

# Lecture 33: Image Options

## Introduction:

- When generating images using Spring AI's ImagePrompt, we can customize output by passing **image options** along with the query. These options let us control aspects such as quality, size, style, and number of images.

## Image Prompt with Options:

- ImagePrompt supports two constructors:
  - One takes just a query (string).
  - Another accepts a query **and** ImageOptions.
- To create options, we use the **Builder pattern**:

```
● ● ●  
OpenAiImageOptions.builder()  
    .quality("hd")  
    .height(1024)  
    .width(1024)  
    .style("natural")  
    .build();
```

## Why Builder?

- Builder makes it easy to set multiple values step by step.
- build() returns an object of OpenAiImageOptions (implements ImageOptions).
- This allows flexible customization of generated images.

## Common Options

- Quality**
  - Values: "standard" or "hd"
  - Example: .quality("hd")
- Height & Width**
  - Supported: 1024 x 1024
  - Example: .height(1024).width(1024)
- Number of Images (n)**
  - For DALL·E 3 → only one image is supported.
- Style**
  - Values: "vivid," "natural"
  - Example: .style("natural")

## Code Implementation:

```
● ● ●
@GetMapping("/image/{query}")
public String genImage(@PathVariable String query) {

    ImagePrompt prompt = new ImagePrompt(query, OpenAiImageOptions.builder()
        .quality("hd")
        .height(1024)
        .width(1024)
        .style("natural")
        .build());

    ImageResponse response = openAiImageModel.call(prompt);
    return response.getResult().getOutput().getUrl();
}
```

## Key Notes

- Errors help discover valid option values (e.g., "hd" must be lowercase).
- Only certain values are allowed; unsupported ones cause exceptions.
- Style affects output appearance (e.g., "natural" gives realistic images).
- Use options to fine-tune generated results instead of default settings.