# Coq Platform Docs:
# Documenting Coq and its Platform

**Thomas Lamiaux**, LS2N, Inria & Nantes University
Pierre Rousselin, LAGA, Université Sorbonne Paris Nord
Théo Zimmermann, LTCI, Télécom Paris, Polytechnic Institute of Paris

October 2024

# Coq Platform Docs

# Coq Platform Docs

Why does action-oriented
documentation matter?

## To keep in mind

1. Having a proper, clean and accessible documentation is one of the keys to the success of software: <u>to get and keep users</u>

# Introduction

## To keep in mind

1. Having a proper, clean and accessible documentation is one of the keys to the success of software: <u>to get and keep users</u>
2. There are different forms of documentation:
   - Abstract and detailed documentation like the ref man
   - Course-shaped documentation like Coq'Art or SF
   - Action oriented documentation like tutorials or how-to guides

## To keep in mind

1. Having a proper, clean and accessible documentation is one of the keys to the success of software: <u>to get and keep users</u>
2. There are different forms of documentation:
   - Abstract and detailed documentation like the ref man
   - Course-shaped documentation like Coq'Art or SF
   - Action oriented documentation like tutorials or how-to guides

## A lack of action-oriented documentation

1. When it exists, it is often incomplete or outdated

## To keep in mind

1. Having a proper, clean and accessible documentation is one of the keys to the success of software: <u>to get and keep users</u>

2. There are different forms of documentation:
   - Abstract and detailed documentation like the ref man
   - Course-shaped documentation like Coq'Art or SF
   - Action oriented documentation like tutorials or how-to guides

## A lack of action-oriented documentation

1. When it exists, it is often incomplete or outdated
2. When it exists, it is often spread out over many repos / blogs

# What is Action Oriented Documentation ?

## Tutorials

Tutorials help users to learn a new feature, step by step, introducing notions gradually through predetermined examples:

$\hookrightarrow$ Searching for Lemma, Combining Tactics, etc...

# What is Action Oriented Documentation ?

## Tutorials

Tutorials help users to learn a new feature, step by step, introducing notions gradually through predetermined examples:

$\hookrightarrow$ Searching for Lemma, Combining Tactics, etc...

## How-to guides

How-to guides provide users with step by step resolution procedures for real life problems, but without touring all that is possible:

$\hookrightarrow$ How to search for a lemma

$\hookrightarrow$ How to set up a Coq Project

# What is Action Oriented Documentation ?

## Tutorials

Tutorials help users to learn a new feature, step by step, introducing notions gradually through predetermined examples:

↪ Searching for Lemma, Combining Tactics, etc...

## How-to guides

How-to guides provide users with step by step resolution procedures for real life problems, but without touring all that is possible:

↪ How to search for a lemma

↪ How to set up a Coq Project

## To Remember

1. Tutorials and how-guides are different and complementary

# What is Action Oriented Documentation ?

## Tutorials

Tutorials help users to learn a new feature, step by step, introducing notions gradually through predetermined examples:

↪ Searching for Lemma, Combining Tactics, etc...

## How-to guides

How-to guides provide users with step by step resolution procedures for real life problems, but without touring all that is possible:

↪ How to search for a lemma

↪ How to set up a Coq Project

## To Remember

1. Tutorials and how-guides are different and complementary
2. They are both about practical knowledge and achieving goals

### To Remember

Tutorials and How-to guides cover different aspects of learning

# Why is it important for Learning

## To Remember

Tutorials and How-to guides cover different aspects of learning

## Tutorials

Tutorials are about learning something new:

− They provide a controlled learning environment
  ↪ simplified examples, step by step explanations, etc...
− They only discuss one specific feature or aspect of a feature
− It tours a feature without trying to be exhaustive as a ref man

# Why is it important for Learning

## To Remember

Tutorials and How-to guides cover different aspects of learning

## Tutorials

Tutorials are about learning something new:

- They provide a controlled learning environment
  - ↪ simplified examples, step by step explanations, etc...
- They only discuss one specific feature or aspect of a feature
- It tours a feature without trying to be exhaustive as a ref man

## How-to guides

How-to guides bridge the gap between learning and doing:

- To find help when stuck on a real-life situation
- To see how to use a feature in a non trivial context

# It is important for Teaching

### When Teaching

1. A lot about Coq is pratical knowledge and can be subtle
2. It is not possible to be behind every student helping them
3. You cannot teach everything in a 24h/48h class:
   - There are too many features, or too many aspects to discuss
   - Some features are too advanced, or too technical to discuss

# It is important for Teaching

## When Teaching

1. A lot about Coq is pratical knowledge and can be subtle
2. It is not possible to be behind every student helping them
3. You cannot teach everything in a 24h/48h class:
   - There are too many features, or too many aspects to discuss
   - Some features are too advanced, or too technical to discuss

## Action-oriented Documentation

1. Tutorials enable students to check and expand their knowledge through learning content and examples

# It is important for Teaching

## When Teaching

1. A lot about Coq is pratical knowledge and can be subtle
2. It is not possible to be behind every student helping them
3. You cannot teach everything in a 24h/48h class:
   - There are too many features, or too many aspects to discuss
   - Some features are too advanced, or too technical to discuss

## Action-oriented Documentation

1. Tutorials enable students to check and expand their knowledge through learning content and examples
2. How-to guides provide students with practical examples helping them to debug themselves when working on projects

# It is important for Teaching

## When Teaching

1. A lot about Coq is pratical knowledge and can be subtle
2. It is not possible to be behind every student helping them
3. You cannot teach everything in a 24h/48h class:
   - There are too many features, or too many aspects to discuss
   - Some features are too advanced, or too technical to discuss

## Action-oriented Documentation

1. Tutorials enable students to check and expand their knowledge through learning content and examples
2. How-to guides provide students with practical examples helping them to debug themselves when working on projects
3. Action-oriented documentation enables students to continue learning after class, and to get towards autonomy in Coq

# It is important for our Community

## For users

1. It enables users to learn and discover new features on their own, and to debug themselves when they are stuck

### For users

1. It enables users to learn and discover new features on their own, and to debug themselves when they are stuck
2. It enables users to find specific practical knowledge

# It is important for our Community

## For users

1. It enables users to learn and discover new features on their own, and to debug themselves when they are stuck
2. It enables users to find specific practical knowledge
3. It enables us to turn folklore into knowledge, and to make it accessible to users that would not know it otherwise

# It is important for our Community

## For users

1. It enables users to learn and discover new features on their own, and to debug themselves when they are stuck
2. It enables users to find specific practical knowledge
3. It enables us to turn folklore into knowledge, and to make it accessible to users that would not know it otherwise

## For experts

1. It makes our work easier to discover and more accessible

# It is important for our Community

## For users

1. It enables users to learn and discover new features on their own, and to debug themselves when they are stuck
2. It enables users to find specific practical knowledge
3. It enables us to turn folklore into knowledge, and to make it accessible to users that would not know it otherwise

## For experts

1. It makes our work easier to discover and more accessible
2. It helps getting new users, and getting them more experts

# It is important for our Community

## For users

1. It enables users to learn and discover new features on their own, and to debug themselves when they are stuck
2. It enables users to find specific practical knowledge
3. It enables us to turn folklore into knowledge, and to make it accessible to users that would not know it otherwise

## For experts

1. It makes our work easier to discover and more accessible
2. It helps getting new users, and getting them more experts
3. It enables us to put forward and promote good practices

# It is important for our Community

## For users

1. It enables users to learn and discover new features on their own, and to debug themselves when they are stuck
2. It enables users to find specific practical knowledge
3. It enables us to turn folklore into knowledge, and to make it accessible to users that would not know it otherwise

## For experts

1. It makes our work easier to discover and more accessible
2. It helps getting new users, and getting them more experts
3. It enables us to put forward and promote good practices
4. Writing documentation forces us to do stuff right (no cheating)

# Coq Platform Docs

## The Project

### Coq Platform Docs

1. Each feature and plugin of Coq and its platform would have one or several <u>interactive tutorials and how-to guides</u>

## Coq Platform Docs

1. Each feature and plugin of Coq and its platform would have one or several interactive tutorials and how-to guides
2. It would be indexed on the Coq Platform to be easy to use

### Coq Platform Docs

1. Each feature and plugin of Coq and its platform would have one or several <u>interactive tutorials and how-to guides</u>
2. It would be <u>indexed on the Coq Platform</u> to be easy to use
3. Possible to extend with explanations or learning games

### Coq Platform Docs

1. Each feature and plugin of Coq and its platform would have one or several interactive tutorials and how-to guides
2. It would be indexed on the Coq Platform to be easy to use
3. Possible to extend with explanations or learning games

### Interactive Web interface

1. It would be accessible online, interactively, e.g., using JsCoq

### Coq Platform Docs

1. Each feature and plugin of Coq and its platform would have one or several interactive tutorials and how-to guides
2. It would be indexed on the Coq Platform to be easy to use
3. Possible to extend with explanations or learning games

### Interactive Web interface

1. It would be accessible online, interactively, e.g., using JsCoq
2. It should be part of Rocq's new website

# Advantages of such a project

## Advantages for users

1. Coq is an interactive and practical software, it <u>needs</u> an interactive and practical documentation

# Advantages of such a project

## Advantages for users

1. Coq is an interactive and practical software, it <u>needs</u> an interactive and practical documentation

2. All the documentation should be <u>accessible at the same place</u>

# Advantages of such a project

## Advantages for users

1. Coq is an interactive and practical software, it <u>needs</u> an interactive and practical documentation

2. All the documentation should be <u>accessible at the same place</u>

3. It should be horizontal making it <u>easier to discover</u> new features,

# Advantages of such a project

## Advantages for users

1. Coq is an interactive and practical software, it <u>needs</u> an interactive and practical documentation

2. All the documentation should be <u>accessible at the same place</u>

3. It should be horizontal making it <u>easier to discover</u> new features,

4. It should be horizontal allowing <u>differentiated learning</u>:
   ↪ Ltac2 for new users  v.s.  Ltac2 for Ltac1's users

## Advantages of such a project

### Advantages for users

1. Coq is an interactive and practical software, it <u>needs</u> an interactive and practical documentation
2. All the documentation should be <u>accessible at the same place</u>
3. It should be horizontal making it <u>easier to discover</u> new features,
4. It should be horizontal allowing <u>differentiated learning</u>:
   ↪ Ltac2 for new users  v.s.  Ltac2 for Ltac1's users

### Advantages for the community

1. It is possible to build <u>modularly and collaboratively</u>

# Advantages of such a project

## Advantages for users

1. Coq is an interactive and practical software, it <u>needs</u> an interactive and practical documentation
2. All the documentation should be <u>accessible at the same place</u>
3. It should be horizontal making it <u>easier to discover</u> new features,
4. It should be horizontal allowing <u>differentiated learning</u>:
   ↪ Ltac2 for new users  v.s.  Ltac2 for Ltac1's users

## Advantages for the community

1. It is possible to build <u>modularly and collaboratively</u>
2. It enables to <u>centralise efforts</u> in a <u>lasting way</u>

# Advantages of such a project

## Advantages for users

1. Coq is an interactive and practical software, it <u>needs</u> an interactive and practical documentation
2. All the documentation should be <u>accessible at the same place</u>
3. It should be horizontal making it <u>easier to discover</u> new features,
4. It should be horizontal allowing <u>differentiated learning</u>:
    ↪ Ltac2 for new users  v.s.  Ltac2 for Ltac1's users

## Advantages for the community

1. It is possible to build <u>modularly and collaboratively</u>
2. It enables to <u>centralise efforts</u> in a <u>lasting way</u>
3. Being horizontal, it should be <u>easier to maintain</u>

## Example: Documenting Equations

### The documentation was inadequate

1. Many features: basics, dependent matching, wf recursion, ...
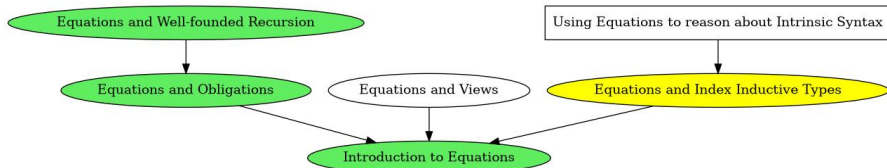2. Despite many examples, the doc was not very fitted nor complete

## Example: Documenting Equations

### The documentation was inadequate

1. Many features: basics, dependent matching, wf recursion, ...
2. Despite many examples, the doc was not very fitted nor complete

### A new documentation

1. We wrote 4 tutorials out of 6 planned, 600l code/ 1200l text
2. It is now much clear, and easy to improve or extend
3. We wrote it modularly with help from the community

# Example: Documenting Equations

## The documentation was inadequate

1. Many features: basics, dependent matching, wf recursion, ...
2. Despite many examples, the doc was not very fitted nor complete

## A new documentation

1. We wrote 4 tutorials out of 6 planned, 600l code/ 1200l text
2. It is now much clear, and easy to improve or extend
3. We wrote it modularly with help from the community

## Many Improvements

1. We found 2/3 bugs writing the documentation
2. We improved the behavior of the main tactic

# Coq Platform Docs

## Demonstration

Search for **Coq Platform Docs**, or go on:
https://coq.inria.fr/platform-docs/

# Coq Platform Docs

**Contributing**

**Everyone can help**

1. <u>Everyone can contribute</u> whatever their expertise !

## Everyone can help

1. <u>Everyone can contribute</u> whatever their expertise !
2. <u>Both</u> experts and casual users' opinions are valuable !

# Contributing

## Everyone can help

1. <u>Everyone can contribute</u> whatever their expertise !
2. <u>Both</u> experts and casual users' opinions are valuable !

## Contributing

There are <u>many ways</u> to contribute:

1. Give feedback on existing tutos / how-tos, or wishes on zulip

# Contributing

## Everyone can help

1. <u>Everyone can contribute</u> whatever their expertise !
2. <u>Both</u> experts and casual users' opinions are valuable !

## Contributing

There are <u>many ways</u> to contribute:

1. Give feedback on existing tutos / how-tos, or wishes on zulip
2. Answer questions, and share knowledge and folklore on zulip

## Everyone can help

1. <u>Everyone can contribute</u> whatever their expertise !
2. <u>Both</u> experts and casual users' opinions are valuable !

## Contributing

There are <u>many ways</u> to contribute:

1. Give feedback on existing tutos / how-tos, or wishes on zulip
2. Answer questions, and share knowledge and folklore on zulip
3. Write tutorials and how-to guides, or write PR to improve them

## Everyone can help

1. <u>Everyone can contribute</u> whatever their expertise !
2. <u>Both</u> experts and casual users' opinions are valuable !

## Contributing

There are <u>many ways</u> to contribute:

1. Give feedback on existing tutos / how-tos, or wishes on zulip
2. Answer questions, and share knowledge and folklore on zulip
3. Write tutorials and how-to guides, or write PR to improve them
4. Help reviewing tutorials and how-to guides

# Contributing

## Everyone can help

1. <u>Everyone can contribute</u> whatever their expertise !
2. <u>Both</u> experts and casual users' opinions are valuable !

## Contributing

There are <u>many ways</u> to contribute:

1. Give feedback on existing tutos / how-tos, or wishes on zulip
2. Answer questions, and share knowledge and folklore on zulip
3. Write tutorials and how-to guides, or write PR to improve them
4. Help reviewing tutorials and how-to guides
5. Help setting up the repo, or the web-interface

## Develop the documentation modularly

For each projet or plugins, we would like to set up:

1. A reviewer team of dev, maintainer, and advanced users
2. A project, and open issues so that people can contribute

## Develop the documentation modularly

For each projet or plugins, we would like to set up:

1. A reviewer team of dev, maintainer, and advanced users
2. A project, and open issues so that people can contribute

## Being a reviewer

The job of the reviewers is not to write documentation, it is to:

## Develop the documentation modularly

For each projet or plugins, we would like to set up:

1. A reviewer team of dev, maintainer, and advanced users
2. A project, and open issues so that people can contribute

## Being a reviewer

The job of the reviewers is not to write documentation, it is to:

1. Identify the needs of documentation, and create issues for it
2. Ensure that the documentation stays consistent through PR
3. Review contributions when some are created

# Coq Platform Docs

## Conclusion

# Sum Up

## The Project

1. Each feature and plugin of Coq and its platform would have one or several <u>interactive tutorials and how-to guides</u>
2. Be accessible <u>interactively online</u>, and part of <u>Rocq's new website</u>

## Contributing

1. Everyone can contribute: you don't need to be a Coq developer
2. Everyone can have a valuable contribution

## Ressources to check out

- Check out the prototype website and the first tutorials
- Come talk to us and give feedback on the zulip stream
- Check out the repo and the CEP

# Next Steps

## Documentation Side

- Get the project known to more people (CoqWorkshop, ...)
- Set up projects and issues, so that people can contribute
- Progress on documenting the main features (Ltac2, ...)
- Get contributors

# Next Steps

## Documentation Side

- Get the project known to more people (CoqWorkshop, ...)
- Set up projects and issues, so that people can contribute
- Progress on documenting the main features (Ltac2, ...)
- Get contributors

## Technical Side

- Set up Alectryon to support reST
- Set up a versioning index on the Coq Platform (Sphinx ?)
- Set up a nice build process to get a dev branch
- Get a better interactive interface (JsCoq2 ?)