TAKAHASHI GROUP (情報化学研究室)

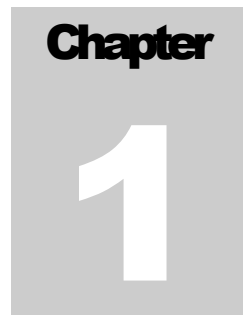## Hokkaido University

CATALYST ACQUISITION BY DATA SCIENCE

CADS

Server Manual

ENGLISH

Table of Contents

Chapter

1

Introduction

Get to know what CADS is and what it can do for you as a researcher.

CADS is a catalyst analysis environment that implements a user-friendly graphic user interface(GUI) that aims to help scientist to analyze data in a more simple and effective way without the need of coding and/or setting up complex IT environments.

CADS Overview

CADS provides three basic functions for users: "Data Management", "Analysis", and "Prediction".

Data Management

CADS provides the ability to upload and record individual data as well as data analysis procedures with the option for multi-user sessions, allowing users to share data and analysis reports with collaborators of their choosing. Meaning that not only can a researcher study and analyze their own data, but they can also compare and study other's shared data for a better and deeper understanding.

Analysis

Data visualization tools (scatter plotting, histograms, etc.) and machine learning functions are available for catalyst data analysis and machine learning model selection. This is the core part of the CADS platform and one of its major strengths. In the workspace area a user can load a various buffet of components through which the user can then analyze and manipulate the data for make new findings and getting further understanding. The range of Components is steadily growing and the power of this section is increasing every month. Besides the classic visualization tools mentioned above, there are many more, ranging from image manipulation, machine learning and advanced analyzing and visualization tools, all with simple and straight forward user interfaces that guides the user to do complex analyzing with few and user-friendly operations.

Prediction

Data prediction capabilities are available within the "Analysis" function and inside the Machine Learning components, allowing users to apply and store learning models for predicting new output (e.g. predicting new catalysts). Under the Prediction tab the user can then apply those pre-trained and saved models in order to predict the result with new data.

CADS Architecture

On the surface to the average user the CADS system is just an ordinary website operated by basic mouse

and keyboard operations. A few clicks and perhaps some simple typing of values and then submit to the CADS server, and voila, the data will be displayed in a new graphical and exciting way.

Under the hood the CADS system is built with Django and React JS and for easy maintenance and sound and familiar structure. The custom and generic Analyzing components are using state of the art technology from various high-end libraries for graphical display of data, e.g. Bokeh JS and Plotly JS. On the server side a wide range of support system is feeding the components with all sorts of data management capabilities like SciKit Learn and SciKit Image and much more. This is not needed to be known for the CADS user, but it might be good to know that under the hood are all the well-known complexity the research world is used to, but through CADS, most of the complexity is removed or toned down. Data analyzing should not just be for IT savvy people, it should be available to everyone.

CADS Online in the Browser

In order to use CADS to its fullest one need to have a user account in order to be able to save data and workspaces. Applying for an account is simple and quick and after confirming with a valid email address the CADS system is fully available to use. All interaction from hereon is based on familiar browser interactions using classic form fields and buttons and no need for any plugins or additional downloads are required. All work can be saved and if wanted to, relatively easily shared.

CADS Server

For more details on CADS in general terms, please look for the CADS user Manual or component manual found on the CADS website. Here we will now dive deep into the world of CADS Server Setup and Maintenance for those of you who wish to run your own CADS system.
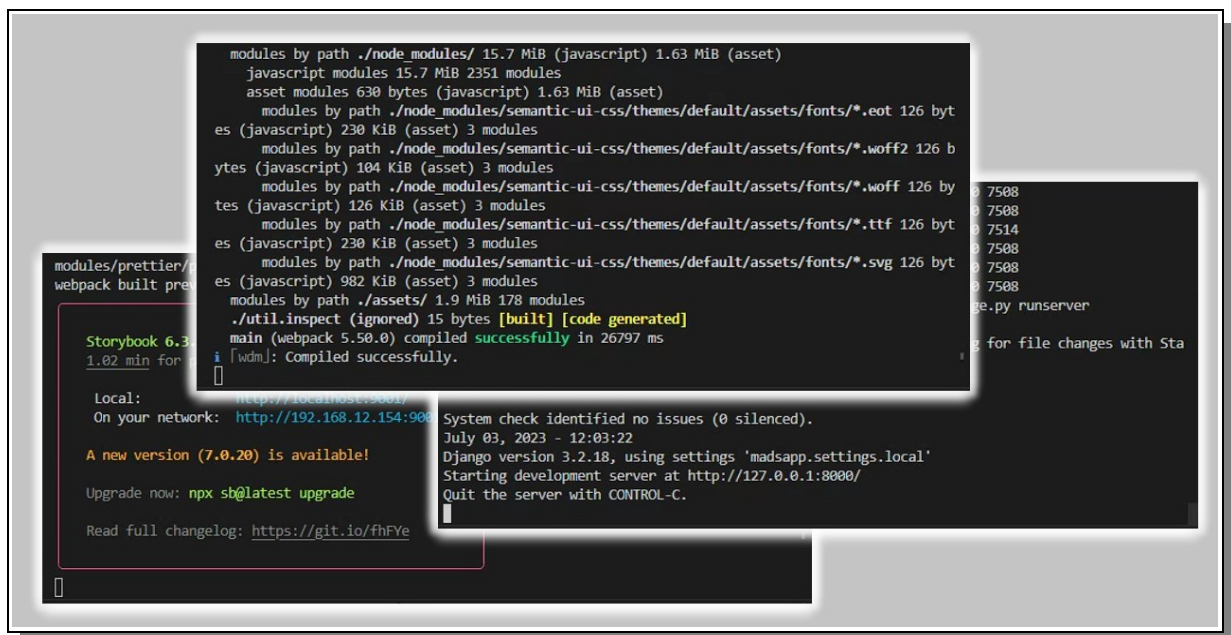
Server Manual

So you have tried the CADS website online, and have the idea that you want to run and manage your own, well then you are in luck, because here is were you learn how to

I n this chapter We will try to explain everything that you need to know about setting up your own CADS Server and also how to use Django Admin to manage the website and its users and data. We will not explain everything in detail in this document alone though, but sometimes just point in the right direction for where the needed information may be found. e.g. a lot of the information needed is already on GitHub, and when that is the case we just basically share the necessary link.
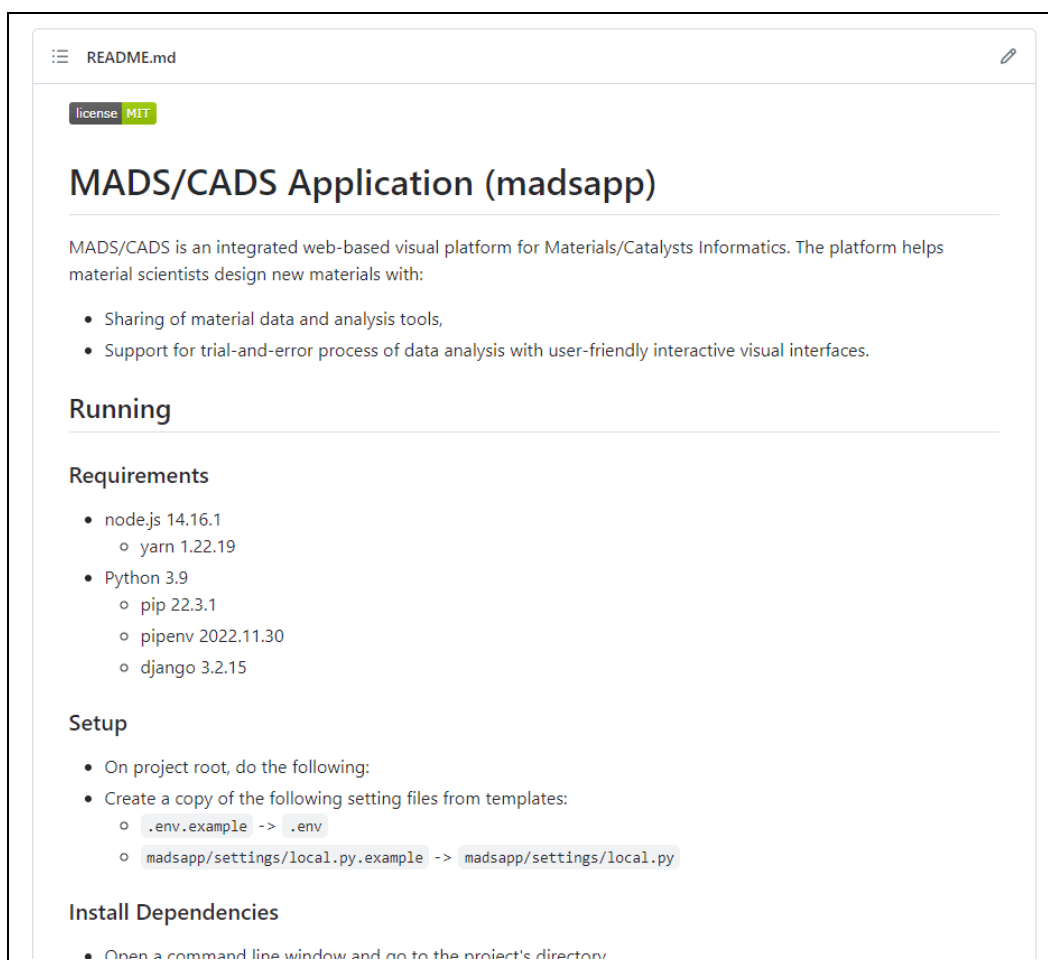

Figure 1: The GitHub ReadMe will be your friend when setting up your local server.

CADS Local Server

The first thing we will do is to setup and run a local server on you local machine, and get that fully up and running. To do that you first need to make sure you have all the prerequisites in order, and when that is all done, starting to install CADS including all its required libraries and server software. The last steps will be to configure the server so that it runs as intended and then comes the final testing.

CADS Server Initial Requirements

CADS server runs using Django (Python server framework) and related support library. Therefore the machine that will host your local CADS server needs to be a Linux based machine (Windows cannot run all the necessary frameworks and support libraries, so that will not work). We mainly recommend Ubuntu, e.g 20.04, but any Linux system should work, but if it it doesn't we will not have any additional suggestions on how to solve it, beyond the content in this manual, except start browsing online for help.

So if you at this point have your Ubuntu machine up and running and ready to go then keep reading.

Download CADS Code

CADS code and a full description on what to install and configure before it will run on your local machine is all available on GitHub via this link: https://github.com/Material-MADS/mads-app

You do not need a GitHub account to visit this page and download the code.

On this page you will find all the CADS code as well as a ReadMe that explains the setup and configuration of the system. Just scroll down below the code files to find the ReadMe as can be seen in Figure 1 on the previous page.

Let's start with downloading the CADS code to your local Linux machine.

If you are not familiar how to do that and/or you are new to GitHub in general, then please begin by reading the GitHub Help documentation found on the GitHub main site.

After you have downloaded all the code into a folder on your local machine, navigate to that folder and begin the next step in getting the CADS server running.

CADS Server Prerequisites

The first thing you need to do is to install all the backbone systems and software that is needed for the CADS server to run. In the ReadMe on GitHub it states which software is needed and which versions. The versions are important for compatibility so do your best to follow that.

At the time of writing this, GitHub mentions the following applications; node.js (14.16.1), yarn (1.22.19), Python (3.9), pip (22.3.1), pipenv (2022.11.30) and django (3.2.15).

The final say is always in the GitHub ReadMe, so if anything written here is different than what is written on GitHub, GitHub is always the correct one to follow.

After all those are installed you can move on in the documentation.

CADS Setup Files

Follow the Setup instructions and create the setup files required. No need to edit them at this moment, just make sure they exist as instructed.

Install Dependencies

CADS has a lot of required libraries in order to run, but that is all managed internally by the Pipfile and the Pipfile.lock files that contains the information the system needs to install all dependencies.

Just follow the GitHub ReadMe instructions on how to install those dependencies.

This is of course done via the terminal, like always when working with Ubuntu and Linux, so if you are not familiar of working with the terminal we recommend that you learn a bit about it somewhere online.

If you run into some kind of problem, follow the advice on the GitHub page in order to trying to rectify the issue, and if that does not help, try Googling the issue or ask ChatGPT about the error message you get and see if that will help you solve the issues.

But basically all should be good and all the dependencies should install all fine and we can move to the next step.

Running CADS

When all is installed as requested we are ready to run the CADS server on your local machine.

We recommend that you open up two terminal window at your CADS project's root folder, one to run the web server and the other to run the website.

Then you follow the instructions in the GitHub's ReadMe to start the web server in one terminal and then how to initialize and run the website in the other. As you can read on GitHub, the first time you start it you need to initialize the database and fill it with base data, but next time you just need to activate the virtual environment and call the '*python manage.py runserver*' command.

Final End Result

If all went well, then you should have been automatically transported to a web browser to the local host link of http://127.0.0.1:8000 and there you would find your own local CADS website running like seen in Figure 2 below.
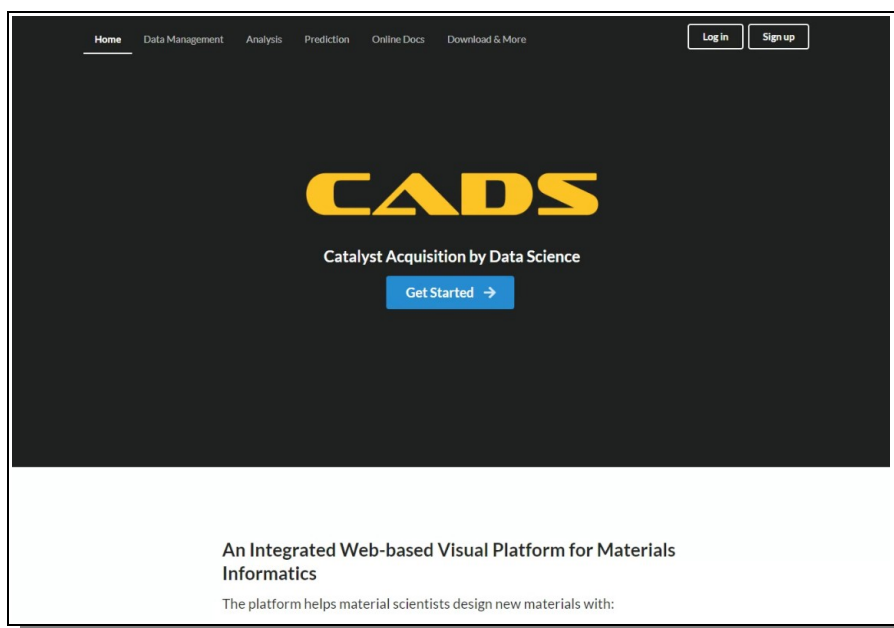

Figure 2: If you did everything correct, then your local CADS website should look like this.

Creating Your Basic User Login Account

The local CADS site does not use real e-mail account details to communicate with the user, the link to access a newly created account, instead a "fake" mail has been generated on the server for the local user to click the link. So after you have filled in the details in the 'Sign Up' form and submitted, an account have been created but no e-mail have been sent. Instead you go back to your server code and find a folder that has been generated called 'tmp_email' and inside of that there will be a file with a long filename containing the date and time of its creation, and at the near of the bottom of that file you will find the link in question, that will activate the login account properly, as can be seen in Figure 3 below
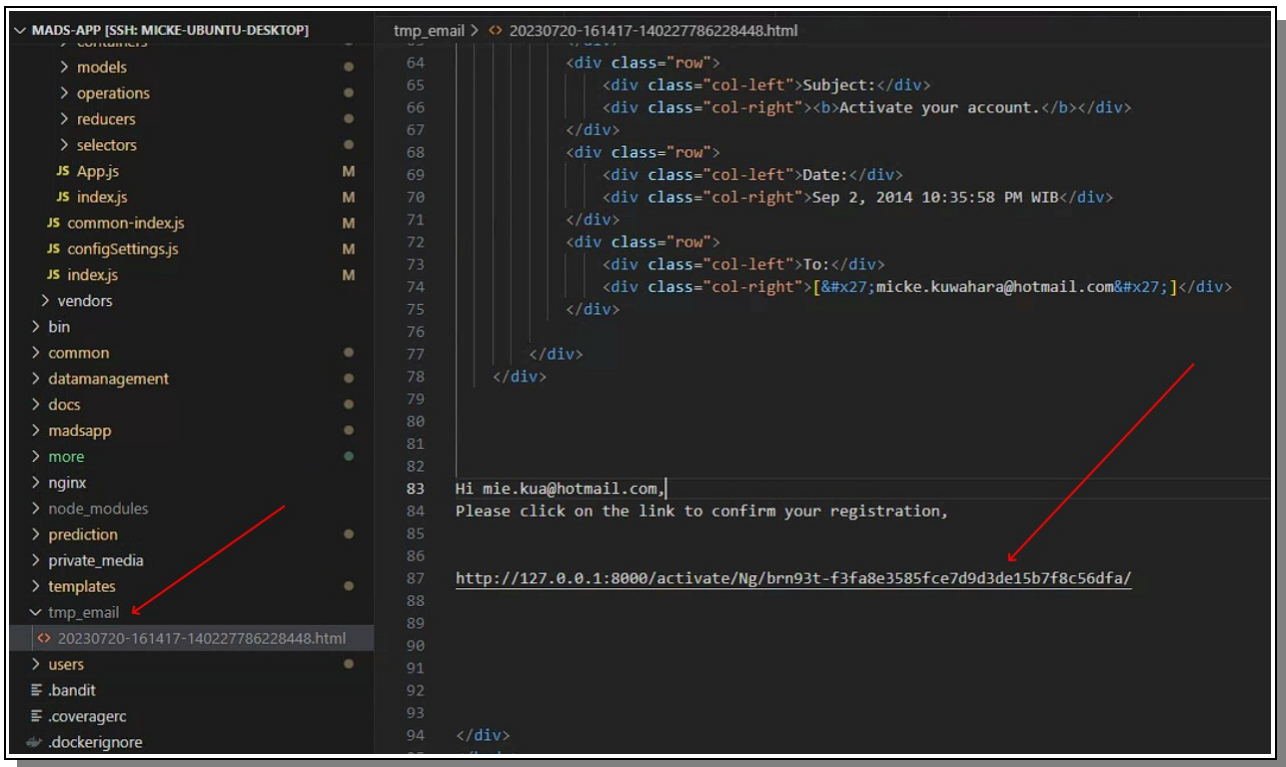
Figure 3: On a local machine any emails are just generated as files in a dedicated folder.

If you test out any other feature where an e-mail is supposed to be sent, then it is in above folder you will always find said "e-mail".

CADS Website Admin Page

With Django server side comes a very handy and useful Administration page that manages various aspects of the site, such as users and data inputs. In order to get access to that admin page you must first create a Super User with admin rights, and that is done via the terminal.

Creating a Superuser Admin Account

Before you run the server with the command 'python manage.py runserver', in the same terminal, you should type:

```
$ python manage.py createsuperuser
```

The system will the ask you for a username, an e-mail (may be blank) and a password. Fill them in and

execute. After that you can start the server as instructed previously.

Opening the Django Admin Page

The link to the admin page is 'http://127.0.0.1:8000/admin/' and when asked for login credentials use the username and the password you typed when creating the superuser previously and if that was done properly the CADS Django Admin page will open and look something similar as in Figure 4 below.
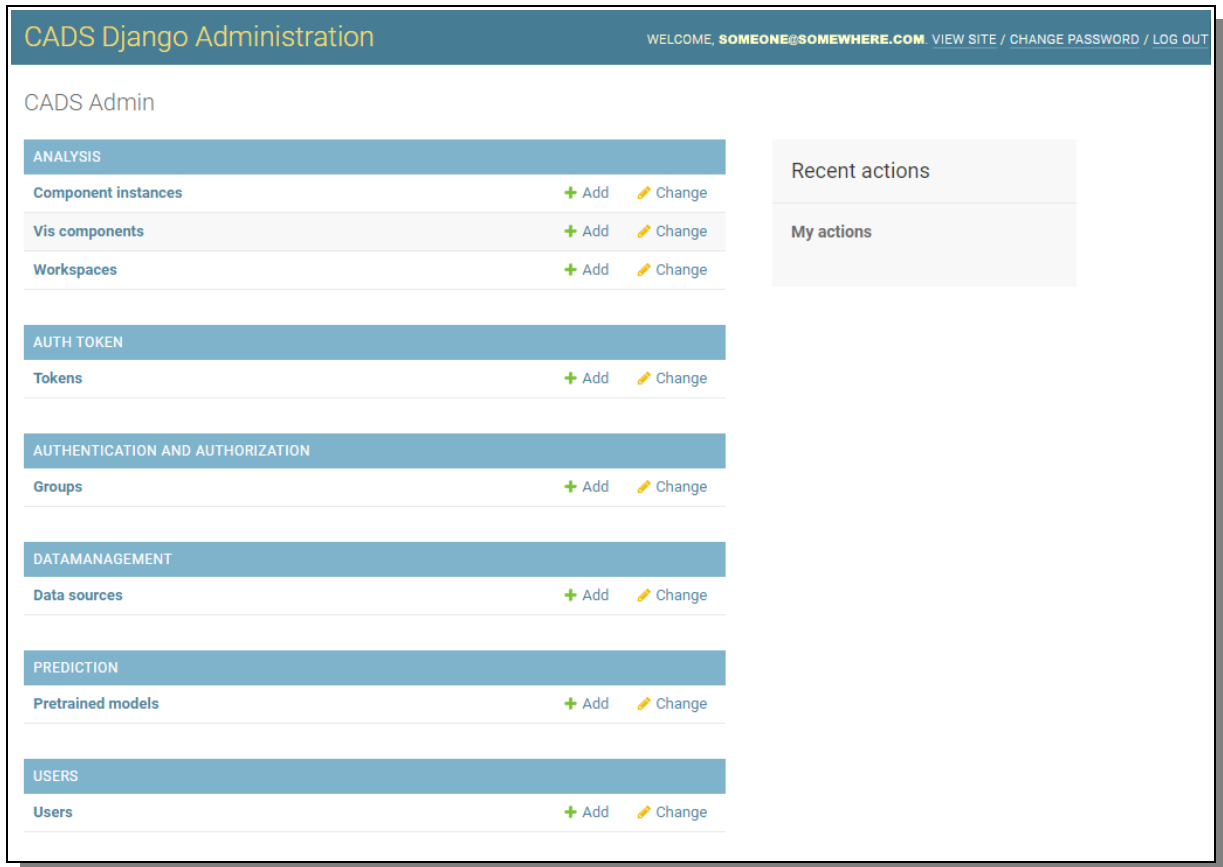

Figure 4: This is the CADS Django Administration page where users and data can be hard-core managed.

Using the Django Admin Page

Inside the Django Administration Page for your CADS site you can view, add, edit and delete various data available on the site, such as e.g. Users, Workspaces and Data Sources.

The Admin page is very much self explanatory and it is also using the Django default design, so if below explanation is not enough, we just humbly point you to the Internet where you can Google for more details on how the Django Admin works.

Component Instances and Vis components, are available but contains no data since the final database structure does not manage those entities via the database, so that section will always be empty and can be ignored.

Workspaces (as described in details in the User Manual) can be managed here though. If the admin clicks on that link then a list of existing workspaces and some basic information will be displayed on the right for further investigation or manipulation.

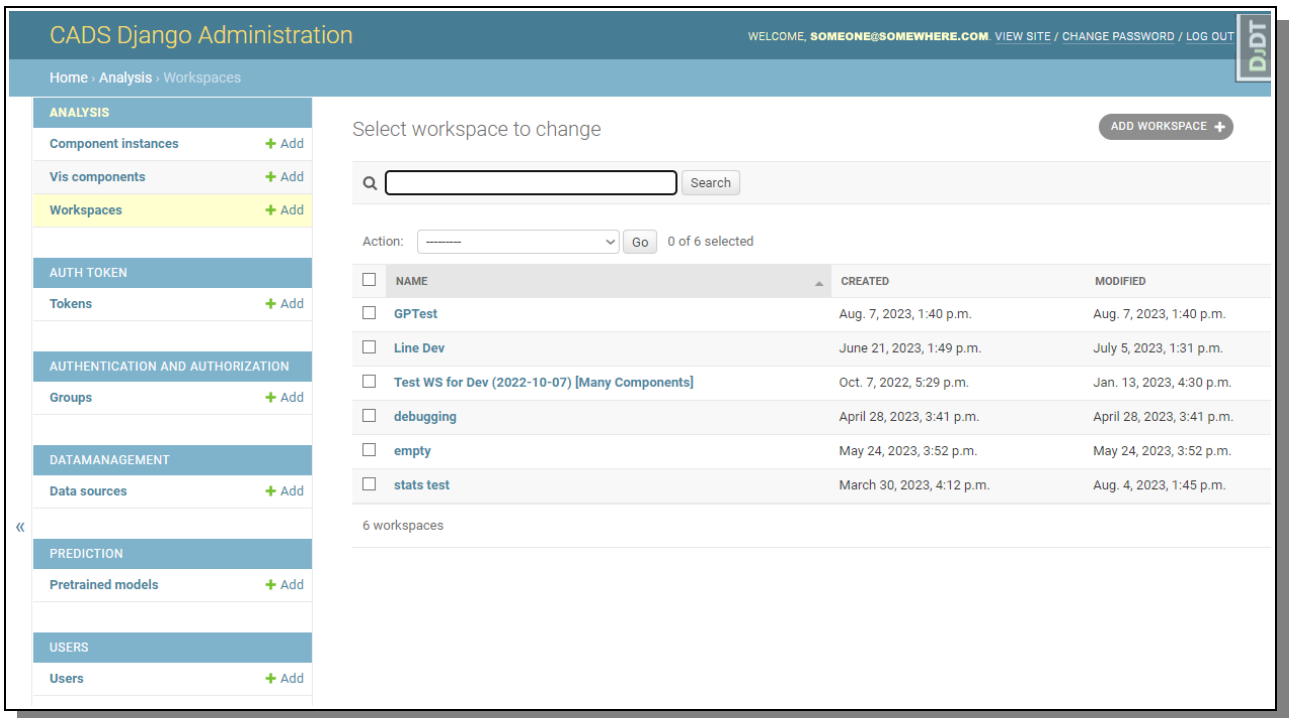In Figure 5 below you can see how an example list of workspaces could look like.

Figure 5: A list of available workspaces as seen in the Django Admin Page.

To view and edit details for a specific workspace, one just click on the list item link to open a new panel where such editing can be done.

In Figure 6 on the next page an example view of the workspace editing admin area can be seen with all its specific and default options and parameters.

As an admin superuser, one can change anything and everything for all and everyone no matter who originally created the data item, in this example a workspace.

Besides changing typical parameters as names and description, one can also change the original owner, the accessibly between private and public.

The admin superuser can also add or remove existing users or groups to have access to the data item if such features is used and needed.

The other areas of the admin page basically looks and works the same, so no need to go into further details for each. But as a quick overview we just describe each area a bit so that you have a better picture of what it means.

- Tokens is a feature that is used to allow the admin staff to generate new users on their own, and then send tokens to that user for one time access, and finalizing the creation and activation of such a user account.

- Groups are a way to manage access to areas of the site to a group of users who have been assigned a specific group. For easier maintenance than to only manage individual users.

- Data Sources are the data CSV files that have been uploaded to the site by users or admin, and they be fully managed on the admin page by the superuser admin as well.

- Pretrained Models are the Machine Learning models the users have saved into the prediction area on the site.

- Users are exactly what it sounds like and here the admin super user can manage all aspects of them.

Figure 6: Data Editing in the Django Admin page is easy and straight forward.

The rest of the CADS Django Administration page should be more or less obvious, but if there are any questions beyond this manual, please just Google Django Admin for more details.

Summary

We wish that this manual have told you everything you need to know about how to setup, configure and maintain your own local or public CADS Server. Feel free to return here whenever your memory fails you on what to do and when.

Also, please read our other detailed manuals on how to use CADS in general terms, our manual on how to use each individual Analysis components currently available and/or our development manual on how to code and develop new CADS components.

Congratulations on your newly gained status as a 'CADS-Server-Master'. Now it is time to do some serious R&D. Enjoy!

Index

Figure Index

Thanks for your support