

OutParser

June 14, 2024

```
[ ]: import { ChatOpenAI } from "@langchain/openai";
import { HumanMessage } from "@langchain/core/messages";
import {
  ↳StringOutputParser,StructuredOutputParser,CommaSeparatedListOutputParser }↳
  ↳from "@langchain/core/output_parsers";
import {PromptTemplate} from '@langchain/core/prompts'
import { load } from "dotenv";
let env = await load();
let process = { env };

let model = new ChatOpenAI({
  configuration: {
    baseUrl: process.env.baseUrl,
  },
});
```

String Output Parset

```
[ ]: let parser = new StringOutputParser();
let chain = model.pipe(parser);
await chain.invoke([new HumanMessage("  ")]);
```

```
[ ]: "          \n          "
```

StructuredOutputParser

```
[ ]: let parser = StructuredOutputParser.fromNamesAndDescriptions({
  answer: "  ",
  evidence: "  ",
  confidence: "  ",
});
let prompt = PromptTemplate.fromTemplate("          \n{instructions} \n{question}");
let chain = prompt.pipe(model).pipe(parser);
let res = await chain.invoke({
  question: "  ",
  instructions: parser.getFormatInstructions(),
});
console.log(res);
```

```
{
  answer: " .      ",
  evidence: " .      ",
  confidence: "100%"
}
```

List Output Parser

```
[ ]: let parser = new CommaSeparatedListOutputParser();
console.log(parser.getFormatInstructions());
let prompt = PromptTemplate.fromTemplate(" {name} \n{instructions}");
let chain = prompt.pipe(model).pipe(parser)
let res = await chain.invoke({
  name: ' ',
  instructions: parser.getFormatInstructions()
})
console.log(res)
```

Your response should be a list of comma separated values, eg: `foo, bar, baz`