Serialization

It is often preferrable to store prompts not as python code but as files. This can make it easy to share, store, and version prompts. This notebook covers how to do that in LangChain, walking through all the different types of prompts and the different serialization options.

At a high level, the following design principles are applied to serialization:

1. Both JSON and YAML are supported. We want to support serialization methods that are human readable on disk, and YAML and JSON are two of the most popular methods for that. Note that this rule applies to prompts. For other assets, like Examples, different serialization methods may be supported.

2. We support specifying everything in one file, or storing different components (templates, examples, etc) in different files and referencing them. For some cases, storing everything in file makes the most sense, but for others it is preferrable to split up some of the assets (long templates, large examples, reusable components). LangChain supports both.

There is also a single entry point to load prompts from disk, making it easy to load any type of prompt.

```
# All prompts are loaded through the `load_prompt` function.
from langchain.prompts import load_prompt
```

PromptTemplate

This section covers examples for loading a PromptTemplate.

Loading from YAML

This shows an example of loading a PromptTemplate from YAML.

```
cat simple_prompt.yaml
```

```
    _type: prompt
    input_variables:
        ["adjective", "content"]
    template:
        Tell me a {adjective} joke about {content}.
```

```
prompt = load_prompt("simple_prompt.yaml")
print(prompt.format(adjective="funny", content="chickens"))
```

```
    Tell me a funny joke about chickens.
```

Loading from JSON

This shows an example of loading a PromptTemplate from JSON.

```
cat simple_prompt.json
```

```
    {
        "_type": "prompt",
        "input_variables": ["adjective", "content"],
        "template": "Tell me a {adjective} joke about {content}."
    }
```

```
prompt = load_prompt("simple_prompt.json")
print(prompt.format(adjective="funny", content="chickens"))
```

```
Tell me a funny joke about chickens.
```

Loading Template from a File

This shows an example of storing the template in a separate file and then referencing it in the config. Notice that the key changes from `template` to `template_path`.

```
cat simple_template.txt
```

```
    Tell me a {adjective} joke about {content}.
```

```
cat simple_prompt_with_template_file.json
```

```
    {
        "_type": "prompt",
        "input_variables": ["adjective", "content"],
```

```
    "template_path": "simple_template.txt"
}
```

```
prompt = load_prompt("simple_prompt_with_template_file.json")
print(prompt.format(adjective="funny", content="chickens"))
```

```
Tell me a funny joke about chickens.
```

FewShotPromptTemplate

This section covers examples for loading few shot prompt templates.

Examples

This shows an example of what examples stored as json might look like.

```
cat examples.json
```

```
[
    {"input": "happy", "output": "sad"},
    {"input": "tall", "output": "short"}
]
```

And here is what the same examples stored as yaml might look like.

```
cat examples.yaml
```

```
    - input: happy
      output: sad
    - input: tall
      output: short
```

Loading from YAML

This shows an example of loading a few shot example from YAML.

```
cat few_shot_prompt.yaml
```

```
    _type: few_shot
    input_variables:
        ["adjective"]
    prefix:
        Write antonyms for the following words.
    example_prompt:
        _type: prompt
        input_variables:
            ["input", "output"]
        template:
            "Input: {input}\nOutput: {output}"
    examples:
        examples.json
```

```
    suffix:
        "Input: {adjective}\nOutput:"
```

```
prompt = load_prompt("few_shot_prompt.yaml")
print(prompt.format(adjective="funny"))
```

```
    Write antonyms for the following words.

    Input: happy
    Output: sad

    Input: tall
    Output: short

    Input: funny
    Output:
```

The same would work if you loaded examples from the yaml file.

```
cat few_shot_prompt_yaml_examples.yaml
```

```
    _type: few_shot
    input_variables:
        ["adjective"]
    prefix:
        Write antonyms for the following words.
    example_prompt:
        _type: prompt
```

```
    input_variables:
        ["input", "output"]
    template:
        "Input: {input}\nOutput: {output}"
examples:
    examples.yaml
suffix:
    "Input: {adjective}\nOutput:"
```

```
prompt = load_prompt("few_shot_prompt_yaml_examples.yaml")
print(prompt.format(adjective="funny"))
```

```
Write antonyms for the following words.

Input: happy
Output: sad

Input: tall
Output: short

Input: funny
Output:
```

Loading from JSON

This shows an example of loading a few shot example from JSON.

```
cat few_shot_prompt.json
```

```
    {
        "_type": "few_shot",
        "input_variables": ["adjective"],
        "prefix": "Write antonyms for the following words.",
        "example_prompt": {
            "_type": "prompt",
            "input_variables": ["input", "output"],
            "template": "Input: {input}\nOutput: {output}"
        },
        "examples": "examples.json",
        "suffix": "Input: {adjective}\nOutput:"
    }
```

```
prompt = load_prompt("few_shot_prompt.json")
print(prompt.format(adjective="funny"))
```

```
    Write antonyms for the following words.

    Input: happy
    Output: sad

    Input: tall
    Output: short

    Input: funny
    Output:
```

Examples in the Config

This shows an example of referencing the examples directly in the config.

```
cat few_shot_prompt_examples_in.json
```

```
    {
        "_type": "few_shot",
        "input_variables": ["adjective"],
        "prefix": "Write antonyms for the following words.",
        "example_prompt": {
            "_type": "prompt",
            "input_variables": ["input", "output"],
            "template": "Input: {input}\nOutput: {output}"
        },
        "examples": [
            {"input": "happy", "output": "sad"},
            {"input": "tall", "output": "short"}
        ],
        "suffix": "Input: {adjective}\nOutput:"
    }
```

```
prompt = load_prompt("few_shot_prompt_examples_in.json")
print(prompt.format(adjective="funny"))
```

```
    Write antonyms for the following words.

    Input: happy
    Output: sad
```

```
Input: tall
Output: short

Input: funny
Output:
```

Example Prompt from a File

This shows an example of loading the PromptTemplate that is used to format the examples from a separate file. Note that the key changes from `example_prompt` to `example_prompt_path`.

```
cat example_prompt.json
```

```
{
    "_type": "prompt",
    "input_variables": ["input", "output"],
    "template": "Input: {input}\nOutput: {output}"
}
```

```
cat few_shot_prompt_example_prompt.json
```

```
{
    "_type": "few_shot",
    "input_variables": ["adjective"],
    "prefix": "Write antonyms for the following words.",
    "example_prompt_path": "example_prompt.json",
    "examples": "examples.json",
```

```
    "suffix": "Input: {adjective}\nOutput:"
}
```

```
prompt = load_prompt("few_shot_prompt_example_prompt.json")
print(prompt.format(adjective="funny"))
```

```
Write antonyms for the following words.

Input: happy
Output: sad

Input: tall
Output: short

Input: funny
Output:
```

PromptTempalte with OutputParser

This shows an example of loading a prompt along with an OutputParser from a file.

```
cat prompt_with_output_parser.json
```

```
{
    "input_variables": [
        "question",
```

```
            "student_answer"
        ],
        "output_parser": {
            "regex": "(.*?)\\nScore: (.*)",
            "output_keys": [
                "answer",
                "score"
            ],
            "default_output_key": null,
            "_type": "regex_parser"
        },
        "partial_variables": {},
        "template": "Given the following question and student answer, provide a correct answer and score the
student answer.\nQuestion: {question}\nStudent Answer: {student_answer}\nCorrect Answer:",
        "template_format": "f-string",
        "validate_template": true,
        "_type": "prompt"
    }
```

```
prompt = load_prompt("prompt_with_output_parser.json")
```

```
prompt.output_parser.parse(
    "George Washington was born in 1732 and died in 1799.\nScore: 1/2"
)
```

```
    {'answer': 'George Washington was born in 1732 and died in 1799.',
     'score': '1/2'}
```