Matematica

Corso di Matematica

Sasso Francesco

Table of contents

• English

1. Introduzione

Esempi:

$$A \subseteq R$$

$$\lim_{n \to +\infty} f(x) = 1$$

$$f(x) = \begin{cases} 1 & -1 \leq x < 0 \\ \frac{1}{2} & x = 0 \\ x & 0 < x \leq 1 \end{cases}$$

$$\begin{matrix} 0 & & 1 \\ | & & | \\ 0 & & 0 \end{matrix}$$

$$\Delta_n(\kappa) = \frac{1}{5\pi^2\kappa^2} \int_0^{\infty} \frac{\sin(\kappa R)}{\kappa R} \frac{\partial}{\partial R} [R^2 \frac{\partial D_n(R)}{\partial R}] \, dR$$

$$\begin{aligned} a &= b + c \\ &= c + d \\ &= d + e \end{aligned}$$

Mathematical expression using inline syntax:

> The homomorphism f is injective if and only if its kernel is only the singleton set e_G, because otherwise $\exists a, b \in G$ with $a \neq b$ such that $f(a) = f(b)$.

- English: This page isn't translated to English.

2. Development tools

2.1 VS Code setup

The following section will explain how to configure VS Code to have a better development experience.

2.1.1 Suggested VS Code extensions

Here a list of extensions that can help developers to better work on the front-end code.

- Eslint: provides both linting and formatting features and is easily customizable by using the `.eslintrc` file (see here).
- Bracket Pair Colorizer: useful extension to handle brackets in the code.
- Code Spell Checker: useful to avoid typos.
- Path Intellisense: provides suggestions when entering a path in the code.
- Trailing Spaces: detects and avoids unnecessary white spaces.
- Python: enable intellisense, formatting and other stuffs related to Python files.

2.1.2 VS Code suggested settings

VS Code settings file and .vscode directory should not be included in the repository. This is a bad-practice because, in this way, if a developer has different settings in its local environment, it has to change the settings file to all developers, causing, potentially, conflicts.

For this project suggested VS Code settings are the following:

```
// .vscode/settings.json
{
    "editor.tabSize": 2,
    "workbench.editor.highlightModifiedTabs": true,
    "editor.minimap.enabled": false,
    "window.title": "${dirty}${activeEditorMedium}",
    "editor.renderWhitespace": "all",
    "trailing-spaces.trimOnSave": true,
    "files.trimFinalNewlines": true,
    "editor.codeActionsOnSave": {
        "source.fixAll": true
    },
    "eslint.alwaysShowStatus": true,
    "eslint.options": {
        "extensions": [
            ".html",
            ".js",
            ".jsx"
        ]
    },
    "eslint.validate": [
        "javascript",
        "javascriptreact"
    ],
    "eslint.workingDirectories": [
        {
            "directory": "./frontend/src/",
            "changeProcessCWD": true
        }
    ],
    "python.pythonPath": "./venv/bin/python",
    "python.analysis.extraPaths": [
        "backend"
    ],
    "cSpell.words": [
        "asgi"
    ],
    "python.linting.pylintEnabled": false,
    "python.linting.flake8Enabled": true,
    "python.linting.enabled": true,
    "python.autoComplete.addBrackets": true,
    "python.analysis.completeFunctionParens": true,
}
```

2.2 Debugging

To debug the backend, do the following:

- Set `DEBUGGER=True` in `./backend/.env`
- Add the following configuration in VS Code:

```
// .vscode/launch.json
{
  "configurations": [
    {
      // requires vscode extension: ms-python.python
      "name": "debugger",
      "type": "python",
      "request": "attach",
      "pathMappings": [
        {
          "localRoot": "${workspaceFolder}/backend",
          "remoteRoot": "/code"
        }
      ],
      "port": 5678,
      "host": "localhost",
      "logToFile": true,
    }
  ]
}
```

- Run `docker-compose up`
- Place a breakpoint in VS Code
- Click the Debug button

To debug the frontend, you can use the keyword `debugger` combined with Google Chrome developer tool.

2.3 Intellisense and autosuggestions for backend

Since this is a dockerized project, it is not trivial to configure the proper interpreter with VS Code. Setting the correct interpreter is indeed fundamental to have intellisense and autosuggetions in our backend code. To this aim, we suggest an easy (even though not elegant) workaround: create a virtualenv and install the backend requirements, i.e.,

```
(root) ⚅ python3 -m venv venv
(root) ⚅ source venv/bin/activate
[venv] (root) ⚅ cd backend
[venv] (backend) ⚅ pip install -r requirements.txt
```

In this way, we will have the python interpreter in `./venv/bin/python` (which is almost identical to the one in the backend container) and we can set that easily in VS Code (see `"python.pythonPath": "./venv/bin/python"` in `settings.json` file above).

2.4 Fabric commands

An additional `fabfile.py` module has been added to facilitate the development experience and to execute the following commands:

- `(root) fab ssh:{service-name}` : used to SSH inside a given container.

> **Note**
>
> `{service-name}` must be one of the following:
>
> - `backend`
> - `worker`
> - `frontend`
> - `postgres`
> - `docs`

- `(root) fab remove_pyc` : used to remove `*.pyc` files from the backend
- `(root) fab remove_migrations` : used to remove migrations files from the backend
- `(root) fab reset_db` : used to reset the DB
- `(root) fab notebook` : used to start Jupyter notebook
- `(root) fab formatter` : used to format backend (through black package) and frontend (through eslint package)
- `(root) fab serve:{build-name}` : used to build the frontend and serve it at http://localhost:4000.

> **Note**
>
> `{build-name}` must be equal to `build` . In future, we might add new scripts in `package.json` , and therefore `{build-name}` will consequently be in a larger set of options.

2.5 Code Analysis

The project also allows to do code analysis with SonarQube. To this aim, execute the following steps:

- run `docker-compose -f docker-compose.sonar.yaml up`
- access http://localhost:9000
- if it is the first time that you do the login, then use the default credentials: `username: admin` & `password: admin` (the system will then enforce you to change the password)
- Now that you're logged in to your local SonarQube instance, let's analyze a project:
  - Click the Create new project button.
  - Give your project a Project key and a Display name and click the Set Up button.
  - Under Provide a token, select Generate a token. Give your token a name, click the Generate button, and click Continue.
  - Select your project's main language under Run analysis on your project, and follow the instructions to analyze your project. Here you'll download and execute a Scanner on your code (if you're using Maven or Gradle, the Scanner is automatically downloaded).

- English: This page isn't translated to English.

3. Backend

Application's backend is based on Django.

3.1 Requirements

In the following, we will list the main requirements used with the related capabilities.

3.1.1 Django background tasks

Django background tasks is a databased-backed work queue for Django. It allows to execute long running processes in background.

> **Note**
>
> This particular requirement is also used by the `worker` service, which is responsible for the execution of long running tasks.

3.1.2 Django filters

Django filters is a generic, reusable application to alleviate writing some of the more mundane bits of view code. Specifically, it allows users to filter down a queryset based on a model's fields, displaying the form to let them do this.

3.1.3 Django rest framework

Django rest framework is a powerful and flexible toolkit for building Web APIs. Here some reasons for using it:

- Serialization that supports both ORM and non-ORM data sources.
- Customizable all the way down - just use regular function-based views if you don't need the more powerful features.
- Extensive documentation, and great community support.
- Used and trusted by internationally recognised companies including Mozilla, Red Hat, Heroku, and Eventbrite.

3.1.4 Django rest framework API Key

Django API Key is a powerful library for allowing server-side clients to safely use your API. These clients are typically third-party backends and services (i.e. machines) which do not have a user account but still need to interact with your API in a secure way.

3.1.5 Django reversion

Django reversion is an extension to the Django web framework that provides version control for model instances. In particular, provides the following features:

- Roll back to any point in a model instance's history.
- Recover deleted model instances.
- Simple admin integration.

It is very useful as an audit tracking tool.

3.1.6 Django split settings

Django split settings organize Django settings into multiple files and directories. Easily override and modify settings. Use wildcards in settings file paths and mark settings files as optional.

3.1.7 Djoser

Djoser provides a set of Django Rest Framework views to handle basic actions such as registration, login, logout, password reset and account activation. It works with custom user model.

3.1.8 DRF spectacular

DRF spectacular is an integral part of API development, and OpenAPI 3 is finally here to make that process a easier. By using drf-spectacular with Django rest framework, your schema and therefore your documentation & client will always stay close to your API.

3.1.9 Faker

Faker is a Python package that generates fake data for you. Whether you need to bootstrap your database, create good-looking XML documents, fill-in your persistence to stress test it, or anonymize data taken from a production service, Faker is for you.

3.1.10 Pygal

Pygal is a dynamic SVG charting library written in python. It is useful to embed info-graphics into PDF reports.

3.1.11 Python socket.io

Python socket.io implements Socket.IO clients and servers that can run standalone or integrated with a variety of Python web frameworks. Useful to create real time applications with Python.

3.1.12 Weasyprint

Weasyprint is visual rendering engine for HTML and CSS that can export to PDF. It aims to support web standards for printing. It is extremely useful to generate PDF reports.

3.2 Examples

Throughout the code, you'll be able to see an optimal set of examples, including:

- Models
- Serializers
- Filters
- Class & function based views
- Authentication & Permissions
- Real time communication
- PDF report generation
- Long running tasks execution

> **Tip**
>
> It is highly recommended to check the code to see how to structure new backend features, and to visit requirements' user guides to learn how to use them.

- English: This page isn't translated to English.

4. Frontend

The application's frontend is based on React.

4.1 Packages

In the following, we will list the main packages used with the related capabilities.

4.1.1 Ant design

Ant design: a design system for enterprise-level products. Create an efficient and enjoyable work experience.

4.1.2 Axios

Axios: promise based HTTP client for the browser and node.js.

4.1.3 i18next

i18next is an internationalization-framework written in and for JavaScript.

4.1.4 Nivo

Nivo provides a rich set of dataviz components, built on top of the awesome d3 and Reactjs libraries.

4.1.5 React error boundary

React error boundary provides a simple and reusable wrapper that you can use to wrap around your components. Any rendering errors in your components hierarchy can then be gracefully handled.

4.1.6 React router

React router is a collection of navigational components that compose declaratively with your application.

4.1.7 SimpleR state

SimpleR state is an ultra-lightweight library that provides the simplest state management for React.

4.1.8 Socket.io client

Socket.io client provides an easy API to create socket.io clients.

4.2 Examples

Throughout the code, you'll be able to see an optimal set of examples, including:

- Authentication & Permissions
- CRUD operations
- Tables
- Chat
- Real time communication
- Forms
- Charts
- Internationalization

> **Tip**
>
> It is highly recommended to check the code to see how to structure new frontend features, and to visit packages' user guides to learn how to use them.

- English: This page isn't translated to English.

5. Material for MKDocs

In this section we will briefly explain how to update the user guide within this repo. For additional info, check the Material for MKDocs official guide.

5.1 Update the user guide

To update the user guide, follow these steps:

- Run the `docker-compose.docs.yaml` configuration:

```
(root) docker-compose -f docker-compose.docs.yaml up
```

- Open the browser at http://localhost:9000.
- Change the documents within the `docs/docs` folder (this will trigger an auto-reload that will automatically perform changes in the UI)

5.2 Build the user guide

To build the user guide, follow these steps:

- Run the `docker-compose.docs.yaml` configuration:

```
(root) docker-compose -f docker-compose.docs.yaml up
```

- SSH into the docs container:

```
(root) docker exec -it docs sh
```

- Build the user guide by running the command:

```
/code # mkdocs build
```

this command will build the user guide into the `docs/site` folder. This folder can then be used for deployment purposes (this can be deployed, for instance, using an S3 bucket).

5.3 Markdown syntax

This user guide is written using the markdown syntax. In this section we report an optimal set of markdown samples that are useful for the creation of the user guide. For more info, visit this link.

> **Tip**
>
> To understand how to reproduce the examples given below, it is highly recommended to see `docs/docs/mkdocs.md`.

5.3.1 Formatting

- **Bold**
- *Italic*
- `Code`
- Highlight

- Underline
- ~~Strikeout~~
- ~~Deleted~~
- Added
- ~~Prev~~New
- /* Commented */
- Small
- Block:

  Formatting can also be applied to blocks, by putting the opening and closing tags on separate lines and adding new lines between the tags and the content.

5.3.2 Admonitions

Phasellus posuere in sem ut cursus

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

Collapsible admonition

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

Info

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

Tip

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

Success

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

Question

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

Warning

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

Failure

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

Danger

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

Bug

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

Example

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

Quote

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

Inline Info

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

```
# code -> line 1
# code -> line 2
# code -> line 3
# code -> line 4
# code -> line 5
```

5.3.3 Code blocks

```
document$.subscribe(function() {
  var tables = document.querySelectorAll("article table")
  tables.forEach(function(table) {
    new Tablesort(table)
```

```
  })
})
```

```
    def bubble_sort(items):
        for i in range(len(items)):
            for j in range(len(items) - 1 - i):
                if items[j] > items[j + 1]:
                    items[j], items[j + 1] = items[j + 1], items[j]
```

```
def bubble_sort(items):
    for i in range(len(items)):
        for j in range(len(items) - 1 - i):
            if items[j] > items[j + 1]:
                items[j], items[j + 1] = items[j + 1], items[j]
```

5.3.4 Content tabs

C

```
#include <stdio.h>

int main(void) {
  printf("Hello world!\n");
  return 0;
}
```

C++

```
#include <iostream>

int main(void) {
  std::cout << "Hello world!" << std::endl;
  return 0;
}
```

Embedded content tabs

Unordered List

Example:

```
* Sed sagittis eleifend rutrum
* Donec vitae suscipit est
* Nulla tempor lobortis orci
```

Result:

- Sed sagittis eleifend rutrum
- Donec vitae suscipit est
- Nulla tempor lobortis orci

Ordered List

Example:

```
1. Sed sagittis eleifend rutrum
2. Donec vitae suscipit est
3. Nulla tempor lobortis orci
```

Result:

1. Sed sagittis eleifend rutrum
2. Donec vitae suscipit est
3. Nulla tempor lobortis orci

5.3.5 Tables

Example

Left

Method	Description
GET	✓ Fetch resource
PUT	✓✓ Update resource
DELETE	✗ Delete resource

Right

Method	Description
GET	✓ Fetch resource
PUT	✓✓ Update resource
DELETE	✗ Delete resource

Center

Method	Description
GET	✓ Fetch resource
PUT	✓✓ Update resource
DELETE	✗ Delete resource

5.3.6 Emoji & Icons

- 😄
- – .icons/material/account-circle.svg
- – .icons/fontawesome/regular/laugh-wink.svg
- – .icons/octicons/repo-push-24.svg

5.3.7 Links

- Link to markdown guide
- Link to **Update the user guide section**
- Link to *Introduction*
- Link to Introduction Start the project

5.3.8 Footnotes

Lorem ipsum[1] dolor sit amet, consectetur adipiscing elit[2].

5.3.9 Abbreviations

The HTML specification is maintained by the W3C.

5.3.10 Lists

- Indented code block:

```
def bubble_sort(items):
    for i in range(len(items)):
        for j in range(len(items) - 1 - i):
            if items[j] > items[j + 1]:
                items[j], items[j + 1] = items[j + 1], items[j]
```

- Indented quote block:

> **Quote**
>
> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

Additional paragraph within this list item.

- Nested (ordered/unordered) list:

a. first

b. second

- unordered item
- another unordered item

`Lorem ipsum dolor sit amet`

Sed sagittis eleifend rutrum. Donec vitae suscipit est. Nullam tempus tellus non sem sollicitudin, quis rutrum leo facilisis.

`Cras arcu libero`

Aliquam metus eros, pretium sed nulla venenatis, faucibus auctor ex. Proin ut eros sed sapien ullamcorper consequat. Nunc ligula ante.

- [x] Lorem ipsum dolor sit amet, consectetur adipiscing elit
- [ ] Vestibulum convallis sit amet nisi a tincidunt
  - [x] In hac habitasse platea dictumst
  - [x] In scelerisque nibh non dolor mollis congue sed et metus
  - [ ] Praesent sed risus massa
- [ ] Aenean pretium efficitur erat, donec pharetra, ligula non scelerisque

---

5.3.11 Images

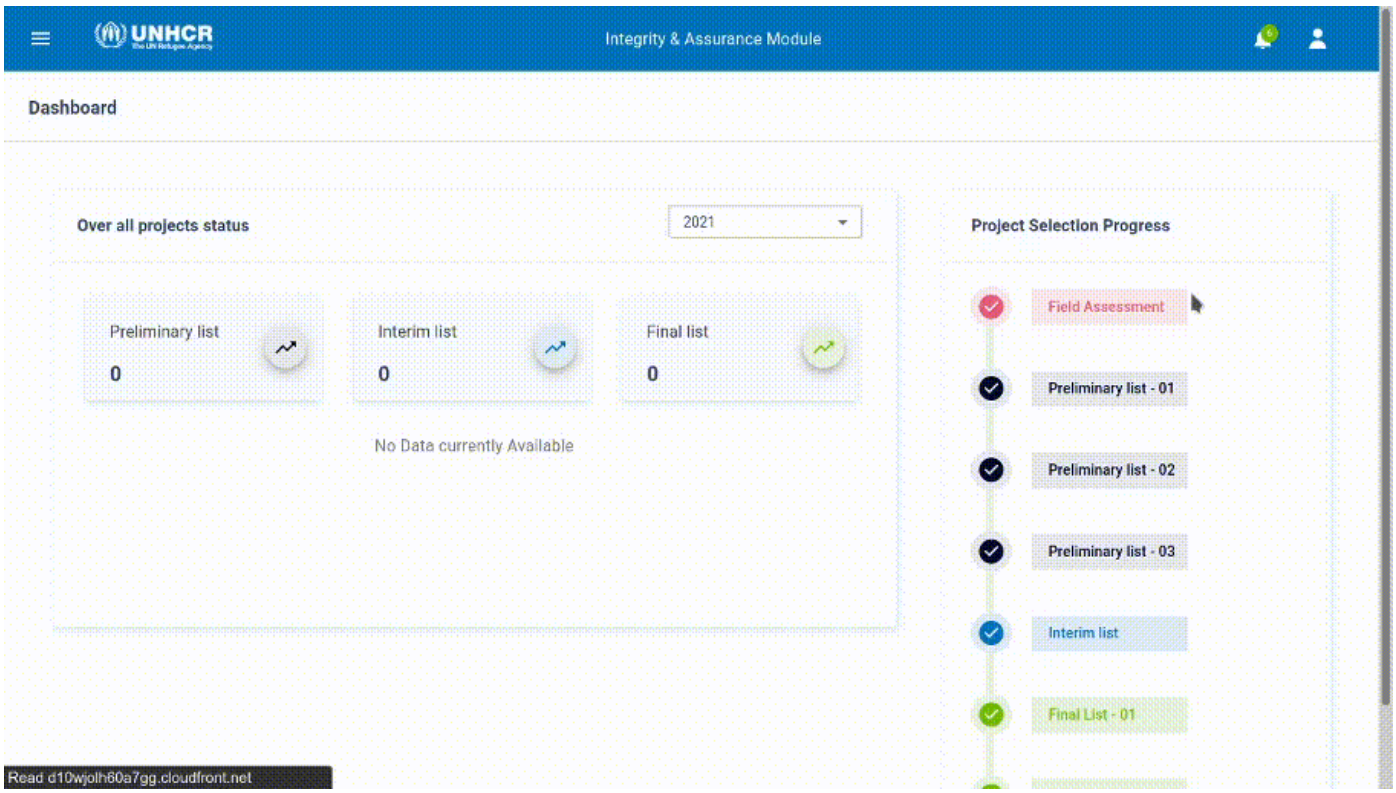

Fig.1 - GIF example with caption.

5.3.12 MathJax

MathJax is a beautiful and accessible way to display mathematical content in the browser.

Mathematical expression using block syntax:

$$\ker f = \{g \in G : f(g) = e_H\}.$$

Mathematical expression using inline syntax:

> The homomorphism f is injective if and only if its kernel is only the singleton set e_G, because otherwise $\exists a, b \in G$ with $a \neq b$ such that $f(a) = f(b)$.

1. Lorem ipsum dolor sit amet, consectetur adipiscing elit. ↩
2. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa. ↩