Moses: give me model configurations for you, GPT-5.1, to be used in my IDE via API. with maximum reasoning powers.

GPT-5.1-Extended Thinking:

High level first:

For “maximum reasoning powers” you want:

- The **flagship model**: `gpt-5.1` (OpenAI Platform)
- Via **OpenAI-compatible API** (your local router or direct).
- With **reasoning turned up**: `reasoning_effort: "high"` (highest setting; lower values are `none | low | medium`). (OpenAI)
- With **plenty of completion tokens** and **low temperature** for reliability.
- Optional: **24h prompt caching** for long-running agent/IDE sessions. (OpenAI)

---

1. Target config at a glance

Layer	Setting	Value (recommended)	Purpose
Transport	Base URL	`http://localhost:8000/v1` or `https://api.openai.com/v1`	Where your IDE sends OpenAI-style requests
Endpoint	Path	`/chat/completions`	Safest, most widely supported in IDEs (Responses API is better but less common yet) (OpenAI Platform)
Model	`model`	`"gpt-5.1"`	Flagship GPT-5.1 model in API (OpenAI Platform)
Reasoning	`reasoning_effort`	`"high"`	Max reasoning depth; prioritize intelligence over latency (OpenAI)
Tokens	`max_completion_tokens`	`8192`	Big headroom for thinking + answer; below hard limits (OpenAI)
Style	`temperature`	`0.2`	Stable, low-noise reasoning
Sampling	`top_p`	`1`	Standard full support sampling
System msg	`messages[0].role` / `content`	Strong, single system prompt	Steers behavior consistently

Layer	Setting	Value (recommended)	Purpose
Caching	`prompt_cache_retention` (optional)	`"24h"`	Keep long IDE sessions cheap/fast (OpenAI)

---

2. Drop-in JSON body (for IDE “custom model” / router)

Use this as your **canonical config** any time your IDE asks for the “OpenAI chat/completions body” or a default request template.

```
{
  "model": "gpt-5.1",
  "reasoning_effort": "high",
  "max_completion_tokens": 8192,
  "temperature": 0.2,
  "top_p": 1,
  "frequency_penalty": 0,
  "presence_penalty": 0,
  "prompt_cache_retention": "24h",
  "messages": [
    {
      "role": "system",
      "content": "You are GPT-5.1 Thinking, a highly capable reasoning model. Always solve problems step by step, check your work, and favor correctness over speed."
    }
  ]
}
```

Notes:

- `reasoning_effort: "high"` is what effectively gives you the “Thinking” behavior programmatically. (OpenAI)
- `prompt_cache_retention` is honored on GPT-5.1 to keep a big initial context cached up to 24h, ideal for IDE/agent sessions. (OpenAI)
- You can append user messages to `messages` as usual; your IDE or router will typically inject them automatically.

---

3. Base HTTP config (for your local router / IDE)

Point any OpenAI-compatible IDE (Cursor, VS Code plugins, Open WebUI, etc.) at your router like this:

- **Base URL**: `http://localhost:8000/v1`
- **Endpoint**: `/chat/completions`
- **HTTP method**: `POST`
- **Headers**:

  - `Authorization: Bearer YOUR_API_KEY`
  - `Content-Type: application/json`

Example raw request (what the IDE will effectively send):

```
curl -X POST "http://localhost:8000/v1/chat/completions" \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "model": "gpt-5.1",
    "reasoning_effort": "high",
    "max_completion_tokens": 8192,
    "temperature": 0.2,
    "top_p": 1,
    "prompt_cache_retention": "24h",
    "messages": [
      {
        "role": "system",
        "content": "You are GPT-5.1 Thinking, a highly capable reasoning model. Always solve problems step by step, check your work, and favor correctness over speed."
      },
      {
        "role": "user",
        "content": "Explain what this code is doing and suggest improvements."
      }
    ]
  }'
```

You can:

- Keep this as your **single “max-reasoning” profile** in IDEs.
- If a specific tool only exposes “Model name + API key + base URL”, just:
  - Model: `gpt-5.1`
  - Base URL: `http://localhost:8000/v1`
  - And configure the extra fields (`reasoning_effort`, `max_completion_tokens`, etc.) wherever it allows “extra JSON” or “default body” overrides.

If you tell me which IDE/plugin first (Cursor, JetBrains AI, VS Code extension, Open WebUI, etc.), I can map this exact config to its specific UI fields one by one.

Moses: Can you double check these? Just making sure I don't get an API error. "reasoning_effort": "high", "max_completion_tokens": 8192, "temperature": 0.2, "top_p": 1,

---

GPT-5.1-Extended Thinking:

Short answer:
All four fields are valid and correctly typed for GPT-5.x reasoning models via the `/chat/completions` API. You will not get an API error from these names/values alone, as long as you include them in a valid JSON body.

---

1. Parameter sanity check

Field	**Valid?**	**Required type**	**Your value**	**OK?**	**Notes**
`reasoning_effort`	Yes	string enum	`"high"`	☑	Allowed values: `"none"`, `"low"`, `"medium"`, `"high"` on 5.1 reasoning models. (OpenAI Platform)
`max_completion_tokens`	Yes	integer	`8192`	☑	Newer reasoning/chat models use `max_completion_tokens` instead of `max_tokens`; this is the correct field. (Microsoft Learn)
`temperature`	Yes	number [0–2]	`0.2`	☑	Standard sampling param; 0.2 is a good “focused reasoning” value. (OpenAI Help Center)
`top_p`	Yes	number (0–1]	`1`	☑	Full nucleus probability mass (i.e., no extra truncation). (OpenAI Help Center)

So this block is safe:

```
{
  "reasoning_effort": "high",
  "max_completion_tokens": 8192,
  "temperature": 0.2,
  "top_p": 1
}
```

---

2. Where these work (and where they don’t)

- `reasoning_effort` + `max_completion_tokens` are accepted on **reasoning-capable GPT-5.x / gpt-5.1 models**, including Azure-style `chat/completions` as shown in Microsoft’s docs. (Microsoft Learn)
- On older non-reasoning models, `reasoning_effort` may be ignored or rejected; `max_tokens` might still be the expected field. But for 5.1 you’re doing it the new way.

As long as:

- `model` is set to something like `"gpt-5.1"` (or the exact 5.1 variant you’re using), and
- The rest of your JSON is valid (messages array, etc.),

you will not get an “unsupported parameter” error from these four.

Moses: Can you have temperature at 0.2 when you have "high" reasoning? crosscheck with original OpenAI documentation. And provide the exact content from the source about this.

GPT-5.1-Extended Thinking:

Short version:
No. When `reasoning_effort` is `"high"` (i.e., GPT-5.1 is running in reasoning mode), you **must not** send a custom `temperature` (0.2 or anything else). For reasoning models, sampling parameters like `temperature` are either fixed internally or rejected, and that’s reflected in both OpenAI’s docs and real error messages.

---

1. What OpenAI’s own docs say

From the **Chat Completions API reference** (this is the page for `/v1/chat/completions`):

> “Parameter support can differ depending on the model used to generate the response, particularly for newer reasoning models.” (OpenAI Platform)

> “For the current state of unsupported parameters in reasoning models, refer to the reasoning guide.” (OpenAI Platform)

The **Reasoning models guide** (the page that text is pointing to) explicitly describes fixed sampling params for o-series reasoning models, e.g. for `o1-preview`:

> “`temperature` and `top_p` are fixed at `1`, while `presence_penalty` and `frequency_penalty` are fixed at `0`.” (LINUX DO)

So for reasoning models, the official guidance is: sampling knobs like `temperature` are not meant to be user-tuned; they’re locked.

The **GPT-5.1 usage guide** then confirms that GPT-5.1 is also a reasoning model controlled via the `reasoning.effort` setting:

> “The reasoning.effort parameter controls how many reasoning tokens the model generates before producing a response.” (OpenAI Platform)

And the **Chat create reference** for GPT-5.1’s reasoning effort makes it explicit that `reasoning_effort` is the supported control:

> “reasoning_effort … Constrains effort on reasoning for reasoning models. … Supported values are none, low, medium, and high. gpt-5.1 defaults to none…” (OpenAI Platform)

Crucially, on that same page, `reasoning_effort` is documented for GPT-5.1, but `temperature` has no special allowance or override for reasoning models; instead you’re told to check the reasoning guide for unsupported parameters. (OpenAI Platform)

---

2. Evidence from actual GPT-5 / reasoning-model behavior

Multiple official/first-party or closely-tied sources show that reasoning models reject `temperature`:

- OpenAI community thread for GPT-5 models:

  > “Unsupported parameter: 'temperature' is not supported with this model.” (OpenAI Developer Community)

- GitHub issues for o-series reasoning models show the same 400 error when `temperature` is passed: (GitHub)

- Third-party libraries that track OpenAI’s parameter matrix summarize it as:

  > “o1 and o3 models … do not support temperature, top_p, frequency_penalty, or presence_penalty parameters.” (openai-structured.readthedocs.io)

- Another integration doc (Mem0) generalizes this for reasoning models:

  > “Note: The following are currently unsupported with reasoning models … temperature, top_p, … max_tokens” (docs.mem0.ai)

A recent bug report for GPT-5.1 notes that temperature only works when **reasoning is off**:

> “Temperature is supported if reasoning is set to none.” (GitHub)

That matches the pattern: once you enable reasoning (`reasoning_effort` ≠ `"none"`), sampling params stop being configurable.

---

3. Direct answer to your question

Question:
“Can you have temperature at 0.2 when you have `"high"` reasoning?”

Answer (based on docs + behavior):

- For **GPT-5.1 with `reasoning_effort: "high"`** (i.e., in reasoning mode), **no**:
  - OpenAI’s reasoning guide treats sampling parameters as fixed (or unsupported) on reasoning models. (OpenAI Platform)
  - Chat API docs tell you to consult that guide for unsupported parameters. (OpenAI Platform)
  - Real calls to GPT-5 reasoning models produce `"Unsupported parameter: 'temperature' is not supported with this model."` when you send `temperature`. (OpenAI Developer Community)
- For **GPT-5.1 with `reasoning_effort: "none"`**:
  - Community + library evidence shows `temperature` behaves like on non-reasoning models and is allowed. (GitHub)

So:

- `reasoning_effort: "high"` + `temperature: 0.2` ⇒ **expect a 400 error**.

- If you want **maximum reasoning**, **do not send `temperature` or `top_p` at all**. Let them stay at the internal defaults.
- If you really need to tune `temperature`, you must run GPT-5.1 with `reasoning_effort: "none"` (no reasoning tokens) or use a non-reasoning model.

---

4. What you should actually send

For “max reasoning” on GPT-5.1 via `/v1/chat/completions`, your safe core config is:

```
{
  "model": "gpt-5.1",
  "reasoning_effort": "high",
  "max_completion_tokens": 8192,
  "messages": [
    { "role": "system", "content": "…" },
    { "role": "user", "content": "…" }
  ]
  // ⚠ Do NOT include temperature or top_p here
}
```

If/when you want temperature control instead of reasoning:

```
{
  "model": "gpt-5.1",
  "reasoning_effort": "none",
  "temperature": 0.2,
  "max_completion_tokens": 8192,
  "messages": [ … ]
}
```

But for your original goal (“maximum reasoning powers”), drop `temperature` entirely when `reasoning_effort` is `"high"` to avoid API errors.