

YAML Intellisense in Jina App Development



Yongxuanzhang

Mar 17 · 4 min read

Developing a Jina app often means writing YAML configs. Our new release now supports Intellisense on PyCharm and VSCode for YAML-level autocompletion, syntax validation, members listing, and help display. These features make writing Flow/Executor YAML much easier and greatly speed up your Jina development. In this blog post, we'll introduce these new features and how to use them step by step.



Image source: [1]

What is YAML?

YAML (a recursive acronym for “YAML Ain’t Markup Language”) is a human-readable data-serialization language [2]. It is often used to write configuration files that are simple and human-readable.



Image source: [3]

How is YAML used in Jina Development?

Jina applications usually contains two types of source files: Python files and YAML files. Python files define the entrypoint and customize logic. YAML is extensively used in Jina app development for Flow and Executor configuration.

The following example demonstrates Jina’s Flow configuration in YAML. Jina’s YAML syntax can be found [here] (<https://docs.jina.ai/chapters/yaml/>).

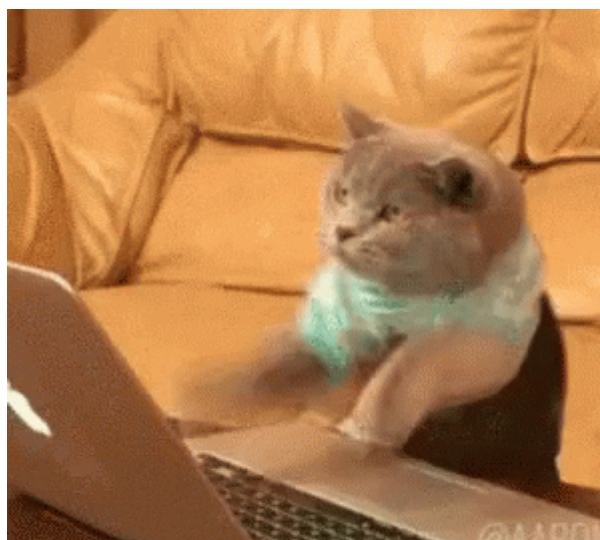
```
jtype: Flow
version: '1'
pods:
  - name: segmenter
    show_exc_info: true
    uses: pods/segment.yml
```

```
  read_only: true
- name: encoder
  show_exc_info: true
  uses: pods/encode.yml
  polling: any
  shards: $JINA_PARALLEL
  timeout_ready: 600000
  read_only: true
- name: chunk_idx
  polling: any
  show_exc_info: true
  uses: pods/chunk.yml
  shards: $JINA_SHARDS
- name: doc_idx
  polling: any
  uses: pods/doc.yml
  needs: gateway
- name: join_all
  uses: _merge
  needs: [doc_idx, chunk_idx]
```

Most IDE's provide Intellisense for Python development. However, for YAML config IDE's can only help to provide simple syntax highlighting. This may slow down your development speed since you need to navigate the documentation or codebase to write a correct and bug-free YAML config. This process may cause frustration and headaches, so Intellisense support is quite important.

Intellisense Support in Jina

Intellisense can provide code completion, syntax validation, argument filtering, default value filling and help text display. Enabling Intellisense will save you lots of time in writing YAML configuration files and avoid potential bugs.



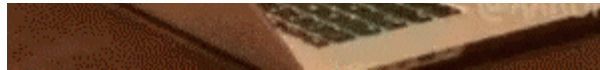
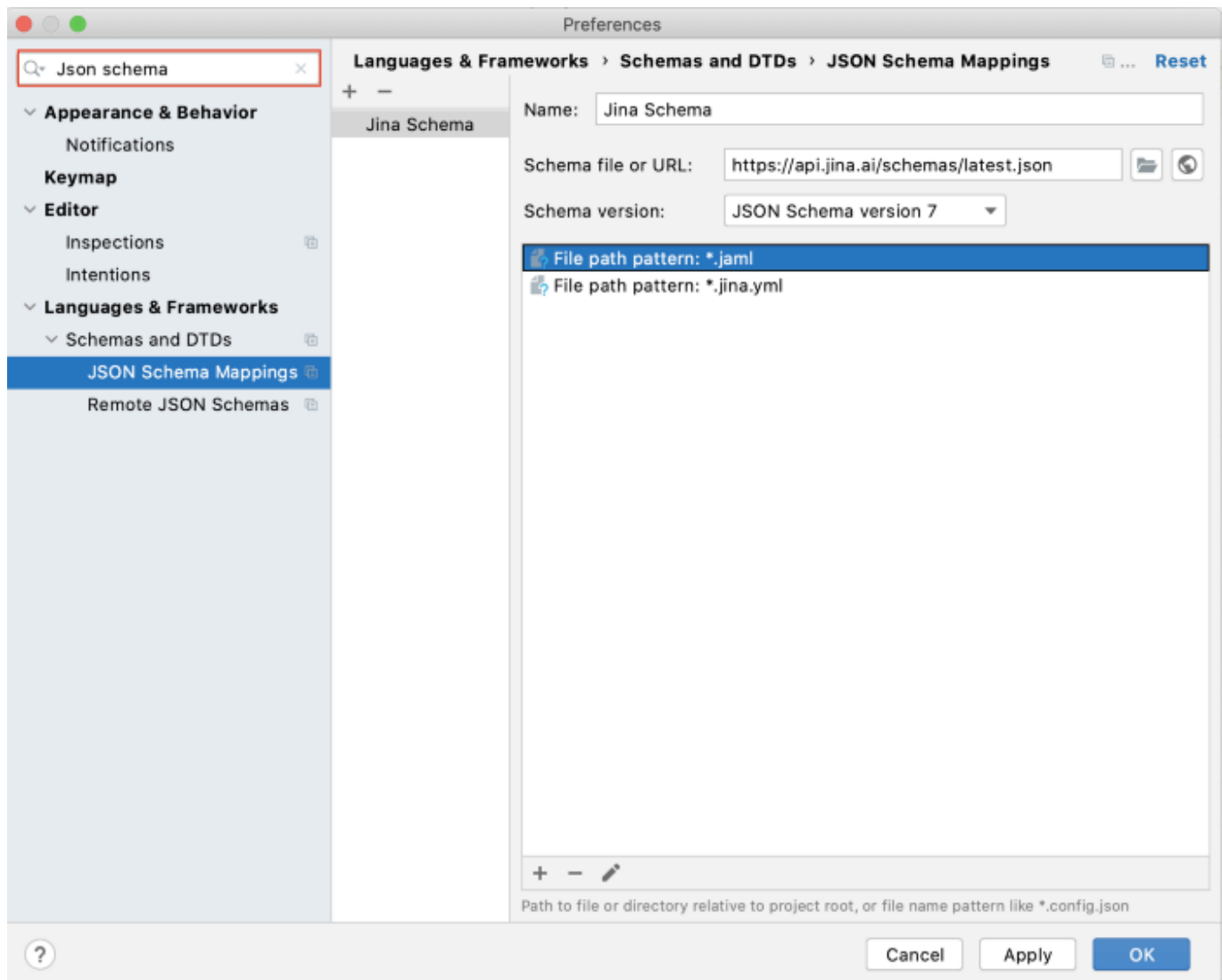


Image source: [4]

Configuration in Pycharm

Step 1:

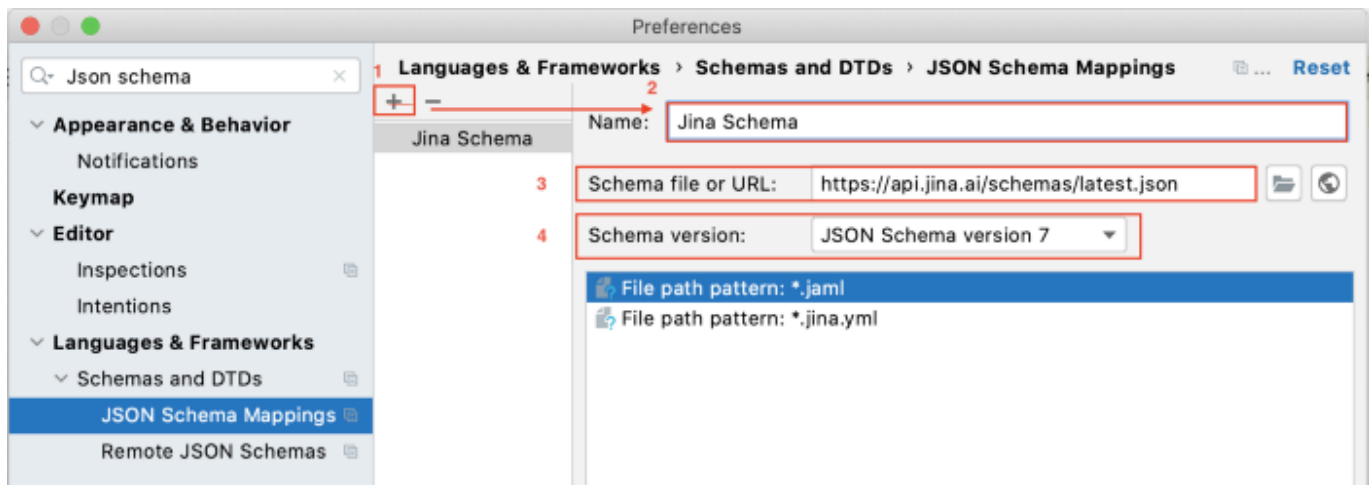
- Click on the menu
- Go to *Preferences*
- Search for JSON Schema mappings



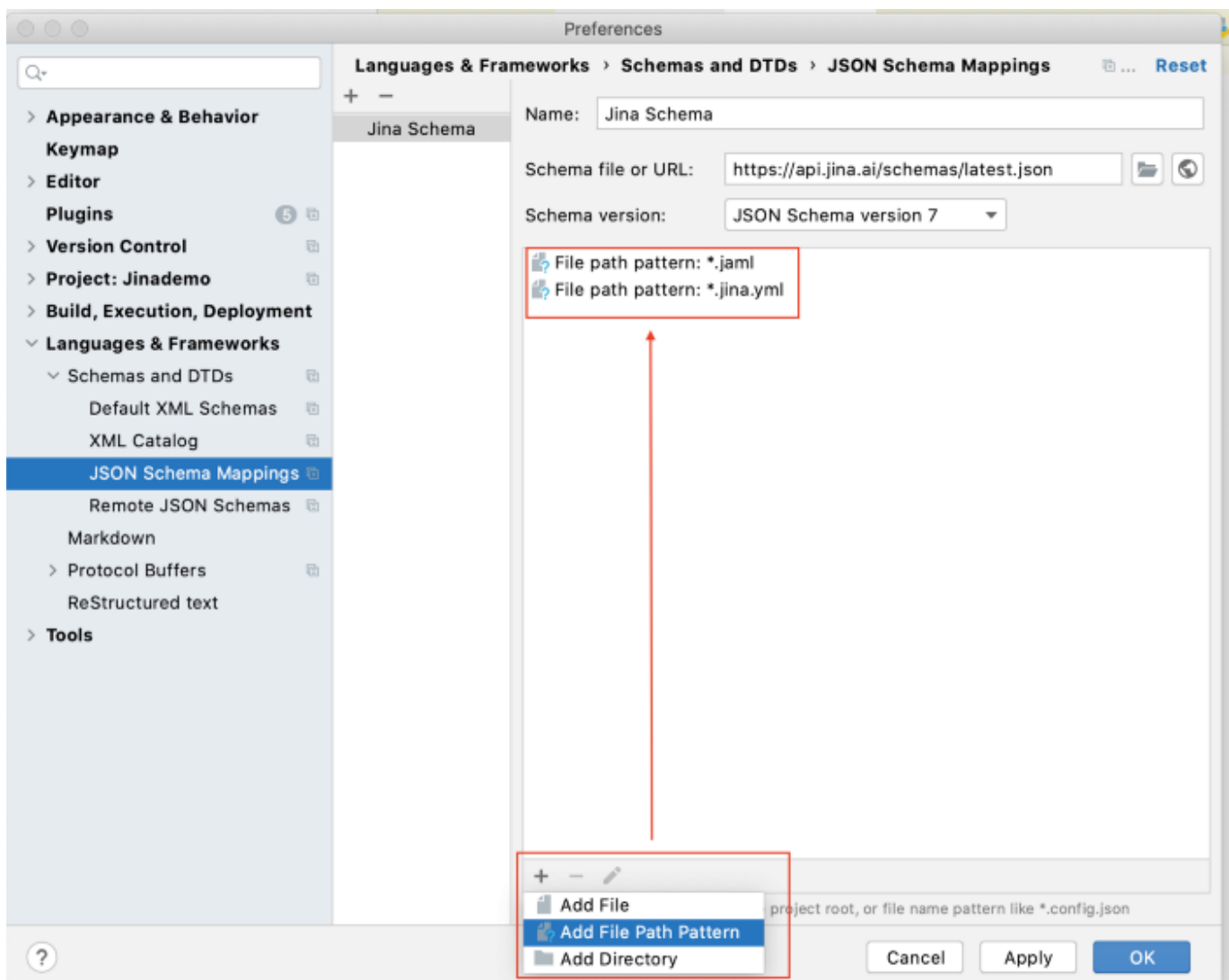
Step 2:

- Add a new schema with a custom name.

- In the *Schema File or URL* write <https://api.jina.ai/schemas/latest.json>.
- Select *JSON Schema Version 7* for *Schema version*.

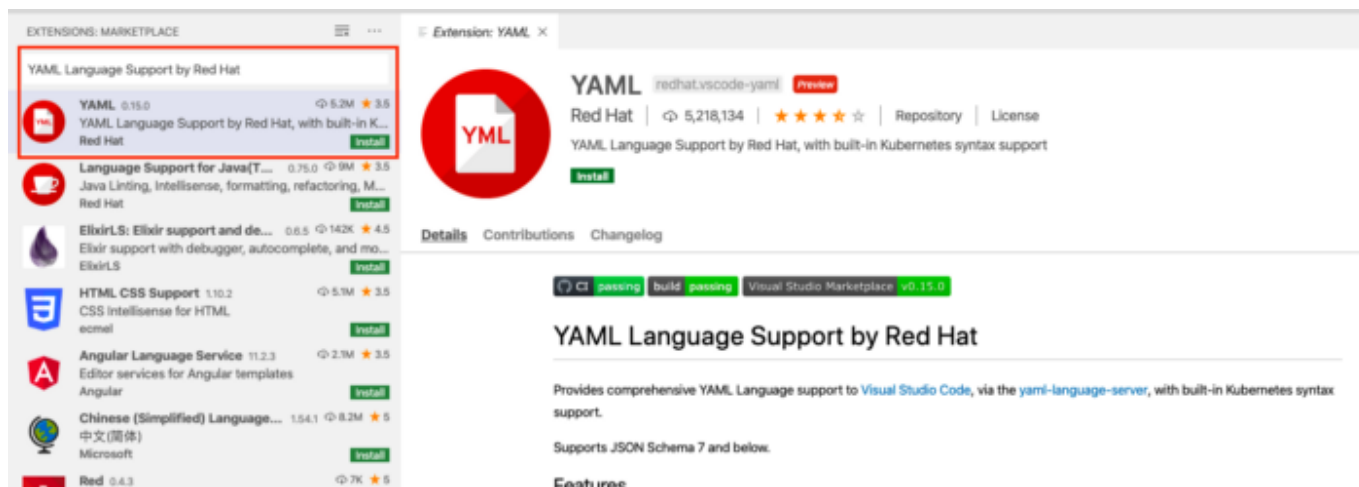


Step 3: Add a *file path pattern* and link it to `*.yaml` and `*.jina.yaml`.



Configuration in VSCode

Step 1: Install the extension: *YAML Language Support by Red Hat*;

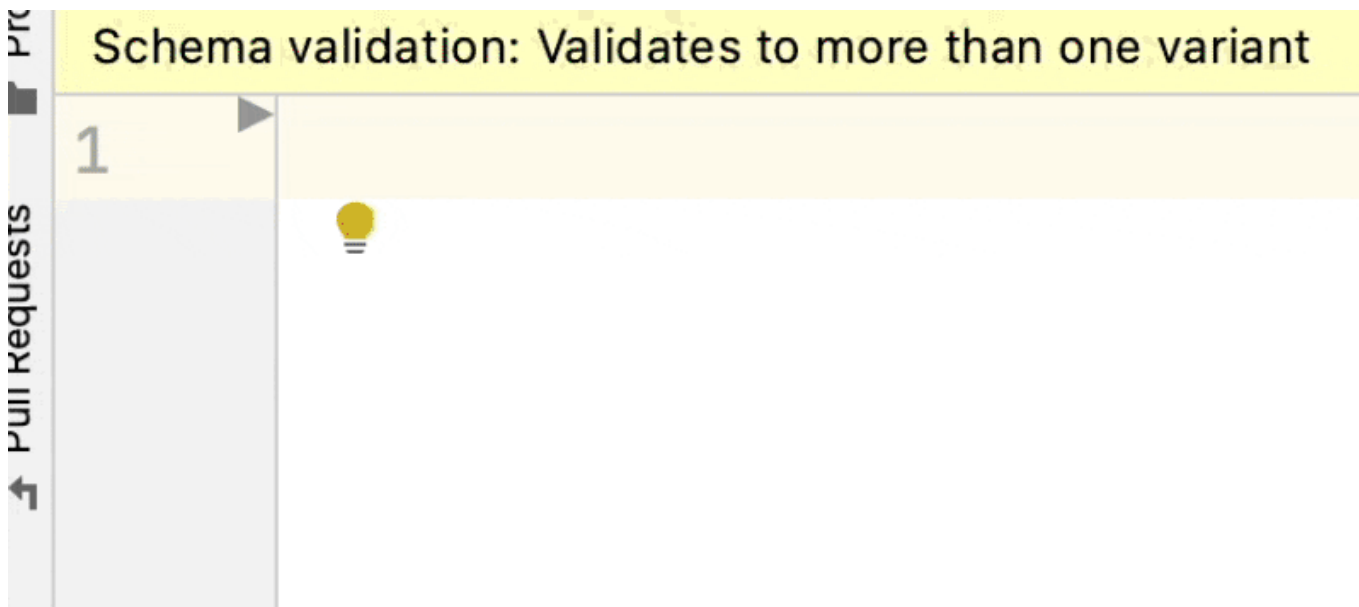


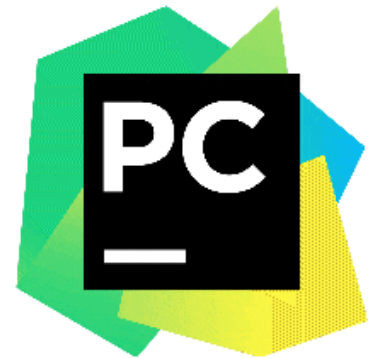
Step 2: In IDE-level settings.json add:

```
"yaml.schemas": {
  "https://api.jina.ai/schemas/latest.json": ["/*.jina.yml",
"/*.jaml"],
}
```

Intellisense In Action

Flow-level YAML





First, let's create `helloworld.jina.yml` file in your IDE. You can also manually configure the schema if your IDE doesn't recognize it.

Now type `jtype` in the first line and you will see the IDE suggests `flow` or a list of Executors in Jina. The IDE immediately marks it as yellow as it is not a valid Jina config file. Hover your mouse over it and it complains you haven't defined the required fields, *version* and *Pods*.

When you write something unrecognizable by the schema, the IDE will mark it as yellow immediately.

Executor-level YAML

```
1  jtype: NumpyIndexer
2
3
```



Now let's create a new YAML file for configuring `NumpyIndexer`.

You can see the IDE will keep only the arguments that are accepted by that Executor. It also works on the nested level when you define a `request:` on , it will only show Drivers. And when you define the Drivers it will show relevant arguments that are accepted by that Driver.

Conclusion

In this post, we introduced our new features for YAML Intellisense. We hope this will help you speed up developing Jina applications.

In the meantime, you can follow us on [Twitter](#), [GitHub](#), or join our [Slack community](#).

Footnotes

1. Cover image source: <https://unsplash.com/photos/FCHlYvR5gJI>
2. <https://en.wikipedia.org/wiki/YAML>
3. Image source: <https://unsplash.com/photos/itPxxOCMMuY>

4. Image source: <https://giphy.com/gifs/memecandy-LmNwrBhejkK9EFP504>

Machine Learning

Yaml

Data Science

Python

Open Source

[About](#) [Help](#) [Legal](#)

Get the Medium app

