

The types of contributors are detailed in Chapter 2: [Structures and roles](#), together with how you can contribute, how your role can grow in the community and what's higher up the chain.

## If I do not have technical skills, could I still be part of it?

But of course you can. Any idea is welcome and any member's knowledge is valuable to the community. For example, you might have an idea of a feature or new project you think is useful and worth adding over FinTP.



You can also contribute by sharing your expertise in discussions on the Forum, the designated space for all debates

within the community.

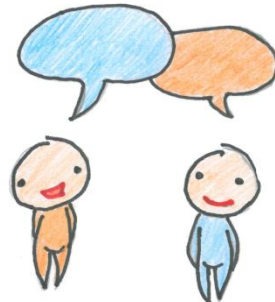
All contributions will be acknowledged and thanked for.

## Oh, but where can I find the code?

We used the [GitHub](https://github.com) portal as the repository for the FinTP source code. But you can also find the binaries on the community's web page, [www.fintp.org](http://www.fintp.org).

## It sounds interesting. Where can I read more about this?

On [www.fintp.org](http://www.fintp.org), the community's portal. You can read about governance, structure and roles, meritocracy, voting, code of conduct and so on. And there are a lot of topics that might interest you on the forum.



## Chapter 2: Structure and Roles

Now that we have clarified some general aspects like what is FINkers United, what is its purpose, how can you become a member and so on, it is time to explain what roles you can have in this community and how you can actually contribute.

### How can I contribute?

There are many ways you can contribute and thus become part of this living community. What you can do depends on your choice of involvement. The more dedicated you become, the more possibilities will open to you. Depending on your level of involvement, you can be a member, a contributor (individual or corporate) or a committer.

### Can you detail the member part?

Anyone can become member of the community. You just have to be interested in what we are doing. You might very well be a user who downloaded and used our code and now just feel the need to give us your feedback. You might want to stay in touch with forum discussions, latest news and events or just burst with new ideas and suggestions.

Apart from that, members usually provide feedback on their experience using our outcomes, inform our developers of strengths and weaknesses they found and identify possible new features, support new users (share the experience they gained).

**Note:** When becoming a member you will be asked to agree to our code of conduct.

### How about the contributors?

Community members who wish to take part in concrete ways in the community running may become contributors, and contributions can take many forms. There is no expectation of commitment to the project, no specific skill requirements and no actual selection process - all you have to do is to agree

upon the terms and conditions of the CLA (Contribution License Agreement). Contributors engage with the project through forum, blog, Idea box, projects, and mailing lists.

When we talk of contributors though, we need to make the distinction between an individual contributor and a corporate contributor.



their contributions will always be acknowledged.

*Individual contributors* are the members willing to give away the outcomes of their activity within the community, no matter if they are a requirements designer, architect, developer, tester, document writer, or translator, you name it. Their artefacts will be submitted as patches and validated by committers or software maintainers, but



avoid any copyright issues that may rise. This will come together with a list of employees the company agrees to have as contributors.

*Corporate contributors* are those companies willing to empower their employees to become contributors within our community. This way they will always be in the middle of things and also have the possibility of influencing the paths the projects are going. All they have to do is to agree the terms and conditions of the CCLA (Corporate Contribution License Agreement) in order to

## What can I, as a contributor, do more than a simple member?

There are several things a member cannot do, but a contributor can, such as: analysing and designing requirements, adding features, programming, fixing bugs, writing documentation, software localization, providing graphics and web design, etc.

You said something about a... committer, was it? What is that?

Generally, a potential committer is a contributor who has shown an understanding of the project, its objectives and its strategy and has provided valuable contributions to the project over a period of time. Consequently, the community will return his/her trust and (s)he may be nominated for committership by an already existing committer (initially, these are the software maintainers). Committers have no more authority over the project than contributors, as their work continues to be reviewed by the community before acceptance in an official release. The key difference between a committer and a contributor is when this approval is sought from the community: a committer seeks approval after the contribution is made, rather than before. Seeking approval after making a contribution is known as a commit-then-review process. The candidates for committership are voted within the project. Once the vote has been held, the aggregated voting results are published. Nominees may decline their appointment as a committer.

A committer who shows an above-average level of contribution to the project, particularly with respect to its strategic direction and long-term health, may be nominated to become software maintainer, and thus part of the Technical Committee.

Ah, so many new things you mentioned! If I understand correctly, I can be more than a committer in FINKers United?

Yeap, you can be a software maintainer. Software maintainers are contributors with leading role in managing the project assigned. Their role is to ensure the smooth running of the project by reviewing code contributions, participating in strategic planning, deciding upon matters for which the consensus cannot be reached, making sure that all governance processes are adhered to. The software maintainers have access to the project's private sections and its archives (used for sensitive issues, such as votes for new committers and legal matters that cannot be discussed in public). Any new



software maintainer is nominated by the existing software maintainers and is voted together with the project members. Technical committee may cast a veto. The software maintainers remain in role until they retire or the majority vote removes any of them.

### **You keep mentioning this voting process. What's it about?**

Issues such as those affecting the strategic direction or legal standing of the project must gain explicit approval in the form of a vote. Besides those there is also a not so dramatic voting request that will help us to prioritize projects, ideas to become new features, etc.

Every community member is expected to share his/her opinion on all matters and so they will simply cast votes. However, only the meritorious members will have a binding vote for the purpose of decision making (software maintainers, committers, Steering Committee members). This does not mean that you cannot make a difference, even if you are none of the above mentioned meritorious members, as a well justified vote may get supported by a binding vote.

A veto can be expressed only by binding votes. Any objection is to be addressed by the community until the objection is either revoked, overruled (in the case of non-binding votes) or the proposal is so altered as to achieve consensus. If no consensus is achieved, then Steering Committee may decide upon it.

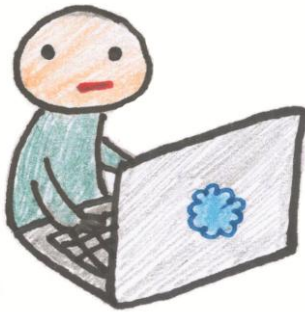
### **I got it, but there is something I want to get back to. What is the technical committee?**

The technical committee includes all the software maintainers. They are entitled to cast binding votes, can nominate committers and have a representative within the Steering Committee of the community. They may approve changes to the governance model and manage the copyrights within the project outputs and decide on matters that affect more projects.

## Is it anything higher up the chain?

Yes, as I mentioned, there is the steering committee. The steering committee is a group of decision makers and has the role of the moderator: the instrumental inspirational authority, expected to draw both the development vision and the guidelines of the community.

## Chapter 3: The coding experience



Time to see how you can actually work in the community, what are the standard processes and so on. For those of you who want to amaze us with your pieces of code, here are some answers to the questions you may have.

### First of all, where can I find the code?

You can either access it on [fintp.org/Projects](http://fintp.org/Projects) or directly on [github.com/FinTP](https://github.com/FinTP). The code is organized in projects. Before starting anything, it might help you to consult the Readme file (at least, it will help you set up your development environment) or just talk to our software maintainer. The dedicated wiki pages include the corresponding documentation.

### Can I do anything I want with it?

Once you download it you can do whatever you like with it. As long as you follow the terms and provisions of the GPLv3 license under which the source code is published, of course. We sure hope you will be willing to give back to the community, in which case you will need to read furthermore.

## What about if I am not quite the guru in coding?



No worries, FINKers United is open to all enthusiasts. You can learn with us: you can get acquainted with the code documentation; you can download the code and have a look through it; you can discuss with the software maintainer; you can address any observations you might have and ask any questions; you can participate in forum discussions; you can get advices from more experienced developers.

And, with time passing, your enthusiasm and some effort, of course, you can even become a guru.

## Ok, let's say I am experienced or already gained experience within FINKers United: what's the development process like?

Well, one thing is for sure, there are no firetraps or black holes. Once you choose what you would like to work on, just start doing what you know best. Depending on the item chosen there might be some things you would like to know upfront. For example, if you decide to work on a feature that is going to be released at a certain date, you could consider letting us know from time to time how things are going.

Other things you might want to work on are solving issues, improving already existing functionalities, integrating parts of the application with third party pre-requisites or applications. In any of these cases assign the issues/tasks to yourself. These are defined per each project.

Oh, one more thing: to smooth the integration process (your piece of code with the already existing one) there are a few things we need from you, besides the source code of course :), code documentation and an unit tests report. We need them to help us review the code and also keep the documentation up to date. So, when you are finished with all these, just put the source code in a fork and make a pull request. If not already

acquainted with it, you can find all about it on <https://help.github.com> or you can always ask for guidance from the software maintainer.

### And how about the code review, how does that go?

During the code review, the software maintainer might start a discussion with you to clarify any aspects that may appear. There may be the need for some rework or some extra information just to have the right understanding of it. But we'll keep you updated on the review's results, don't worry.

Now a little bit of officialdom: in case your contribution is accepted, before the code becomes actual part of FinTP, you will need to sign a CLA (Contribution License Agreement) or a CCLA (Corporate Contribution License Agreement), as described in chapter 2. These documents are available on [fintp.org](http://fintp.org) and concern all contributions, irrespective of project and type of contribution. They are intended only to prevent any issue that might arise from intellectual property infringement. From that moment on you can call yourself a proud contributor to the FinTP project.

### Just to summarize, the phases are...?

The phases of the development process are:

- *Initiate project.* A project is a set of interrelated activities intended to cover a specific area and reach a certain goal. Projects can be proposed by you, as a member of the community or by a group. The one who proposes becomes the Initiator. Each project will have an approach strategy: the main development iterations (partial code commitments), the timelines for the entire code commitment and testing, release criteria, branching. The planning of the projects starts loosely with every involved party's input and once agreed it becomes a baseline for release management. You can find the project description and other project documentation on the portal in a dedicated section. A responsible for each project - the Project Manager (PM) - is assigned.

- *Plan a new release.* Generally, the releases are periodical. Special situations may occur when they are event-triggered. As the project evolves, new requirements and change proposals from you or other contributors/users may appear or requirements may be postponed, but all these will be transparent to the project's contributors. The new requirements and change requests are reviewed and approved/rejected/postponed by the Technical Committee. When the content of a release has been defined, the Project Manager launches the new release iteration, publishes the product specifications and invites contributions. Tasks are proposed and placed in the task pool.
- *Develop, Test, Build.* To contribute to a specific project, you must first adhere to the project team. You and any contributor to the project have to acknowledge the project specific constraints, if any, and the release planning. You are free to choose what you will do: new code, documentation, testing. You will choose your tasks and communicate in due time any prerequisites, if not already available, and their estimated completion time and dependencies, if applicable. It would be nice if you could inform your software maintainer, on a regular basis, about the progress of the items you are working on, so that the release conditions are met. Project generic documentation is available to any interested party.
- *Release the community version.* Work on the new release stops as planned and you, along with the other contributors, are invited to send your latest work to be validated by the software maintainers. The components are integrated, using the release branch. You are invited to test the Community Release. Test results are made available and a plan for fixing emerging issues is drawn and implemented. When the release criteria are met, the community code is published and a new baseline is created.
- *Project management & control.* The Project Manager follows the progress of the project in terms of timeline and quality using your reviews and the ones made by other contributors, and monitoring task deadlines. If you or anyone else fails to follow the community rules/project constraints, you'll receive a warning. If repeated, the Technical Committee can decide other steps. The Project Manager acts as the

interface between the Technical Committee and the project team and moderates the project communications.

### **And what exactly can I develop?**

Based on your interests, you can access any of the already defined projects and choose from the issues/tasks in the issues tracker. You can also consult the list of ideas (<http://www.fintp.org/projects/idea-list/>) and start a discussion around them. Once you started working on a more complex idea, you may find out that you will have to work with more components, so you will actually need to download all the corresponding projects. Feel free to do that. Basically, when you make a pull request, do it within all the projects where you added/modified the code. If you feel a new project should be created, please talk to one of the software maintainers. Most probably there will be a discussion on the opportunity of defining a new project. You can even be proposed to become its software maintainer.

### **And if my piece of code gets accepted?**

As we said, if your code passes the review process and becomes an actual part of FinTP, you will become and can start calling yourself a proud contributor to the FinTP project.

Remember, all contributions are acknowledged and given credit for.

### **Cool, let's get to work!**

This concludes the FINkers United for Dummies. We hope you found it useful. For more information please visit [www.fintp.org](http://www.fintp.org).

[www.fintp.org](http://www.fintp.org)

create

share

inspire