In [ ]:

```python
from bs4 import BeautifulSoup
import pandas as pd
import requests

# Define the URL of the website to scrape
url = 'https://www.cityfurniture.com/browse/living-room'  # Replace with the actual URL of the inventory page

# Send a request to the website
response = requests.get(url)
#response.raise_for_status()  # Check if the request was successful

# Parse the webpage content
soup = BeautifulSoup(response.content, 'html.parser')


# Find the relevant data (This example assumes there are divs with class 'product' containing inventory info)
products = []
for product_div in soup.find_all('div', class_='product'):  # Adjust the class name based on the actual HTML structure
    product_name = product_div.find('div', class_='product-name').text.strip()  # Adjust based on actual HTML structure
    inventory_status = product_div.find('div', class_='inventory-status').text.strip()  # Adjust based on actual HTML structure

    # Add the data to the list
    products.append({
        'Product Name': product_name,
        'Inventory Status': inventory_status
    })

# Convert the list of dictionaries to a pandas DataFrame
df = pd.DataFrame(products)

# Save the DataFrame to a CSV file
df.to_csv('inventory_data.csv', index=False)

print('Inventory data has been saved to inventory_data.csv')
print(soup.prettify())  # Print the entire HTML structure
for product_div in soup.find_all('div', class_='product-item'):
    print(product_div)  # Print each product div
    product_name = product_div.find('div', class_='product-title').text.strip()
    print(product_name)  # Print the product name
    inventory_status = product_div.find('div', class_='stock-status').text.strip()
    print(inventory_status)  # Print the inventory status
```

```
Inventory data has been saved to inventory_data.csv
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://ww
w.w3.org/TR/html4/loose.dtd">
<html>
 <head>
  <meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>
  <title>
   ERROR: The request could not be satisfied
  </title>
 </head>
 <body>
  <h1>
   403 ERROR
  </h1>
  <h2>
   The request could not be satisfied.
  </h2>
  <hr noshade="" size="1px"/>
  Request blocked.
We can't connect to the server for this app or website at this time. There
might be too much traffic or a configuration error. Try again later, or co
ntact the app or website owner.
  <br clear="all"/>
  If you provide content to customers through CloudFront, you can find ste
ps to troubleshoot and help prevent this error by reviewing the CloudFront
documentation.
  <br clear="all"/>
  <hr noshade="" size="1px"/>
  <pre>
Generated by cloudfront (CloudFront)
Request ID: ARjPzAupZj8GgSAi0Nv3uvFqcRXlJazO2srq0wlZRjJcT69Ae6Lh6Q==
</pre>
  <address>
  </address>
 </body>
</html>
```