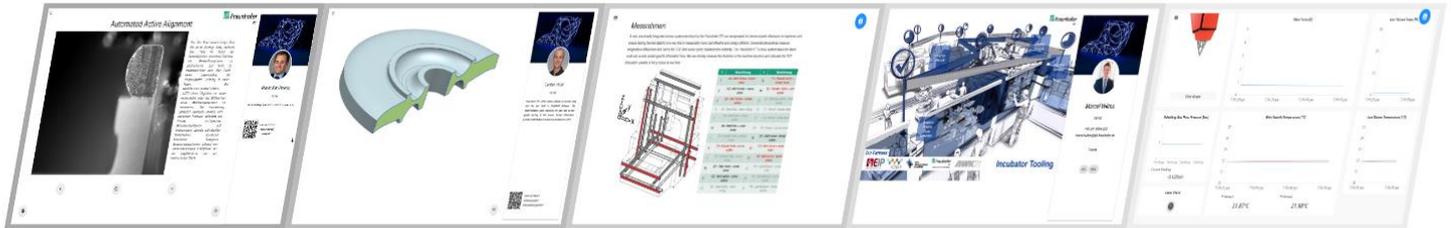


IPT/WZL Dashboards

Documentation, Version 2020.06.29



This is the printed one-document documentation for the dashboards of Fraunhofer IPT and RWTH WZL.

You find this documentation online under <https://ipt-210.pages.fraunhofer.de/dashboards/dashboarddocumentation> - also links in this document may link to there.

Welcome to DashboardDocumentation

Welcome to our documentation regarding interactive **Dashboards** for demonstrators, machines and test benches. The following pages will provide a short overview regarding creation and maintenance of your dashboard. Feel free to use the search function for quick results and frequently asked questions.

If you are creating a dashboard, please inform us about that, such that we can add you to dashboards mailinglist

Getting started - Quickguide

Simple Dashboard - Quickstart

- Navigate to <http://generator.dashboards.vfk.ipt.fraunhofer.de>
- Enter a suitable name for your dashboard
- Select your preferred Gitlab instance (RWTH or Fraunhofer Gitlab)
- Push the "Generate"-Button
- After some minutes, you find your dashboard with the url <https://dashboardname.dashboards.vfk.ipt.fraunhofer.de>
- Go to your dashboard and use the "Edit" menu, to change contents

Advanced Dashboards - Create a GitLab-Account

Ensure that you already have access to the Fraunhofer GitLab or RWTH Aachen Gitlab (WZL users)

Advanced Dashboards - Cloning and your first Commit

Create your Dashboard the same way as a simple Dashboard.

Your new dashboard gets automatically pushed to your Gitlab instance. Clone it by clicking the "Clone"-button and copy the link to your preferred Git tool.

Please refer to the [Getting Started](#) section

Further information

- You find some guidelines, Dos and Don'ts in the [Design Guidelines](#)
- Additional Information about going from Simple to Advanced Dashboards can be found in the [Expanding section](#)
- Detailed information for the usage of IDEs like [Visual Studio 2017/2019](#) and [Visual Studio Code](#)
- For integration of data and database systems, there is an overview in the [Data integration Section](#)

Troubleshooting

Please refer to the search function and the [FAQs](#).

If your question is not answered yet, feel free to contact us directly.

Getting Started

In this section you will learn how to generate a new dashboard from scratch and how to apply changes and customize it.

Create a new Dashboard

Navigate to <https://generator.dashboards.vfk.ipt.fraunhofer.de> and generate a new dashboard by entering a suitable name and selecting "Fraunhofer" or "RWTH Aachen" as your preferred Gitlab instance.

The generation process takes a few seconds. Your dashboard will be **online after around two minutes**.

Generate new dashboards

The DashboardGenerator will create your new dashboard and directly host it in the Virtual Fort Knox Cloud.

Choose a suitable name

MyFancyDashboard
Tutorial

Select GitLab-Instance

- RWTH Aachen
- Fraunhofer

Start Generation

The generation process will take up to two minutes until your dashboard is hosted and can be accessed by you.

CREATE DASHBOARD

Current Step: **Finished**

Dashboard: Tutorial

Add Content

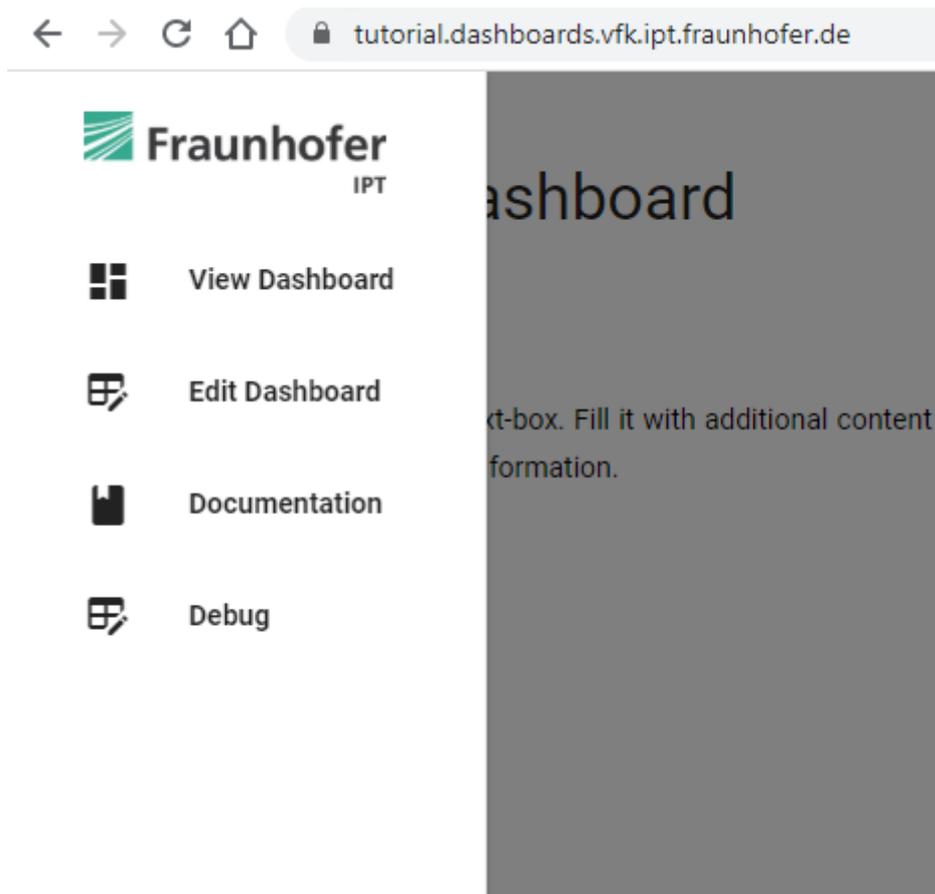
Open your dashboard by navigating to the following URL with your desired browser:

`https://<your-dashboard>.dashboards.vfk.ipt.fraunhofer.de`

E.g. SimpleDashboard is available at

`https://simple.dashboards.vfk.ipt.fraunhofer.de`

Move your mouse to the left side of the screen and a menu appears. Select `Edit` and the editor appears.



After adding all changes in the editor your SimpleDashboard is ready by pushing the "Submit" - button.

Editor problems ?

- Make sure you are working online. The local assets & archive folder can't be used online
- Delete your cache

Please refer to the following sections if you want to expand it and create a so-called "AdvancedDashboard". See [here](#).

Read FAQ

Please refer to the [FAQ](#) before [contacting us](#).

Expanding (AdvancedDashboard)

This page deals with a workflow for expanding an existing SimpleDashboard and preparing it for LiveData - Integration.

Please keep in mind, that also **AdvancedDashboards** contain all elements of a **SimpleDashboard**. This is mandatory for fallback reasons if your dashboard is faced by problems with:

- mobile devices / touch
- uncommon screen resolutions
- network issues
- infrastructure

Create a GitLab-Account

Ensure that you already have access to Fraunhofer GitLab **or** RWTH Aachen Gitlab (WZL users)

- Fraunhofer Gitlab: <https://gitlab.cc-asp.fraunhofer.de/>
- RWTH Aachen Gitlab: <https://git-ce.rwth-aachen.de/>

Please do **not** use the Educational version of RWTH Gitlab <https://git.rwth-aachen.de/>. This is strictly forbidden. See [here](#) for explanation.

Fraunhofer Gitlab

You will require your Fraunhofer ID for the following steps. If you have not created a Gitlab-Account / have not logged in yet, perform the following steps.

Change primary password and get Fraunhofer ID

1. Navigate to the IPT-Portal

2. Click on "Fraunhofer Verzeichnisdienst"
3. Smartcard-Login
4. Click on "Change primary password", if not done before
5. Write down your Fraunhofer ID

Login to Gitlab (IPT)

1. Navigate to Fraunhofer Gitlab: <https://gitlab.cc-asp.fraunhofer.de/>
2. Login by using the Tab "Fraunhofer Verzeichnis" and enter your Credentials.
3. Contact stephan.behm@ipt.fraunhofer.de or malte.janas@ipt.fraunhofer.de and ask for access to the dashboard group

RWTH Gitlab / WZL

Please only use the CE (Community Edition) instance of the RWTH Gitlab. It can be found [here](#). The use of `git.rwth-aachen.de` is only allowed for educational projects.

Login to Gitlab (RWTH)

Accessing the Web-UI of Gitlab can be easily performed by visiting: <https://git-ce.rwth-aachen.de>.

GitLab of the RWTH Aachen University



Willkommen beim GitLab der RWTH Aachen!

Angehörige der RWTH Aachen (Studierende und Mitarbeitende) nutzen bitte ausschließlich den **DFN AAI Single Sign-On** und suchen im folgenden Fenster die RWTH Aachen University aus. Erst dann erhalten Nutzer die Möglichkeit, Projekte und Gruppen anzulegen. Externe Nutzer können sich über **GitHub** authentifizieren, können aber keine Gruppen und Projekte anlegen.

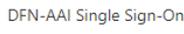
Ab sofort gelten folgende Nutzungsbedingungen:
[Nutzungsbedingungen](#)

[https://gitlab.rwth-aachen.de](#)

Sign in

Sign in with

 GitHub

 DFN-AAI Single Sign-On

Remember me

Click on the Button with `DFN-AAI Single Sign-On`

DFN-AAI

DFN
DEUTSCHES FORSCHUNGSNETZ

[DFN-AAI](#) | [Impressum](#) | [Datenschutz](#)

Organisation auswählen

Um auf den Dienst **GitLab Community Edition** zuzugreifen, wählen oder suchen Sie bitte die Organisation, der Sie angehören.

RWTH Aachen University

Auswählen

- Auswahl für die laufende Webbrowser Sitzung speichern.
- Auswahl permanent speichern und diesen Schritt von jetzt an überspringen.

Betrieben mit [SWITCHwayf](#).

Proceed by selectiong `RWTH Aachen University` .

Get Access to the Dashboard group

Unfortunately, there is no invitation link functionality in Gitlab - therefore we have to provide access for each user separately.

Please request access by writing a mail to one of the following persons:

- stephan.behm@ipt.fraunhofer.de
- malte.janas@ipt.fraunhofer.de
- arno.schmetz@ipt.fraunhofer.de

Start developing

Continue by selecting your IDE:

- [Visual Studio 2017/2019](#) (Fraunhofer IPT)

- [Visual Studio Code](#) (WZL/RWTH Aachen)

Remark: The folders assets and archive will be overridden in the online version.
The content of the contact card has to be set via the edit mode or hardcoded in the dashboard code itself!

Design Guideline

One of the most important aspects of the dashboard project is, to maintain a common appearance of all dashboards. Therefore we have defined a Design Guideline to help you designing your dashboard.

Basic elements

A dashboard itself consists of two main elements. The first one is the content area, which is fully customizable by you. The second element is the so-called ContactCard - a predefined element which can be easily filled with your contact and project data.

Always keep in mind: you only have a very limited timeframe to attract users. Be respectful with their time. Users should be able to determine the main topic of your dashboard within 3 seconds and decide whether that is the right topic for their interest.

RWTH and Fraunhofer Guidelines

Based on the feedbacks and talks with the WZL and IPT PR departments, we collect additional guidelines here. Keep them in the loop for your work in best case.

- Always use sans-serif fonts for charts and text
- To not alter/change the Logos of RWTH/Fraunhofer without explicit permission
- Check your wording and nomenclature with your local department/PR
- Text block always have to be left-aligned (old dashboards sometimes had justify)
- All what you present via dashboards is by definition **world public**. Make sure, not to show private information!

- Before communicating a dashboard to the public, get a feedback from your PR department
- This also holds for fairs and conferences

SimpleDashboard

There are different display configuration for the order of text boxes and images in the standard poster format.

Mobile display will be vertically arranged:

- Image 1
- Textbox 1
- Image 2
- Textbox 2

Full screen display order:

- Top-Left: Image 1
- Bottom-Left: Textbox 2
- Top-Right: Textbox 1
- Bottom-Right: Image 2

ContactCard

The contact card is a recurring element, such that all users know about where to get in touch for detailed information. Also you can include your team members or (short) description texts. The QR code is a vCard allowing the users to store information about the experts fast.

For Simpledashboards, set the contents of the contact card via the edit mode online, as the local json file for debugging gets overridden!

Dos and Don'ts

Do

- Be respectful with the time of the users
- Allow fast determination of key benefit
- Use images, animations and very brief text elements
- Allow interactions for users to "touch" your project directly
- In case of "live" data, prepare a replay for the time, the machine is offline or not doing interesting stuff
- Ideally: Tell a story with your dashboard
- Think about different screen sizes (smartphone, tablet, Desktop)

Don't

- Create Wall of Texts - most users will never read it
- Live-only view - If your machine stands still, your dashboard shows no value
- Information-bombardement - If you try to show all information at once, users have no chance to find their way into your topic
- Huge 3D models - If you use 3D models, try to reduce them with respect to the users devices having to receive and render it (also, extensive rendering drains battery fast)
- Browser-specific features - You do not know beside your own device(s), which setup users have. If you rely on specific features (e.g. only chrome) you loose users immediately

Helpful frontend libraries

- Material Design Lite: Standard library used for basic design elements.
<https://getmdl.io>
- Chart.js: A library for (animated) graphs and charts <https://www.chartjs.org/>
- three.js: A library for displaying and rendering 2D/3D objects (e.g. STL-files from your products) <https://threejs.org/>
- D3.js: A library for Plots and Data Driven Documents <https://d3js.org/>

Data integration

Available Databases

We Provide two managed and hosted databases in separate VMs to handle a high amount of traffic. If you need another database, please [contact us](#)

These databases are no systems for production or high volume data. Please use them only for the data necessary for your dashboards.

The database management systems described here, are only reachable from the Virtual Fort Knox or Fraunhofer IPT network. If you need access to the VFK network, we can create a VPN access for you (For WZL: you will also need the Aruba client from your IT Department)

MongoDB (No-SQL, preferred)

Path: `ipt-210-db-mongo1.cpi002.svc.fortknox.local`

Please keep in mind that you have to be connected with the IPT network (Physically or via VPN) to access this database. Deployed dashboard will automatically have access.

Example

See here: <https://docs.mongodb.com/ecosystem/drivers/csharp/>

What is MongoDB?

MongoDB is a No-Sql-Database, which means that it is not organized like traditional SQL databases in organized tables, but in so-called documents which are described by BSON (JSON-like binary encoded format for better performance). There is no predefined structure and no complex database schemas have to be created before starting to collect data.

A typical MongoDB instance consists of the following types of elements:

- Databases: Typical each dashboard has a single database for itself. It gets automatically generated during the creation process by the DashboardGenerator. It should at least contain a single collection.
- Collections: A database can have multiple collections, each with a different stored type of document. The objects in a collection are called documents and have a similar structure to each other. MongoDB automatically creates an index if none is defined by the developer. It is typically denoted as `_id`.
- Documents: A document contains the JSON-like encoded data structure filled with one or multiple data fields.
- Fields: Each field contains a single data value.

See [here](#) for official docs.

Tags: database, data storage, mongodb

MongoDB Compass

Compass is a lightweight tool to explore your existing collections.

See [here](#) for downloads and documentation.

Please make sure to use the Community Edition, which is free to use, while the standard version comes at costs!

MariaDB (SQL)

Path: `ipt-210-db-sql1.cpi002.svc.fortknox.local`

Please keep in mind that you have to be connected with the IPT network (Physically or via VPN) to access this database. Deployed dashboard will automatically have access.

See [here](#) for official docs.

Example

See here: <https://devandy.de/c-und-mysql-mariadb/>

InfluxDb (Time series)

Path: `ipt-210-db-influx1.cpi002.svc.fortknox.local`

Please keep in mind that you have to be connected with the IPT network (Physically or via VPN) to access this database. Deployed dashboard will automatically have access.

See [here](#) for official docs.

Example

See here: <https://github.com/influxdata/influxdb-csharp>

Tips and Recommendations

- Avoid complexity: Keep your data simple, such that you do not need extensive processing of the data in the databases
- Reduce volume: Keep the volume tight to what generates benefit for the users and store only relevant data
- Replay first: Make sure, that at every time, users can see something interesting when interacting with your data. You can use replays of real data processes for example

Frequently asked questions

We will keep this page up-to-date with frequently upcoming questions. If you have a problem or issue with your dashboard, please use the [issue function of the simple dashboard](#) to report errors.

What is the minimal version of a dashboard?

The minimal version is the auto-generated SimpleDashboard, which can be generated here: [DashboardGenerator](#)

What should the minimal version of my desktop contain?

The minimal version should contain a minimal selection of static content of your demonstrator.

- ContactCard
- 2x Textboxes
- 2x Images

The user should become a basic understanding of your project by looking at it for a short period of time.

Who updates my dashboard?

If you do not touch the code of your SimpleDashboard, we will take care of everything. But if you do so, your dashboard will automatically become an AdvancedDashboard.

What is an AdvancedDashboard?

All dashboards with modifications exceeding the integrated editor are called AdvancedDashboards. You should be familiar with Git and Gitlab to proceed.

They can contain multiple pages, visualizations and data sources and have to be updated manually.

Which knowledge is required for creating a dashboard for my project?

You don't need any advanced knowledge for the creation of a dashboard. The minimal version can be generated by using our provided [DashboardGenerator](#) and by filling the requested fields.

If you want to show interactive elements, you should have basic knowledge about the following topics:

- HTML
- CSS (media queries, classes)
- Javascript (jQuery, HTTP-Requests)
- C# (optional)

If you want to work with data you should have a basic understanding of object-oriented programming.

Do I need a specific software? Can you recommend one?

No, you don't need any special software for creating a dashboard. The generation process is triggered via a Web-Application and all code changes can be performed in the Gitlab WebIDE or via the integrated Markdown-Editor in the Generator.

What is Gitlab?

GitLab is a web-based software that allows the management of Software projects and source code. It comes with the local installation of git for version control.

About the editing

Visual Studio 2017/2019

Visual Studio is an [Integrated Development Environment](#) for developing and debugging various kinds of software projects. We use Visual Studio 2017 for the development of our WebApps. On the Getting Started page we offer a quick tutorial on how to start developing.

Please keep in mind, that Visual Studio 2015 is not suitable for developing Apps for/with Dotnet Core/ASP.Net Core. In this case you should consider upgrading to a newer version or to use Visual Studio Code.

Furthermore we want you to understand, that the usage of the Community edition for developing is not permitted. Please contact your IT department for the Profession Editions, if they are not yet installed on your device

Visual Studio Code

Visual Studio Code is a free Text editor developed by Microsoft. It can be enhanced by a tremendous plugin structure. See the Getting Started page for a tutorial on how to use it for developing you dashboard.

HTML-Editor

Your online SimpleDashboards can be edited using the Web-Editor. Just put in the content you want to. You can also add HTML-based strings to add additional formatting/functions.

Which language should I use?

All dashboards for AWK should use English as their primary language.

My Umlauts in a .json file / ContactCard are displayed by cryptic icons - what should I do?

Save the whole document in UTF-8 encoding by using Visual Studio Code. This error most likely occurs by saving *.json - files in Visual Studio 2017, which uses a different format.

What resolutions should my images and design elements have?

The following resolutions are generally minimum values - consider using this or a higher resolution.

Where can I create a QR-Code and which data should it contain?

The QR-Code in the ContactCard is for providing a fast and easy way to save the contact data of a project. We use <https://www.qrcode-generator.de/> to create QR-Codes for each dashboard. The data is a vCard including your name, the institution name and your mail/phone.

Which audience do I have?

Your audience has limited time and may lack detailed knowledge of your research field. Handle their time in a respectful way and don't waste it by providing overloaded dashboards. Take care, that your dashboard may needs a human assistant to help understanding complex topics.

How do I keep my dashboard up-to-date

Most basic maintenance stuff is performed by us. You don't have to care about hosting process, as long as you committed your changes to the master-Branch and the pipeline finishes successfully.

Which database should I use? Can you recommend one?

We provide a fully-managed MongoDB instance in the Fraunhofer Cloud, also known as VFK (Virtual Fortknox). See [here](#) for detailed instructions.

We also provide a MariaDB (SQL) in a separate virtual machine (VM). See [here](#) for detailed instructions.

What is MongoDB?

MongoDB is a No-Sql-Database, which means that it is not organized like traditional SQL databases in organizes tables, but in so-called documents which are described by BSON (JSON-like binary encoded format for better performance). There is no predefined structure and no complex database schemas have to be created before starting to collect data.

A typical MongoDB instance consists of the following types of elements:

- Databases: Typical each dashboard has a single database for itself. It gets automatically generated during the creation process by the DashboardGenerator. It should at least contain a single collection.
- Collections: A database can have multiple collections, each with a different stored type of document. The objects in a collection are called documents and have a similar structure to each other. MongoDB automatically creates an index if none is defined by the developer. It is typically denoted as `_id`.
- Documents: A document contains the JSON-like encoded data structure filled with one or multiple data fields.
- Fields: Each field contains a single data value.

Tags: database, data storage, mongodb

I can't access the provided MongoDB - what should I do?

Please check, that the following conditions are met:

- Access to VFK network via Aruba VPN client
- Correct login data (URL, Port)

Can I use another backend like Angular, React or Node.JS?

Yes and No - we strongly control that the provided pipeline stays in shape, our provided `ipt_dashboards.css` - file is used to ensure a constant user experience and that all dashboards are packed as a docker container. Please contact us, before implementing your own solution.

I have already implemented a fancy dashboard or app by using another framework - what now?

First, you should reflect yourself whether your implementation is a good choice for being displayed on an exhibition. What is your target group? Will they stay long enough on your demonstrator to understand the basic idea and concepts behind?

Usually, you will come to the conclusion that you can reduce the amount of displayed information to a minimum. Therefore a reimplementing as a dashboard is not that much of work and can be done in a fraction of time you have spent in creating your demonstrator itself. If you think, that you don't have to reduce your solution, you should take a look at `iFrames`.

How can I restart my dashboard manually?

You can restart your dashboard by hand, if you perform the following steps:

- Navigate to your Gitlab project
- Go to CI/CD section (left nav menu)
- Open the Pipelines section
- Push the `Run Pipeline` button
- Select `master` branch
- Press `Run Pipeline` button

Your pipeline is triggered and the full build and deploy process starts again. Please note, that all deploy steps are only performed when running a pipeline for the `master` branch. You can change this, by modifying the `.gitlab.ci.yml` file in your project.

Is there a restart policy for my dashboard?

All dashboards will restart automatically after crashing. This is realized by a docker restart flag `--restart always`.

Click [here](#) for more information.

Additionally we will restart all dashboards automatically after 7 days, usually on sundays. You can deactivate this, by deactivating the scheduled `Nightly build and restart` -Job in your corresponding Gitlab project.

I can't clone code / my project from GitLab

There are at least two login methods: Auth via Username (FraunhoferId, e.g. `mm1234567`) or Oauth2 / Authentication via Token

Tags: Authentication, Gitlab, Credentials, Cloning

Method 1: Authentication via Username and Password

You may reset your Git Credential Manager to enter valid login data.

Method 2: Oauth2 / Private Access Token

- Create a new Personal Access Token in Gitlab [See here](#)
- Enter "oauth2" as your username
- Use your generated token as your password

You can clone your project via the following command to your machine:

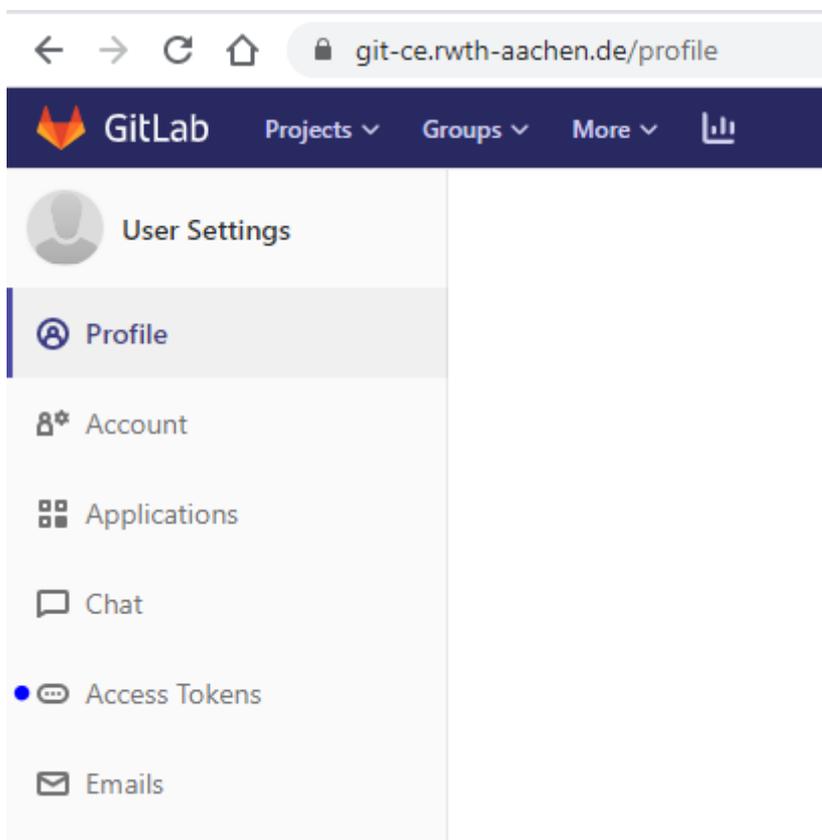
```
RWTH/WZL: $ git clone https://oauth2:<your-token>@git-ce.rwth-aachen.de/<your-project>.git
```

```
Fraunhofer: $ git clone https://oauth2:<your-token>@gitlab.cc-asp.fraunhofer.de/<your-project>.git
```

How can I create a new PAT (Personal Access Token)?

You can find the official Docs [here](#). Otherwise you can use the appended instructions.

1. Select **Access Tokens** in User Settings.



2. Enter a name of your choice and select the `read_repository` and `write_repository` options. Click on `Create personal access token`.

User Settings > Access Tokens

Personal Access Tokens

You can generate a personal access token for each application you use that needs access to the GitLab API.

You can also use personal access tokens to authenticate against Git over HTTP. They are the only accepted password when you have Two-Factor Authentication (2FA) enabled.

Add a personal access token

Pick a name for the application, and we'll give you a unique personal access token.

Name

Expires at

Scopes

- api**
Grants complete read/write access to the API, including all groups and projects, the container registry, and the package registry.
- read_user**
Grants read-only access to the authenticated user's profile through the /user API endpoint, which includes username, public email, and full name. Also grants access to read-only API endpoints under /users.
- read_repository**
Grants read-only access to repositories on private projects using Git-over-HTTP or the Repository Files API.
- write_repository**
Grants read-write access to repositories on private projects using Git-over-HTTP (not using the API).
- read_registry**
Grants read-only access to container registry images on private projects.

[Create personal access token](#)

3. Your created access token gets displayed. Copy it so a safe location. You will not be able to request it again.

User Settings > Access Tokens

🔔 Your new personal access token has been created. ✕

Personal Access Tokens

You can generate a personal access token for each application you use that needs access to the GitLab API.

You can also use personal access tokens to authenticate against Git over HTTP. They are the only accepted password when you have Two-Factor Authentication (2FA) enabled.

Your New Personal Access Token

ix199BkTiihG77woRdBx 📄

Make sure you save it - you won't be able to access it again.

Add a personal access token

Pick a name for the application, and we'll give you a unique personal access token.

Name

4. You can revoke/delete your token by clicking the `Revoke` button.

Active Personal Access Tokens (3)

Name	Created	Expires	Scopes	
Login	Mar 6, 2020	Never	read_repository, write_repository	Revoke

Tags: Token, Clone, Authentication, Login

How can I switch the display order of my SimpleDashboard in mobile view?

The item order of images and textboxes in your SimpleDashboard can be changed, by editing the `~YourDashboard/wwwroot/js/index.js` file.

The order of items is defined by the following code:

```
//re-add them in desired order
/*
 * [ second | cell ][ ContactCard]
 * [ first  | cell ][ ContactCard]
 */

var firstRow = document.getElementById("secondCell");
firstRow.appendChild(img1div);
firstRow.appendChild(text2div);

var secondRow = document.getElementById("firstCell");
secondRow.appendChild(img2div);
secondRow.appendChild(text1div);
```

What are the environments for the online version?

The online version runs in a containerized linux machine, with english locale.

This especially means:

- Filenames are case-sensitive (other than Windows)
- Default serialization of numbers/dates is different (e.g. 1.25 instead of 1,25)
 - Hint: Use `InvariantCulture` in your Code to be independent

I have an advanced dashboard, but my contact card contents are not the same as in the local version

The folders assets and archive are overridden in the online version. To change the contents of the contact card, you need to perform the changes via the edit mode (recommended) or hard code it into the dashboard code (not recommended)

I have no team members, but my contact card shows "Team" with an empty list

Please refresh the page and refresh the cache (in most browsers ctrl+F5 or shift+F5), that should do the trick.

Contact information

Feel free contacting us if you have any problems with your dashboards.

Stephan Behm

stephan.behm@ipt.fraunhofer.de

Malte Janas

malte.janas@ipt.fraunhofer.de

Arno Schmetz

arno.schmetz@ipt.fraunhofer.de

