Splunk Leaflet Maps - Corrected Inline Search Queries for Testing

Quick Start - Test These Queries in Splunk

After applying the fixes, test the visualization with these inline searches. Open Splunk Web at **http://localhost:8000** and navigate to Search & Reporting.

✅ Test 1: Basic City Markers (Recommended First Test)

Purpose: Verify basic functionality with simple data

Query:

```
| makeresults
| eval latitude=40.7128, longitude=-74.0060, description="New York City", lay-
er="city"
| append [| makeresults | eval latitude=34.0522, longitude=-118.2437, description="Los
Angeles", layer="city"]
| append [| makeresults | eval latitude=41.8781, longitude=-87.6298, descrip-
tion="Chicago", layer="city"]
| append [| makeresults | eval latitude=29.7604, longitude=-95.3698, descrip-
tion="Houston", layer="city"]
| append [| makeresults | eval latitude=33.4484, longitude=-112.0740, descrip-
tion="Phoenix", layer="city"]
| table latitude longitude description layer
```

Expected Result:
- Map displays with 5 markers across the United States
- All markers are the same color (default for “city” category)
- Clicking markers shows popup with description
- Layer controls on the right show “city” layer with 5 items

✅ Test 2: Multiple Categories with Colors

Purpose: Test layer controls and category separation

Query:

```
| makeresults
| eval latitude=40.7589, longitude=-73.9851, description="Times Square", cat-
egory="landmark"
| append [| makeresults | eval latitude=40.7829, longitude=-73.9654, descrip-
tion="Central Park", category="park"]
| append [| makeresults | eval latitude=40.6892, longitude=-74.0445, descrip-
tion="Statue of Liberty", category="landmark"]
| append [| makeresults | eval latitude=40.7614, longitude=-73.9776, descrip-
tion="Bryant Park", category="park"]
| append [| makeresults | eval latitude=40.7484, longitude=-73.9857, descrip-
tion="Empire State Building", category="landmark"]
| append [| makeresults | eval latitude=40.7505, longitude=-73.9934, descrip-
tion="Madison Square Garden", category="landmark"]
| table latitude longitude description category
```

Expected Result:
- Two different colored marker groups (landmarks and parks)
- Layer controls show both “landmark” and “park” categories
- Can toggle each category on/off independently
- Can change colors using color pickers

✅ Test 3: Using Short Field Names (lat/lon)

Purpose: Verify field name variations are recognized

Query:

```
| makeresults
| eval lat=37.7749, lon=-122.4194, name="San Francisco", type="city"
| append [| makeresults | eval lat=47.6062, lon=-122.3321, name="Seattle",
type="city"]
| append [| makeresults | eval lat=45.5152, lon=-122.6784, name="Portland", type="city
"]
| append [| makeresults | eval lat=32.7157, lon=-117.1611, name="San Diego", type="cit
y"]
| table lat lon name type
```

Expected Result:
- 4 markers on the West Coast
- Visualization correctly recognizes “lat” and “lon” field names
- Uses “name” field for description
- Console logs show field mapping: lat→latitude, lon→longitude

---

✅ Test 4: Custom Colors Per Marker

Purpose: Test custom color support

Query:

```
| makeresults
| eval latitude=40.7128, longitude=-74.0060, description="New York City", cat-
egory="city", color="#FF0000"
| append [| makeresults | eval latitude=34.0522, longitude=-118.2437, description="Los
Angeles", category="city", color="#00FF00"]
| append [| makeresults | eval latitude=41.8781, longitude=-87.6298, descrip-
tion="Chicago", category="city", color="#0000FF"]
| append [| makeresults | eval latitude=29.7604, longitude=-95.3698, descrip-
tion="Houston", category="city", color="#FFFF00"]
| table latitude longitude description category color
```

Expected Result:
- Each marker displays in its specified color
- Red marker for NYC, Green for LA, Blue for Chicago, Yellow for Houston
- Custom colors override category defaults

✅ Test 5: Testing Error Handling - Missing Fields

Purpose: Verify improved error messages

Query:

```
| makeresults
| eval loc_lat=40.7128, loc_lon=-74.0060, place="New York City"
| table loc_lat loc_lon place
```

Expected Result:
- Error message displayed: “Required fields not found”
- Error shows available fields: loc_lat, loc_lon, place
- Error suggests accepted field names
- Error provides SPL tip to rename fields:

`spl`
`  | rename loc_lat AS latitude, loc_lon AS longitude`

Fixed Query:

```
| makeresults
| eval loc_lat=40.7128, loc_lon=-74.0060, place="New York City"
| rename loc_lat AS latitude, loc_lon AS longitude, place AS description
| table latitude longitude description
```

✅ Test 6: US Highway Rest Areas (Arkansas Example)

Purpose: Test with predefined category type

Query:

```
| makeresults
| eval latitude=35.8242, longitude=-90.7043, description="Rest Area Mile 10", cat-
egory="rest_area"
| append [| makeresults | eval latitude=34.9273, longitude=-92.3890, descrip-
tion="Welcome Center North", category="welcome_center"]
| append [| makeresults | eval latitude=35.3859, longitude=-94.3985, description="Rest
Area I-40 West", category="rest_area"]
| append [| makeresults | eval latitude=35.0070, longitude=-91.9543, descrip-
tion="Weigh Station East", category="weigh_station"]
| append [| makeresults | eval latitude=35.1495, longitude=-90.0490, descrip-
tion="Welcome Center East", category="welcome_center"]
| table latitude longitude description category
```

Expected Result:
- 3 different category types with distinct colors
- rest_area: Turquoise markers
- welcome_center: Sky Blue markers
- weigh_station: Light Salmon markers

✅ Test 7: World Capitals (Testing Map Range)

Purpose: Verify map handles global coordinates

Query:

```
| makeresults
| eval latitude=51.5074, longitude=-0.1278, description="London, UK", cat-
egory="capital"
| append [| makeresults | eval latitude=48.8566, longitude=2.3522, description="Paris,
France", category="capital"]
| append [| makeresults | eval latitude=52.5200, longitude=13.4050, descrip-
tion="Berlin, Germany", category="capital"]
| append [| makeresults | eval latitude=35.6762, longitude=139.6503, descrip-
tion="Tokyo, Japan", category="capital"]
| append [| makeresults | eval latitude=-33.8688, longitude=151.2093, descrip-
tion="Sydney, Australia", category="capital"]
| table latitude longitude description category
```

Expected Result:
- Markers appear on multiple continents
- Map automatically adjusts view
- All coordinates within valid ranges (-90 to 90 lat, -180 to 180 lon)

✅ Test 8: Testing Invalid Coordinates (Error Handling)

Purpose: Verify coordinate validation

Query:

```
| makeresults
| eval latitude="invalid", longitude=-74.0060, description="Test Invalid Lat"
| append [| makeresults | eval latitude=40.7128, longitude="invalid", descrip-
tion="Test Invalid Lon"]
| append [| makeresults | eval latitude=999, longitude=-74.0060, description="Test Out
of Range Lat"]
| append [| makeresults | eval latitude=40.7128, longitude=999, description="Test Out
of Range Lon"]
| table latitude longitude description
```

Expected Result:
- Error message: “No valid data points found”
- Console logs show which rows were skipped and why
- Detailed error explains coordinate requirements
- Provides SPL tip to convert to numbers

Fixed Query:

```
| makeresults
| eval latitude=40.7128, longitude=-74.0060, description="Valid Point 1"
| append [| makeresults | eval latitude=34.0522, longitude=-118.2437, descrip-
tion="Valid Point 2"]
| eval latitude=tonumber(latitude), longitude=tonumber(longitude)
| table latitude longitude description
```

Debugging Tips

How to View Console Logs

1. Open your browser’s Developer Tools:
   - **Chrome/Edge**: Press `F12` or `Ctrl+Shift+I`
   - **Firefox**: Press `F12` or `Ctrl+Shift+K`
   - **Safari**: Press `Cmd+Option+I`

2. Click on the **Console** tab

3. Run your search with the visualization selected

4. Look for detailed logs showing:
   `=== formatData called ===`
   `Data object: {...}`
   `Field names: ["latitude", "longitude", "description", "layer"]`
   `Field indices found:`
   `latIndex: 0 (field: latitude)`
   `lonIndex: 1 (field: longitude)`
   `Processing summary:`
   `Valid rows: 5`
   `Invalid rows: 0`
   `=== formatData completed successfully ===`

Common Issues and Solutions

Issue	Cause	Solution
Map shows no tiles	Internet connection or incorrect tile URL	Verify fixes were applied correctly
“Required fields not found”	Field names don’t match accepted variations	Rename fields using `| rename`
“No valid data points found”	Coordinates are invalid or out of range	Use `| eval latitude=tonumber(latitude)`
“No data available”	Search returned no results	Add `| table *` to verify search results
Markers in wrong location	Latitude and longitude are swapped	Check console logs, verify lat is -90 to 90

After Testing

If Tests Work ✅

Your visualization is working correctly! You can now:
1. Use it with your own CSV files
2. Connect to indexed data
3. Customize layer colors and categories

If Tests Fail ❌

1. **Check browser console** for detailed error messages
2. **Verify Splunk was restarted** after pulling changes
3. **Clear browser cache** (Ctrl+Shift+R)
4. **Check internet connection** (needed for map tiles)
5. **Verify file permissions** on visualization files

Integration with Your Data

Example: Using with CSV Lookup

```
| inputlookup your_locations.csv
| rename lat AS latitude, lon AS longitude
| eval category="your_category"
| table latitude longitude description category
```

Example: Using with Indexed Data

```
index=your_index sourcetype=your_sourcetype
| eval latitude=tonumber(location_lat), longitude=tonumber(location_lon)
| table latitude longitude description category
```

Example: Aggregating Data

```
index=events
| stats count by location_lat, location_lon, event_type
| rename location_lat AS latitude, location_lon AS longitude, event_type AS category
| eval description=event_type + ": " + tostring(count) + " events"
| table latitude longitude description category
```

Performance Considerations

- **Maximum recommended data points**: 10,000
- **For large datasets**, consider:
- Filtering by time range: `earliest=-24h`
- Geographical filtering: `| where latitude > 30 AND latitude < 50`
- Aggregation: `| stats count by rounded_lat, rounded_lon`
- Head limit: `| head 1000`

Support

If issues persist after testing these queries:

1. Open browser console (F12) and copy all logs
2. Note which test query you were running
3. Share the error messages and console output
4. Check FIX_SUMMARY.md (FIX_SUMMARY.md) for additional troubleshooting

---

Last Updated: November 14, 2025
Compatible with: Splunk 9.4.x
Visualization Version: 1.1.0 (Fixed)