

# Lecture 45: Bean Output Converter with List

## Introduction:

- The **Bean Output Converter** is not limited to converting a single object
- It can also be used to fetch and map a **list of objects** in JSON format.
- This is useful when we want multiple records, such as a list of movies for a given actor.

## Why Use Bean Output Converter with List?

- For a single object → normal **BeanOutputConverter<Movie>**.
- For multiple objects → use **BeanOutputConverter** with a **Parameterized Type Reference**.
- Ensures AI responses map directly to a **List<Movie>**.
- Returns structured JSON data that can be directly passed into Java collections.

## Code Implementation:

```
@GetMapping("/moviesList")
public List<Movie> getMovieList(@RequestParam String name) {

    BeanOutputConverter<Movie> opCon = new BeanOutputConverter<Movie>(Movie.class);

    List<Movie> movies = chatClient.prompt()
        .user(u -> u.text("Top 5 movies of {name}").param("name", name))
        .call()
        .entity(new BeanOutputConverter<List<Movie>>( new ParameterizedTypeReference<List<Movie>>() {}));

    return movies;
}
```

## Explanation:

- **Prompt** → Top 5 movies of {name}.
- **BeanOutputConverter with ParameterizedTypeReference** → Tells Spring AI to expect a list of **Movie** objects.
- **AI Response** → JSON array of movies, each with fields (movieName, leadActor, director, year).
- **Conversion** → Maps directly into **List<Movie>**.
- **Return Value** → Ready-to-use structured list of objects.

## Example Output:

- **Request:** /moviesList?name=Tom Hanks
- **Response (JSON):**

```
[
  {
    "movieName": "Forrest Gump",
    "leadActor": "Tom Hanks",
    "director": "Robert Zemeckis",
    "year": 1994
  },
  {
    "movieName": "Cast Away",
    "leadActor": "Tom Hanks",
    "director": "Robert Zemeckis",
    "year": 2000
  }
]
```

## Key Points:

- For lists, use `ParameterizedTypeReference` with **BeanOutputConverter**.
- **Works the same way as single bean conversion but returns multiple objects.**
- Useful for scenarios like lists of movies, products, books, etc.
- Provides consistent, structured data from AI to application.

## Summary:

- **Bean Output Converter with List** allows converting AI output into **List<T>**.
- Requires **ParameterizedTypeReference** for type safety.
- Returns structured JSON mapped to Java objects
- Makes AI integration seamless for multiple records.