

# Passive Income with Stock Images

## OpenAI Model Instructions

### **\*\* Task \*\***

Create metadata for a single image in valid JSON format.

The output must be a JSON array containing one or more objects, where each object includes:

"Title" – A descriptive name for the image. Between 50 - 80 characters

"Keywords" – A list of 55 keywords, each relevant to the image content, with a max length of 30 characters each

### **\*\* Format Requirements \*\***

- Return a valid JSON array (no markdown or extra text)
- The Keywords field must always include exactly 55 strings
- Each keyword string must be under or equal to 30 characters
- Follow strict JSON syntax

### **\*\* Example Output \*\***

```
[
  {
    "Title": "Sunset Over the Ocean",
    "Keywords": ["sunset", "ocean", "sea", "water", "golden hour", "nature", "scenic",
    "landscape", "travel", "peaceful", "sun", "sky", "evening", "coast", "waves", "reflection",
    "tranquil", "serene", "beach", "clouds", "light", "summer", "dusk", "horizon", "natural beauty"]
  }
]
```

### **\*\* Important \*\***

Only return the JSON array. Do not include explanations, formatting syntax, or headers.

## JSON code

```
// Extract the JSON response from the previous step
const inputData = $input.first().json.content;

// Strip out possible markdown wrappers like ```json
const sanitizedData = inputData.replace(/```json\n\n/g, "");

try {
  // Attempt to parse the JSON string
  const dataObjects = JSON.parse(sanitizedData);

  // Format each object: convert keywords array into a comma-separated string
  const result = dataObjects.map(entry => ({
    json: {
      Title: entry.Title,
      Description: entry.Description,
      Keywords: entry.Keywords.join(', ')
    }
  }
  )
```

```
});  
  
    return result;  
} catch (err) {  
    // Throw an error if the JSON is invalid  
    throw new Error("JSON parsing failed: " + err.message);  
}
```